Modeling and managing noise in quantum error correction

by

Stefanie J. Beale

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, 2023

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:	Robert Raussendorf Professor, University of British Columbia, Dept. of Physics and Astronomy
Supervisor(s):	Joel J. Wallman Adjunct Professor, University of Waterloo, Dept. of Applied Mathematics
	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:	Joseph Emerson Professor, University of Waterloo Dept. of Applied Mathematics
Other Member(s):	Richard Cleve Professor, University of Waterloo Dept. of Physics and Astronomy

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 new results presented in this thesis are all the product of close collaboration between Stefanie Beale and Joel Wallman. Other contributions were given by Andrew Doherty and Ben Criger in the form of questions or discussion that sparked further work by Stefanie Beale; these contributions are acknowledged below where they appear.

1. Chapter 1: Introduction

    - The materials presented in this section were drawn from material presented in Stefanie Beale's Masters thesis, [1], drafts and final versions of papers presented in subsequent chapters [2–4] (all of which resulted from collaboration between Stefanie Beale and Joel Wallman), and original writings by Stefanie Beale. 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: Efficiently computing logical noise in quantum-error-correcting codes

    - The writing and results presented in this chapter is modified from work previously published in [2], for which Stefanie Beale was the first author. Her contributions included development of ideas, mathematical proofs, and writing. This paper was co-authored by Joel Wallman. Andrew Doherty contributed to this work by suggesting one of the presented symmetries in the toric code.
    - The proof for symmetries in multiple rounds of QEC was not included in [2] and was derived post-publication by Stefanie Beale.

3. Chapter 3: Randomized compiling for subsystem measurements

    - The writing and results presented in this chapter is modified from work previously published in [3], for which Stefanie Beale was the first author. Her contributions included development of ideas, mathematical proofs, derivations of specific and general examples, and writing. This paper was co-authored by Joel Wallman. Some background material drew loosely on unpublished notes, to which Samuele Ferracin, Matthew Graydon, and Emily Wright additionally contributed.

4. Chapter 4: Logical randomized compiling

    - The writing and results presented in this chapter is modified from work previously published in [4], for which Stefanie Beale was the first author. Her contributions included development of ideas, mathematical proofs, and writing. This paper was co-authored by Joel Wallman. The argument that stabilizer randomization can still be used when the surrounding circuits differ was included in response to a question from Ben Criger.

Abstract

Simulating a quantum system to full accuracy is very costly and often impossible as we do not know the exact dynamics of a given system. In particular, the dynamics of measurement noise are not well understood. For this reason, and especially in the context of quantum error correction, where we are studying a larger system with branching outcomes due to syndrome measurement, studies often assume a probabilistic Pauli (or Weyl) noise model on the system with probabilistically misreported outcomes for the measurements. In this thesis, we explore methods to decrease the computational complexity of simulating encoded memory channels by deriving conditions under which effective channels are equivalent up to logical operations. Leveraging this method allows for a significant reduction in computational complexity when simulating quantum error correcting codes. We then propose methods to enforce a model consistent with the typical assumptions of stochastic Pauli (or Weyl) noise with probabilistically misreported measurement outcomes: first via a new protocol we call *measurement randomized compiling*, which enforces an average noise on measurements wherein measurement outcomes are probabilistically misreported; then by another new protocol we call *logical randomized compiling*, which enforces the same model on syndrome measurements and a probabilistic Pauli (or Weyl) noise model on all other operations (including idling). Together, these results enable more efficient simulation of quantum error correction systems by enforcing effective noise of a form which is easier to model and by reducing the simulation overhead further via symmetries. The enforced effective noise model is additionally consistent with standard error correction procedures and enables techniques founded upon the standard assumptions to be applied in any setting where our protocols are simultaneously applied.

Acknowledgements

First and foremost, I would like to thank my supervisors, Joel Wallman and Ray Laflamme for their invaluable support and guidance throughout this degree; I have learned a lot from our interactions and am grateful to have been able to grow with them. I would also like to thank my advisory and examining committee members, Roger Melko, Richard Cleve, Robert Raussendorf, and Joseph Emerson. The research presented in this thesis was undertaken thanks in part to funding from an NSERC-CGSD award, an Ontario Graduate Scholarship, by the University of Waterloo, and by the provincial and federal governments.

Thanks also to Kristine Boone, Annie Ray, Julia Amoros-Binefa, Clifford Plesha, Aditya Jain, Arnaud Carignan-Dugas, and many other friends and colleagues; the people I've had the honour and luck to connect with throughout this process have given me so much in terms of personal and intellectual growth and support, and I am grateful to have found them.

Last, but not least, I would like to thank my grandparents, Bill and Shirley Kilpatrick; I am very grateful to have had the opportunity to get closer to them during this final stage of my academic journey.

Table of Contents

List of Figures

List of Tables

Nomenclature

CNOT Quantum controlled-NOT operation, also written as $\hat{X}$.

IID Independent and identically distributed.

LRC Logical randomized compiling.

POVM Positive-operator-valued measure.

PVM Projection-valued measure.

QEC Quantum error correction.

QECC Quantum error correcting code.

RC Randomized compiling.

$\hat{A}$ Controlled-A gate. E.g. the CNOT (CX) gate would be denoted $\hat{X}$.

$\hat{\hat{A}}$ Controlled-controlled-A gate. E.g. the Toffoli (CCX) gate would be denoted $\hat{\hat{X}}$.

$\chi_a(b)$ When $a, b \in \mathbb{Z}_d^n$, $\chi_a(b) = e^{2a\cdot b\pi i/d}$. When $a, b \in \mathbb{PW}_{d,n}$, see eq. (1.29).

$\mathbb{CL}_{d.n}$ The Clifford group on n d-dimensional qudits.

C Confusion matrix. That is, matrix containing the probabilities of measuring and reporting different outcomes in a noisy measurement. Adding a bar to get $\bar{C}$ gives the symbol for a logical confusion matrix, which gives the probilities of measuring and reporting different error syndromes.

F The quantum Fourier transform operation. In the qubit case, this corresponds to the Hadamard gate.

$\mathbb{H}$, $\mathbb{K}$ Hilbert spaces, that is, spaces of bounded linear operators from $\mathbb{C}^a$ to itself.

$\mathbb{B}(\mathbb{H}, \mathbb{K})$ The space of bounded linear maps from $\mathbb{H}$ to $\mathbb{K}$. $\mathbb{B}(\mathbb{H}) = \mathbb{B}(\mathbb{H}, \mathbb{H})$.

$\mathbb{U}(\mathbb{H}, \mathbb{K})$ The set of isometries from $\mathbb{H}$ to $\mathbb{K}$. Recall that an isometry is a bounded linear operator U for which $U^\dagger U = I$. $\mathbb{B}(\mathbb{H}) = \mathbb{U}(\mathbb{H}, \mathbb{H})$.

$\text{Pos}(\mathbb{H})$ The set of bounded and positive semi-definite operators from $\mathbb{H}$ to itself.

I The identity matrix.

$|a\rangle$ The ath computational basis element when a is an integer (or a lower-case Roman letter representing an integer), a pure state otherwise.

$|a)$ Vectorization of a computational basis state $|a\rangle\langle a|$.

$|A\rangle\rangle$ Vectorization of an operator A.

$\mathbb{M}$ A measurement. Specified as either a set of projective operators or a set of maps.

$\bar{\mathcal{N}}_U(\mathcal{M}_R)$ Noisy encoded memory channel as a function of measurement map $\mathcal{M}_R = \mathcal{U}\mathcal{U}^\dagger\mathcal{R}^\dagger$ for an encoding U. $\bar{\mathcal{N}}_U(\mathcal{M}_R) = \mathcal{U}^\dagger\mathcal{R}^\dagger\mathcal{N}\mathcal{U}$. Adding a tilde, $\tilde{\bar{\mathcal{N}}}_U(\mathcal{M}_R)$, indicates noisy measurement with reported outcome R.

$\dot{\mathcal{N}}_U(R)$ Noisy encoded memory channel as a function of recovery operation R for an encoding U. $\dot{\mathcal{N}}_U(R) = \bar{\mathcal{N}}_U(\mathcal{M}_R) = \mathcal{U}^\dagger\mathcal{R}^\dagger\mathcal{N}\mathcal{U}$. Adding a tilde, $\tilde{\dot{\mathcal{N}}}_U(R)$, indicates noisy measurement with reported outcome R.

$\dot{\mathcal{N}}_U(\{(R_i, \mathcal{N}_i)\})$ Noisy encoded memory channel for an encoding U with multiple interleaved noise processes and QEC steps with noise and recovery at step i given by $\mathcal{N}_i$ and R_i, respectively. $\dot{\mathcal{N}}_U(\{(R_i, \mathcal{N}_i)\}) = \mathcal{U}^\dagger \left(\prod_{i=1}^m \mathcal{R}_i^\dagger \mathcal{N}_i\right) \mathcal{U}$.

π_i Projective operator. When i is a syndrome (pure error), they project onto the eigenspace corresponding to that syndrome (pure error). Typically i will be a lower-case letter for a syndrome and an upper-case letter for a pure error.

$\mathbb{S}$ The stabilizer group of a stabilizer code.

$\mathbb{G}$ The stabilizer generators of a stabilizer code.

$\mathbb{L}$ The set of logical Pauli (or Weyl) operators for an error correcting code.

$\mathbb{T}$ The set of pure errors for a stabilizer code.

$\mathbb{V}$ Set of recovery maps for an error correction. Typically, when referring to a single recovery operation, we will use $R \in \mathbb{V}$.

$\Gamma(A)$ Gadget implementation of a map A. A is commonly an operation, state preparation, or measurement. There is also an error correction gadget.

$\mathbb{GS}(U)$ The general stabilizer group - the set of operations that stabilize $\mathcal{U}$, where $\mathcal{U}$ is an encoding map, i.e. $\mathbb{GS}(U) = \{\mathcal{S} : \mathcal{SU} = \mathcal{U}\}$.

$\mathbb{GS}(U, \mathcal{N})$ The subset of $\mathbb{GS}(U)$ that commutes with $\mathcal{N}$.

$\mathbb{GL}(U)$ The general logical group - the set of logical operations for an encoding $\mathcal{U}$. I.e. the set of operations $\mathcal{L}$ such that there exists some $\bar{\mathcal{L}}$ for which $\mathcal{LU} = \mathcal{U}\bar{\mathcal{L}}$.

$\mathbb{GL}(U, \mathcal{N})$ The subset of $\mathbb{GL}(U)$ that commutes with $\mathcal{N}$.

$\Theta(\mathbb{M})$ Noisy implementation of $\mathbb{M}$. $\mathbb{M}$ is commonly a measurement, though this can also be used for an operation $\mathcal{A}$, as in $\Theta(\mathcal{A})$.

$\mathbb{W}_{d.n}$ The Weyl group on n d-dimensional qudits.

$\mathbb{PW}_{d,n}$ The projective Weyl group (i.e. the Weyl group modulo phases) on n d-dimensional qudits.

X The generalized X operator. For qubits, X is the Pauli matrix, σ_x.

Y The generalized Y operator. For qubits, Y is the Pauli matrix, σ_y.

Z The generalized Z operator. For qubits, Z is the Pauli matrix, σ_z.

$\mathbb{Z}_D$ The cyclic group of order D.

Chapter 1

Introduction

For the past several decades information technology has advanced rapidly, approximately following Moore's law, which states that the density of transistors that can be held on an integrated chip will double roughly every two years [5]. This rapid advancement has allowed computers to become an integral component of our society, driving research advancements in medicine, aerospace, cosmology, and many other vital fields. Unfortunately, in recent years this rate of doubling has slowed, and soon we will encounter a roadblock to the continuation of this technological evolution; if the size of transistors continues to decrease, quantum effects will come into play as transistors become progressively smaller and it will become impossible to continue to advance in this manner using solely classical technologies.

One shortcoming already present in traditional computation is that it is prohibitively expensive to simulate quantum systems with classical data. As such, it is difficult to design medications or simulate chemical reactions using classical computers. An alternative approach to simulating quantum systems was proposed by Richard Feynman in 1982 [6]. Feynman suggested that a quantum computer, which would manipulate quantum data rather than classical data, could significantly outperform its traditional classical counterpart. The advent of quantum computing would additionally address the issue of quantum effects coming into play as Moore's law continues; quantum computers would themselves rely on these effects. Since 1982, quantum technologies have advanced significantly, and researchers worldwide are now racing to get as many quantum bits (qubits - see section 1.2) as possible on a myriad of platforms.

There are a few significant barriers to the successful implementation of a scalable quantum computer. Among these barriers is the high sensitivity of quantum systems to interactions with the surrounding environment; any interaction of the quantum system with the envi-

ronment can introduce errors. This high sensitivity to noise informs the need for the work presented in this thesis.

Many techniques have been developed for characterizing, mitigating, tailoring, and correcting noise in quantum devices. Characterization techniques include methods for fully describing a state or operation, called tomography [7, 8], which are not scalable with device size. Other more scalable techniques are in wide use that quantify the performance for operations in terms of different figures of merit. Such techniques include randomized benchmarking [9], cycle benchmarking [10], extended randomized benchmarking [11], and, recently, a technique that gives a description of the noise in terms of probabilities of different errors called cycle error reconstruction [12]. For a thorough description and cross-comparison of each of these techniques and other error characterization techniques, see ref. [13].

Mitigation techniques sample from implementations with increased noise to get multiple data points about the expected output and extrapolate from these points to estimate the ideal output. These techniques include zero noise extrapolation [14, 15], probabilistic error cancellation [15], and noisy output extrapolation [16]. For a thorough overview of error mitigation techniques, see [17]. Mitigation techniques are expected to be useful in the near-term while devices are noisy but small enough that quantum error correction is not yet the best option.

Noise tailoring methods aim to modify implementations to enforce an effective noise of a given form. Notably, randomized compiling [18] is used to enforce an effective noise model on operations that is consistent with standard assumptions made of noise in studies of quantum error correction (QEC). However this technique was developed outside of the context of QEC and was not extended to non-destructive measurements [18]. The results presented in this thesis include protocols analogous to RC for these contexts [2, 4].

Quantum error correction improves performance by introducing redundancy to stored information by encoding in a larger space, then using redundancy created by this encoding to gain information about what error might have occurred without disturbing the stored information (in an ideal implementation). This retrieved information is referred to as an error syndrome, and is used to select a recovery operation that attempts to correct the error. QEC and other methods for fault-tolerant quantum computation are widely considered the most likely path to effective quantum computation as quantum computers scale up [19, 20]. However, modeling computations under QEC is difficult for the same reason that quantum computers are expected to be effective, namely that large quantum systems are computationally expensive to simulate. The difficulty of simulating the performance of quantum error correcting codes (QECCs) is compounded by the fact that the effective noise process depends

on the specific error syndromes that are observed as part of the correction process, where the number of possible syndromes typically grows exponentially with the number of qubits and rounds of computation. Consequently, little is known about the behaviour of generic noise in quantum error correcting codes and studies of quantum error correction make assumptions about the form of the noise.

Measurement refers to a step in quantum computation through which information about the state of the system is retrieved. This information comes in the form of a sample from a probability distribution over a set of outcomes. Measurement noise is particularly difficult to characterize in full and is very dependent on the type of platform used to implement the quantum computation (for example, measurement errors arise due to thermal excitations in superconducting qubits or difficulty in detecting dark states in ion qubits [17]) so it is difficult to select a model that is both accurate and consistent across platforms. Consistency across platforms is desirable to be able to use results from a given study across different implementations. As such, a noise model is typically assumed in which measurements are ideally performed and the outcomes are probabilistically misreported. This model is not well-motivated by physical phenomena, but provides a form of noise that is relatively easy to simulate. In studies of QEC, when measurement errors are considered, they are typically assumed to have an analogous form, wherein error syndromes are assumed to be probabilistically misreported. Protocols for fault-tolerant syndrome measurement also generally rely on this model.

In this thesis, we develop methods to enable more efficient simulation of noise in QECCs. We further propose protocols for noise tailoring in measurements and QECCs that enforce noise of a form consistent with typical assumptions made in studies thereof. We begin this introductory chapter by reviewing quantum states and methods for describing their evolution. Next, we describe error correcting codes and the stabilizer formalism, followed by an overview of fault tolerance. Finally, we review some methods for noise tailoring in quantum systems by giving an overview of twirling and randomized compiling. The background provided in this chapter is intended to provide the context to understand the results presented in subsequent chapters, as outlined in broad strokes below.

In chapter 2, we present new results which enable a significant reduction in the computational complexity of the simulation of QECCs. This work generalizes and extends the work presented in refs. [1, 21]. The reduction in computational complexity comes from showing how symmetries in QECCs can be leveraged to find relationships between the noise maps dependent on different syndrome measurement outcomes in a QECC. We show how to leverage symmetries in general quantum error correcting codes (where previous work restricted atten-

tion to a specific type of QECC called stabilizer codes), extend treatment to include noisy syndrome measurements, give symmetry conditions in the setting where multiple rounds of QEC are applied, and present a treatment for concatenated codes. We also present specific symmetries for the toric code, which is one popular example of a stabilizer code. This is the first instance where symmetry operations were leveraged in this way for a topological code.

In chapter 3, we propose *measurement randomized compiling*, a protocol which tailors measurement noise to a form consistent with standard assumptions. The resultant noise is easy to analyze and allows techniques developed under the typical model to be applied effectively. Measurement randomized compiling does not introduce significant overhead and ensures that entanglement between the measured system and the unmeasured system is removed by the effective measurement.

Finally, in chapter 4, we propose *logical randomized compiling* (LRC), a protocol which tailors noise in a QEC implementation to be consistent with standard assumptions on the noise. We include compilation methods for a universal gate set, state preparation, measurement, and syndrome extraction, which are the base elements needed for a quantum error corrected implementation of an arbitrary quantum computation. Like measurement randomized compiling, LRC can be implemented with little to no additional overhead.

1.1 Preliminaries: Notation for quantum states, channels, and operations

In this section, we lay out some preliminary notation needed to describe quantum computation (states, operations, measurements) mathematically. We use the following general notation throughout this thesis. For any Hilbert spaces $\mathbb{H}$ and $\mathbb{K}$, let $\mathbb{B}(\mathbb{H},\mathbb{K})$ denote the space of bounded linear maps from $\mathbb{H}$ to $\mathbb{K}$, $\mathrm{Pos}(\mathbb{H}) \subset \mathbb{B}(\mathbb{H},\mathbb{H})$ be the set of bounded and positive semi-definite operators from $\mathbb{H}$ to itself, and let $\mathbb{U}(\mathbb{H},\mathbb{K}) \subset \mathbb{B}(\mathbb{H},\mathbb{K})/\mathbb{U}(1)$ denote the set of isometries from $\mathbb{H}$ to $\mathbb{K}$, where we remove global phases from the set of isometries because they are unobservable in quantum mechanics. When $\mathbb{H} = \mathbb{K}$, we use the shorthands $\mathbb{B}(\mathbb{H}) = \mathbb{B}(\mathbb{H},\mathbb{H})$ and $\mathbb{U}(\mathbb{H}) = \mathbb{U}(\mathbb{H},\mathbb{H})$.

1.2 Quantum States

The most common base element of quantum information used in quantum computing is a quantum bit, or qubit. A qubit is a two-level quantum system that can be used to store information[1]. Alternatively, a d-level quantum system is referred to as a qudit. We define the computational basis for a single qubit, $\{|0\rangle, |1\rangle\}$,

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{1.1}$$

This convention for representing states for qubit systems can be trivially extended, e.g. by letting $|i\rangle = e_i$, where $\{e_i\}$ is the canonical basis, to span the space of a qudit. The $|0\rangle$ and $|1\rangle$ states typically correspond to a ground and excited state, respectively, in a physical system. An arbitrary pure single qubit state, $|\psi\rangle \in \mathbb{C}^2$, can be expressed in the computational basis as

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \tag{1.2}$$

where $\alpha, \beta \in \mathbb{C}$ are referred to as probability amplitudes, and $|\alpha|^2$ and $|\beta|^2$ give the probability of measuring 0 and 1, respectively, when the state is projected onto the computational basis (see section 1.7).

Alternatively, we can write a single qubit state in terms of a density matrix, ρ. A density matrix is a positive semi-definite operator with unit trace. For a pure state, $\rho = |\psi\rangle\langle\psi|$, where $\langle\psi|$ is the conjugate transpose of $|\psi\rangle$. A mixed state is a probabilistic combination of pure states $|\psi_k\rangle$,

$$\rho = \sum_k p_k |\psi_k\rangle\langle\psi_k|, \tag{1.3}$$

where $p_k \geq 0 \forall k$ and $\sum_k p_k = 1$. An arbitrary single-qubit state can be expressed in the form of a density matrix as a linear combination of basis elements of hermitian matrices in $\mathbb{C}^{2\times 2}$. An n-qubit state can be expressed as a density matrix in $\mathbb{C}^{2^n \times 2^n}$.

[1] More concretely, a qubit is a quantum system whose observables are the Pauli matrices and identity operator, but as these elements are yet to be introduced, the less specific description provided here is sufficient for understanding.

More generally, a qudit of dimension d is a quantum system whose state can be described by an element of the Hilbert space $\mathbb{H}_d = \mathbb{B}(\mathbb{C}^d)$, that is, the set of bounded linear operators from $\mathbb{C}^d$ to itself. It is sometimes convenient to represent quantum states as vectors of expansion coefficients. Relative to the canonical orthonormal basis $\{e_j : j \in \mathbb{Z}_{d^2}\}$ of $\mathbb{C}^{d^2}$ and a fixed trace-orthogonal basis $\{B_j : j \in \mathbb{Z}_{d^2}\}$ of $\mathbb{H}_d$, where $\mathbb{Z}_D$ is the cyclic group with order D, we define a vectorization map[2] to be the function

$$|\rangle\rangle : \mathbb{H}_d \to \mathbb{C}^{d^2} :: |M\rangle\rangle = \sum_{j \in \mathbb{Z}_{d^2}} \frac{\mathrm{Tr}\left(B_j^\dagger M\right)}{\sqrt{\mathrm{Tr}\left(B_j^\dagger B_j\right)}} e_j. \tag{1.4}$$

Note that we explicitly include the normalization factors rather than defining the map relative to a trace-orthonormal basis. We can translate expectation values into standard vector inner products by defining $\langle\langle \cdot| = |\cdot\rangle\rangle^\dagger$ and using the usual shorthand $\langle\langle A|B\rangle\rangle = \langle\langle A||B\rangle\rangle$, so that

$$\langle\langle A|B\rangle\rangle = \mathrm{Tr}\left(A^\dagger B\right) \tag{1.5}$$

for all $A, B \in \mathbb{H}_d$. For example, one can readily see that the normalization factors are such that

$$|B_j\rangle\rangle = \sqrt{\mathrm{Tr}\left(B_j^\dagger B_j\right)} e_j, \tag{1.6}$$

and so the trace-orthogonality relations between the B_j are equivalent to orthogonality relations between the $|B_j\rangle\rangle$. We can extend the above definitions to n qudits by setting $d \to d^n$, where we can construct bases for $\mathbb{H}_{d^n}$ by taking the n-fold tensor product of a basis for $\mathbb{H}_d$. We will frequently work with the n-qudit computational basis states and so define

$$|) : \mathbb{Z}_d^n \to \mathbb{C}^{d^2} :: |j) = |\, |j\rangle\langle j|\rangle\rangle. \tag{1.7}$$

Note that because the $|)$ stores a computational basis state, it can be treated as a classical dit or as a qudit with a restricted state space that mimics a classical dit. For clarity, note that if we conjugate a state ρ by a Kraus operator $L = M \otimes |k\rangle$, we get the state

$$L\rho L^\dagger = M\rho M^\dagger \otimes |k\rangle\langle k|\,, \tag{1.8}$$

where we have used the overhanging tensor factor of $|k\rangle$ in L to append an additional state space. Vectorizing the result then gives

$$|L\rho L^\dagger\rangle\rangle = |M\rho M^\dagger\rangle\rangle \otimes |k), \tag{1.9}$$

so that when we switch between vectorized and unvectorized statements we replace $|k)$ by $|k\rangle\langle k|$.

[2] There are many ways to write quantum states as vectorizations of the density matrix.

1.3 Quantum computation

Any quantum computation can be written as a sequence of three types of primitive operations that act on a quantum state. These operations are:

1. *state resets*;
2. *unitary operations*; and
3. *measurements*.

We call a sequence of these operations a circuit, and the map that a circuit describes it a channel.

Definition 1. *A quantum channel is a completely positive and trace-preserving (CPTP) linear map that maps quantum states to quantum states.*

A state can evolve via deliberate operations or from the effects of noise introduced by poorly implemented operations or interaction with its surroundings. The combined evolution due to deliberate operations and noise can often be represented by a channel, though in some cases, when information is lost to the environment, the map will be trace-decreasing rather than trace-preserving.

1.4 Representations of quantum channels

An arbitrary linear map, $\mathcal{A}$, acting on a density matrix, ρ, can be described in the Kraus formalism as follows.

$$\mathcal{A}[\rho] = \sum_k A_k \rho K_k^\dagger \tag{1.10}$$

where $\{A_k\}, \{K_k\} \subset \mathbb{B}(\mathbb{H}_d, \mathbb{H}_e)$ are known as Kraus operators [22]. If $\mathcal{A}$ is completely positive, $A_k = K_k \quad \forall k$. So for all channels, $A_k = K_k \forall k$. The selection of Kraus operators is non-unique for a given map. Throughout this document, calligraphic font will be used to denote maps and Roman font to denote operators.

It is also possible to express channels in a form that acts on a vectorized state via multiplication. We call these representations superoperators, and they are elements of

$\mathbb{B}(\mathbb{B}(\mathbb{H}_d), \mathbb{B}(\mathbb{H}_e))$. We now introduce a superoperator representation consistent with the vectorizated representation introduced in the previous section for states.

A quantum operation is a map from quantum states to quantum states and so is an element of the set $\mathbb{B}(\mathbb{H}_d, \mathbb{H}_e)$ of bounded linear operators from $\mathbb{H}_d$ to $\mathbb{H}_e$. As such, one can write any $\mathcal{Q} \in \mathbb{B}(\mathbb{H}_d, \mathbb{H}_e)$ in terms of vectorized operators as

$$\mathcal{Q} = \sum_j |A_j\rangle\!\rangle\langle\!\langle B_j| \tag{1.11}$$

for some $\{A_j\} \subset \mathbb{H}_e$ and $\{B_j\} \subset \mathbb{H}_d$. When $\mathcal{Q}$ is a completely positive map and $\mathbb{H}_d = \mathbb{B}(\mathbb{C}^d)$, there exist Kraus operators $\{K_k\} \subset \mathbb{C}^{e \times d}$ such that for any trace-orthogonal basis $\{B_j : j \in \mathbb{Z}_d^2\}$ of $\mathbb{H}_d$, we have

$$\mathcal{Q} = \sum_{\substack{k \\ B_j \in \mathrm{Basis}(\mathbb{H}_d)}} |K_k B_j K_k^\dagger\rangle\!\rangle\langle\!\langle B_j|. \tag{1.12}$$

As a special case, any (usually unitary) operator $A \in \mathbb{B}(\mathbb{H}_d, \mathbb{H}_e)$ defines a superoperator $\mathcal{A} \in \mathbb{B}(\mathbb{B}(\mathbb{H}_d), \mathbb{B}(\mathbb{H}_e))$ acting by conjugation, that is, $\mathcal{A}(\rho) = A\rho A^\dagger \ \forall \rho \in \mathbb{H}_d$. Note that the map $A \to \mathcal{A}$ is a bijection up to a global (and irrelevant) phase, so we abuse notation by interchanging A and $\mathcal{A}$. Also note that the maps $A \leftrightarrow \mathcal{A}$ preserve multiplication, that is, $AB \leftrightarrow \mathcal{AB}$. Given this relation, we get a channel $\mathcal{A}$ from an operator A by letting

$$\mathcal{A} = \sum_{B_j \in \mathrm{Basis}(\mathbb{H}_d)} |AB_jA^\dagger\rangle\!\rangle\langle\!\langle B_j|. \tag{1.13}$$

While unitary channels are fundamental to quantum computation, they are never implemented perfectly. In full generality, an imperfect implementation of a unitary channel can be a completely arbitrary channel. However, a common class of errors that are analyzed are stochastic channels [23], which admit a Kraus operator description $\{K_j\}$ such that $K_0 = pI$ for some $p \in (0, 1]$ and

$$\mathrm{Tr}\left(K_j^\dagger K_k\right) = \delta_{j,k}\mathrm{Tr}\left(K_j^\dagger K_j\right). \tag{1.14}$$

Throughout this document, when a channel acts on a state, the state will be enclosed in square braces. When a channel depends on some parameter(s), the parameter(s) will be enclosed in round braces. In the case where the representation allows channels to act as multiplication (as in vectorized states with superoper representations of channels, $|\cdot\rangle\!\rangle$), the square brackets will be omitted. For clarity we will use capital Roman letters (and π for projectors, to be introduced in section 1.7) to denote operators and calligraphic font to denote quantum operations (and Π for a map that applies a projector π).

1.4.1 The natural representation

We introduce here a specific state representation and corresponding superoperator that will prove useful for a proof later in this thesis.. This is treated as an aside as it is only used once and (outside of that proof) is not required for comprehension of the material presented in this thesis. We will refer the reader to this section when it arises. The natural representation uses the standard row vectorization map to get a vectorized state, that is $|\rho\rangle\rangle = \mathrm{vec}(\rho)$, and the corresponding superoperator for a channel $\mathcal{A}$ with Kraus operators $\{A_k\}$ is $\mathcal{A} = \sum_k A_k \otimes A_k^*$.

1.5 Circuit Representation

It is often convenient to represent quantum channels pictorially. This is achieved through the use of quantum circuit diagrams, in which qudits are denoted by horizontal lines and operations as images overlaid over the qudits. Figure 1.1 shows a basic example of a quantum circuit.

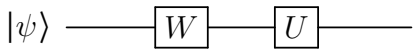
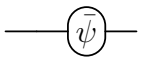
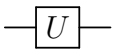


Figure 1.1: Basic example of a quantum circuit diagram in which an operator W is applied to the state $|\psi\rangle$, followed by an operator U. Time flows from left to right by convention.

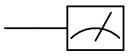
More concretely, the pictorial representations of primitive operations are given below:



(State reset)



(Unitary operation)



(Measurement)

Multi-qudit operations are represented by images overlaid over multiple qudits in a circuit, as demonstrated in fig. 1.2. A multi-qudit unitary operation is expressed as an operator in a larger space. If it acts on each qudit individually, it can be broken down as a tensor product of single qudit operators. Throughout this thesis, when we examine multi-qudit operators that act in this manner, we will often omit the tensor product symbol for notational simplicity. For example, a Pauli operation $X \otimes X \otimes I$ will be written as XXI.

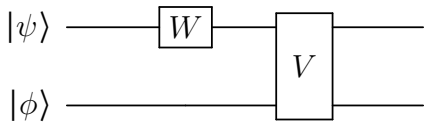


Figure 1.2: Basic example of a multi-qudit quantum circuit diagram in which a single qudit gate, W, acts on $|\psi\rangle$, followed by a multi-qudit gate, V, which acts on $W|\psi\rangle$ and $|\phi\rangle$.

In the case of encoded qudits, it is often convenient to denote a single logical qudit, which is encompassed by many physical qudits (see section 1.8), as a single horizontal line in a quantum circuit diagram.

1.6 Common Unitary Operations and Errors on Quantum States

In classical systems, after a system is sampled from an analog to a digital signal, the net effect of any errors will be that some bits flip (that is, some desired 0s will appear as 1s and vice versa); the exact mechanisms for data processing in classical systems will not be explored in this thesis. In quantum systems, states are more complex and thus, single-qubit errors and operations can take a much more complicated form. Let $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$. The quantum analog of a bit flip operator is denoted by X, and maps

$$|\phi\rangle \xrightarrow{X} \alpha|1\rangle + \beta|0\rangle, \tag{1.15}$$

or, equivalently, $X|\phi\rangle = \alpha|1\rangle + \beta|0\rangle$, where X is the Pauli matrix

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{1.16}$$

Quantum systems can also undergo phase flips, which map

$$|\phi\rangle \xrightarrow{Z} \alpha|0\rangle - \beta|1\rangle. \tag{1.17}$$

Phase flips are denoted Z, and the combination of a bit flip error and phase flip error is $Y = iXZ$, where Y and Z are Pauli matrices,

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{1.18}$$

Together with the identity matrix, I, the Pauli matrices form a complete basis over hermitian matrices in $\mathbb{C}^{2\times 2}$, so that any single qubit operation or error can be represented as a linear combination of these matrices. We define the single qubit Pauli group, $\mathbb{P} = \{I, X, Y, Z\}$, and the n-qubit Pauli group, $\mathbb{P}_n = \{I, X, Y, Z\}^{\otimes n}$.

1.6.1 Weyl operators

We now introduce the Weyl operators, which are a natural way to generalize the standard Pauli operators to a higher dimensional group. The single-qubit Pauli operators X and Z can be expressed in terms of bra-ket notation as

$$\begin{aligned} X &= \sum_{j\in\mathbb{Z}_2} |j \oplus 1\rangle\langle j| , \\ Z &= \sum_{j\in\mathbb{Z}_2} (-1)^j |j\rangle\langle j| . \end{aligned} \tag{1.19}$$

We can generalize these operators to qudits of dimension d by extending the sums in eq. (1.19) to to be over $\mathbb{Z}_d$ and replacing the -1 by a dth root of unity, that is

$$\begin{aligned} X_d &= \sum_{j\in\mathbb{Z}_d} |j \oplus 1\rangle\langle j| , \\ Z_d &= \sum_{j\in\mathbb{Z}_d} \chi_1(j) |j\rangle\langle j| , \end{aligned} \tag{1.20}$$

where we define

$$\chi_a(b) = \exp\left(\frac{2\pi i a b}{d}\right), \tag{1.21}$$

which is a dth root of unity for any $a, b \in \mathbb{Z}_d$. For convenience, we omit the subscripted d where it is implicit from context. We can define the Weyl group on single qudits in terms of generalized X and Z operators, $X(d)$ and $Z(d)$, respectively:

$$\mathbb{W}_d \equiv \{\chi_1(j) X(d)^x Z(d)^z : j \in \mathbb{Z}_d, x, z \in \mathbb{Z}_d\}. \tag{1.22}$$

In some cases, it is convenient to consider the phaseless Weyl operators. We therefore define the set of single-qudit projective Weyl operators, $\mathbb{PW}_{d,1} = \{X_d^x Z_d^z : x, z \in \mathbb{Z}_d\}$. For

convenience, we define the functions $x, z : \mathbb{PW}_{d,n} \to \mathbb{Z}_d^n$ such that for any $P \in \mathbb{PW}_{d,n}$, we have $P = X^{x(P)} Z^{z(P)}$. That is, $x(P)$ and $z(P)$ specify the powers of X and Z on each tensor factor to define P. The set of n-qudit Weyl operators is $\mathbb{PW}_{d,n} = \mathbb{PW}_{d,1}^{\otimes n}$, which can be written as

$$\mathbb{PW}_{d,n} = \{X_d^x Z_d^z : x, z \in \mathbb{Z}_d^n\}, \tag{1.23}$$

where we use the short-hand

$$A^a = \bigotimes_{i \in \mathbb{Z}_n} A^{a_i}, \tag{1.24}$$

for an operator A, a vector $a \in \mathbb{R}^n$. $\mathbb{PW}_{d,n}$ is a trace-othogonal basis for $\mathbb{H}_{d^n}$. Explicitly, for $x, z \in \mathbb{Z}_d^n$, we have

$$\begin{aligned} X_d^x &= \sum_{j \in \mathbb{Z}_d^n} |j \oplus x\rangle\langle j| \\ Z_d^z &= \sum_{j \in \mathbb{Z}_d^n} \chi_z(j)\, |j\rangle\langle j|\,, \end{aligned} \tag{1.25}$$

where for $a, b \in \mathbb{Z}_d^n$ we define

$$\chi_a(b) = \prod_{i \in \mathbb{Z}_n} \chi_{a_i}(b_i) = \exp\left(\frac{2\pi i a \cdot b}{d}\right). \tag{1.26}$$

Similarly, we define the n-qudit Weyl group, $\mathbb{W}_{d,n} \equiv \{\chi_1(j) X(d)^x Z(d)^z : j \in \mathbb{Z}_d, x, z \in \mathbb{Z}_d^n\}$.

Using the relation

$$Z^z X^x = \chi_x(z) X^x Z^z, \tag{1.27}$$

(which follows trivially from the definitions of X and Z), we can readily verify that the projective Weyl operators satisfy

$$PQP^\dagger = \chi_P^*(Q) Q, \tag{1.28}$$

where we define

$$\chi_P(Q) = \chi_{x(Q)}(z(P)) \chi_{x(P)}^*(z(Q)). \tag{1.29}$$

One useful feature of the functions $\chi_A(B)$ introduced above arises as a result of their relationship to the cyclic group. The cyclic group of order d is the set $\mathbb{Z}_d$ equipped with addition modulo d. The irreducible representations of $\mathbb{Z}_d$ are the d functions

$$\chi_a : \mathbb{Z}_d \to \mathbb{C} :: \chi_a(b) = \exp(2\pi i a b/d), \tag{1.30}$$

which are the same as the coefficients in eq. (1.21). We can extend the above representations to obtain the d^n irreducible representations of the Cartesian product $\mathbb{Z}_d^n$ by

$$\chi_a : \mathbb{Z}_d^n \to \mathbb{C} :: \chi_a(b) = \prod_{i \in \mathbb{Z}_n} \chi_{a_i}(b_i), \tag{1.31}$$

which match eq. (1.26). These representations are all characters and thus obey Schur's orthogonality relations [24],

$$\mathbb{E}_{a \in \mathbb{Z}_d^n} \left(\chi_j(a) \chi_k^*(a) \right) = \delta_{j,k} \forall j, k \in \mathbb{Z}_d^n, \tag{1.32}$$

where

$$\mathbb{E}_{s \in \mathbb{S}} \left(f(s) \right) = \frac{1}{|\mathbb{S}|} \sum_{s \in \mathbb{S}} f(s) \tag{1.33}$$

denotes the uniform average of a function f over a set $\mathbb{S}$. Schur's orthogonality relations will be used in several places in this thesis.

1.6.2 Clifford operators and universal gate sets

The n-qudit Clifford group on d-dimensional qudits, $\mathbb{CL}_{d,n}$, is a subgroup of the unitary group that maps non-identity Weyl operators to non-identity Weyl operators,

$$\mathbb{CL}_{d,n} = \{U \in U(2^n) | U \sigma U^\dagger \in \mathbb{W}_{d,n} \backslash I^{\otimes n} \ \forall \sigma \in \mathbb{W}_{d,n} \backslash I^{\otimes n}\} / U(1). \tag{1.34}$$

Importantly, Clifford gates on prime-dimensional qudits include entangling multi-qudit gates such as the controlled-X (often referred to as controlled-NOT or CNOT in 2-dimensions) gate, which will be defined in section 1.6.3. A universal gate set is one which can map from an arbitrary starting state to any state in the system. For a quantum computer to be fully general (i.e. be able to perform any quantum computation), it must be able to implement a universal gate set. One way of constructing a universal gate set is to add any non-Clifford

operation to the set of Clifford operations. A common choice of non-Clifford gates for qubit systems is the T gate,

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}. \tag{1.35}$$

Note that the T gate is the 4^{th} root of the Z gate. Analogously, for d-dimensional qudits, we can define a T gate as the 4^{th} root of the d-dimensional Z operator.

1.6.3 Other Common Unitary Operations on Quantum Channels

This section introduces three common multi-qubit operations used in quantum computing, as well as the single-qubit Hadamard operator and the generalization of each to the qudit case. The first 2-qubit gate is the controlled-NOT (CNOT) gate. The CNOT gate applies an X gate to one qubit when the other qubit is in the excited state, and does nothing otherwise. The qubit which determines whether X is applied is called the control qubit, and the qubit acted upon is the target qubit. The operator for a CNOT gate is

$$\hat{X} \equiv \text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{1.36}$$

For qudit systems with dimension d, a CNOT gate is given by

$$\hat{X} = \sum_{j \in \mathbb{Z}_d} |j\rangle\langle j| \otimes X^j, \tag{1.37}$$

and, more generally, a controlled-A gate for a unitary A is given by

$$\hat{A} = \sum_{j \in \mathbb{Z}_d} |j\rangle\langle j| \otimes A^j. \tag{1.38}$$

Another operation that is commonly used in quantum computing is the SWAP gate, which exchanges the position of two qubits. The SWAP gate on qubits is

$$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1.39}$$

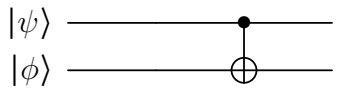


Figure 1.3: The circuit representation of a quantum CNOT gate, where $|\psi\rangle$ is the input state of the control qubit, and $|\phi\rangle$ is the input state of the target qubit. A controlled gate with a different action on the target qubit is represented by a similar symbol in which the circle on the target qubit is replaced by the symbol representing the desired operation.

An arbitrary permutation of qubits can be achieved by applying a sequence of SWAP gates.

The Toffoli gate, a.k.a. the CCX gate is

$$\hat{\hat{X}} = \sum_{i,j \in \mathbb{Z}_d} |i\rangle\langle i| \otimes |j\rangle\langle j| \otimes X^{ij}, \tag{1.40}$$

which applies an X to the target qubit when both control qubits are in the excited state, $|1\rangle$. For qudits, it applies a power of X based on the states of the control qudits. The circuit representation of the Toffoli gate is given in fig. 1.4.

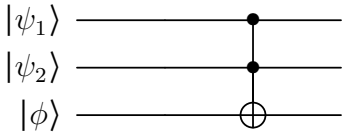


Figure 1.4: The circuit representation of a quantum Toffoli gate, where $|\psi_1\rangle$ and $|\psi_2\rangle$ are the input states of the control qubits, and $|\phi\rangle$ is the input state of the target.

Much like the CNOT gate, the controlled-controlled-X (CCX a.k.a. $\hat{\hat{X}}$) gate can be generalized to an analogous operation which applies powers of an arbitrary unitary A as

$$\hat{\hat{A}} = \sum_{i,j \in \mathbb{Z}_d} |i\rangle\langle i| \otimes |j\rangle\langle j| \otimes A^{ij}. \tag{1.41}$$

The single-qubit Hadamard transform is a gate which maps $X \leftrightarrow Z$, i.e. implements a basis change between the computational basis and the $\{|+\rangle, |-\rangle\}$ basis, and is given by

$$F = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{1.42}$$

A more generalized version is the quantum fourier transform for qudits, given by

$$F = \frac{1}{\sqrt{d}} \sum_{a,b \in \mathbb{Z}_d} \chi_a(b) \, |a\rangle\langle b| \, . \tag{1.43}$$

An identity that will be useful to keep in mind is that the $\hat{X}$ gate can be decomposed into a sequence of gates via

$$\hat{X} = (I \otimes F^\dagger)\hat{Z}(I \otimes F). \tag{1.44}$$

1.6.4 Modeling Quantum Error Channels

This section describes some common sources of errors in quantum systems, as well as the theory used to describe them. These concepts are integral to understanding how best to leverage a noisy quantum system, including how to correct errors effectively.

Definition 2. *A unitary channel is a channel whose action on a density matrix, ρ, can be written $\mathcal{U}[\rho] = U\rho U^\dagger$, for some unitary operator, U.*

A unitary operation can be imagined as a rotation of the state space. It is often convenient to picture the state space as a unit sphere construct called the Bloch sphere, see e.g. [25]. A noisy implementation of a map $\mathcal{A}$ will be denoted by $\Theta(\mathcal{A})$.

Operations on quantum states are often referred to as gates. One common source of error is over or under-rotation when gates are applied. This is generally modeled by a unitary rotation channel, which, for qubits, can be expressed as a single Kraus operator given by $e^{i\theta\vec{n}\cdot\vec{\sigma}}$, where $\vec{n}$ is a vector with $\vec{n}^\dagger \cdot \vec{n} = 1$, $\vec{\sigma} = (X, Y, Z)$, and

$$e^{i\theta\vec{n}\cdot\vec{\sigma}} = \cos\theta I + i\sin\theta(\vec{n}\cdot\vec{\sigma}). \tag{1.45}$$

Definition 3. *A mixed unitary channel, Φ, is a channel that can be expressed as a convex combination of unitary channels as follows*

$$\Phi[\rho] = \sum_k p(k) U_k \rho U_k^\dagger, \tag{1.46}$$

where p is a probability vector, $0 \le p(k) \le 1\ \forall k$ and $\sum_k p(k) = 1$.

Definition 4. *A stochastic Pauli (Weyl) channel is a mixed unitary channel with $U_k \in \mathbb{P}_n(\mathbb{PW}_{d,n})\ \forall k$.*

Stochastic Pauli channels are useful because they are easy to simulate and, because the Pauli matrices form a complete basis over Hermitian matrices in $\mathbb{C}^{2\times 2}$, a code can correct an arbitrary single-qubit error if and only if it can correct any single-qubit Pauli error.

Not all channels are stochastic Pauli channels. A common non-Pauli source of error in physical systems is the decay of an excited state to the ground state. This is modeled by an amplitude damping channel, in which an excited state decays from $|1\rangle$ to $|0\rangle$ with some probability p, while the ground state $|0\rangle$ remains unchanged. The amplitude damping channel can be described by the Kraus operators $\{A_i\}$ in eq. (1.47). The amplitude damping channel is not mixed unitary and does not leave the identity matrix invariant.

$$A_0 = \begin{bmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{bmatrix} \text{ and } A_1 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{bmatrix}. \tag{1.47}$$

It is possible to achieve a Pauli approximation of a channel via a process referred to as Pauli twirling [26]. This is achieved by taking the average result of conjugating a channel with Pauli operators. More generally, twirling is the process of taking the average channel under conjugation by a set of unitaries. For more detail, see section 1.11. Pauli twirling is achievable experimentally and is easily achieved in simulations and theory [26].

The depolarizing channel is a common example of a Pauli channel used to model noise; it describes a process in which the input state is replaced with the maximally mixed state with probability p, and left unchanged with probability $1-p$. The depolarizing channel can be described by the Kraus operators $A_0 = \sqrt{1-3p/4}I$, $A_1 = \sqrt{p/4}X$, $A_2 = \sqrt{p/4}Y$, and $A_3 = \sqrt{p/4}Z$. As with more general stochastic Pauli channels, it is possible to attain a depolarizing channel by twirling. The twirl which produces a depolarizing channel is over the Clifford group rather than the Pauli group.

When studying noise in a quantum system, it is often useful to be able to refer to the number of subsystems that an error acts on. We therefore define the *weight* of an operator.

Definition 5. *An operator has weight w if it acts non-trivially on w qubits.*

1.7 Measurement

Measurements are the only method available to retrieve information from a quantum system. However, any interaction with a quantum system changes it. In this section, we introduce

measurement operations, including an overview of mathematical descriptions of projection-valued measures, computational basis measurements amd indirect measurements.

A projection-valued measure (PVM) $\mathbb{M} \subset \mathrm{Pos}(\mathbb{H})$ is a set of orthogonal projectors that sum to the identity, that is, $\pi^2 = \pi$ for all $\pi \in \mathbb{M}$ and $\sum_{\pi\in\mathbb{M}} \pi = I_{\mathbb{H}}$ where $I_{\mathbb{H}}$ denotes the identity element of $\mathbb{B}(\mathbb{H})$. An ideal K-outcome projective measurement is described by a set of projectors $\mathbb{M} = \{\pi_k : k \in \mathbb{Z}_K\} \subset \mathbb{H}$. The most common type of measurement is a measurement in the computational basis, which projects onto computational basis states. For a single-qubit system, a computational basis measurement takes the form $\{|j\rangle\langle j| : j \in \mathbb{Z}_2\}$. For a computational basis measurement of a larger n-qudit system of d-dimensional qudits, the set would be $\mathbb{M}_{\mathrm{comp}} = \{|k\rangle\langle k| : k \in \mathbb{Z}_d^n\}$. When we consider a composite system, we may want to measure the ith subsystem. This can be represented by the Kraus operators

$$\mathbb{M}_{\mathrm{comp},i} = \{\sum_{j\in\mathbb{Z}_d^n : j_i = k} |j\rangle\langle j| : k \in \mathbb{Z}_d\}, \tag{1.48}$$

which, in the case $i = 0$ reduces to

$$\mathbb{M}_{\mathrm{comp},0} = \{|k\rangle\langle k| \otimes I_d^{\otimes n-1} : k \in \mathbb{Z}_d\}. \tag{1.49}$$

That is, the kth projector acts as $|k\rangle\langle k|$ on the ith subsystem and I on the rest of the subsystems.

When we apply a projective measurement whose projectors π_i admit a decomposition of an operator A as

$$A = \sum_i \lambda_i \pi_i, \tag{1.50}$$

where $\{\lambda_i\}$ are the eigenvalues of A, we say that we are measuring A. In this case, π_i is a projector onto the eigenspace of A corresponding to λ_i. If we measure A on some state, $|\psi\rangle$ we will observe some outcome λ_i, with probability $\langle\psi|\,\pi_i\,|\psi\rangle$, and the state after measurement is in the corresponding eigenspace. Specifically, the post-measurement state is

$$\frac{\pi_i|\psi\rangle}{\sqrt{\langle\psi|\pi_i|\psi\rangle}}. \tag{1.51}$$

The most common type of measurement is a measurement of the Z operator, which implements a computational basis measurement. We can see that measuring Z projects the state onto the computational basis since Z can be decomposed as

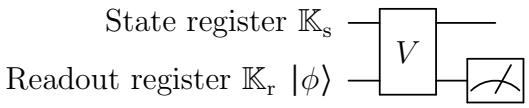
$$Z = \sum_{j\in\mathbb{Z}_d} \chi_1(j)\,|j\rangle\langle j|\,. \tag{1.52}$$

1.7.1 Indirect measurements

Sometimes, measurements are mediated through an additional readout space. We call such measurements indirect measurements. This allows us to construct measurements of operations which return less information than would be needed if we measured the state space directly to measure the same operations. For example, if we wanted to measure the operation ZZ on a 2-qubit space and could only measure the state space directly, we would measure each of the 2 qubits, giving us 2 bits of information about the state corresponding to the eigenspace of IZ and of ZI that the state space is in after each measurement. We would then use these two bits to calculate the eigenspace of ZZ. However, with an indirect measurement, it is possible to measure ZZ (or any single operation for which we can construct suitable projectors on the joint space) and retrieve only one bit of information, indicating which eigenspace of ZZ the state is in after measurement. Because we have collected less information, the state of the system is disturbed less; measuring ZI and then IZ collapses onto an eigenspace of ZI and then IZ, whereas measuring ZZ collapses onto an eigenspace of ZZ, leaving the state in a larger subspace of the state space than the double measurement allows and thereby retaining more quantum information. For example, the superposition of the state $|00\rangle + |11\rangle$ is destroyed if we measure two bits by measuring the qubits directly but preserved if we use an indirect ZZ measurement to measure one bit. To perform an indirect measurement on a state space $\mathbb{H}_\mathrm{s}$, we prepare the readout space $\mathbb{H}_\mathrm{r}$ in a state $|\phi\rangle$ and apply a unitary $V \in \mathbb{U}(\mathbb{H}_\mathrm{s} \otimes \mathbb{H}_\mathrm{r})$. Then, we measure $\mathbb{H}_\mathrm{r}$ via some measurement $\mathbb{M} \subset \mathbb{H}_r$. The operation applied for an indirect measurement then takes the form

$$(\mathcal{I} \otimes \mathbb{M})\mathcal{V} \tag{1.53}$$

and the circuit diagram for an indirect measurement is given in fig. 1.5.



(1.54)

Figure 1.5: Circuit diagram of an indirect measurement. Note that we include a measurement with no classical register; this is for ease of readability and we will follow this convention for the remainder of this subsection. In practice, a classical register is necessary to retrieve the classical information from a measurement.

An explicit form for a circuit which performs an indirect measurement of a Weyl operation $A \in \mathbb{W}(\mathbb{H}_s)$ is given in fig. 1.6.

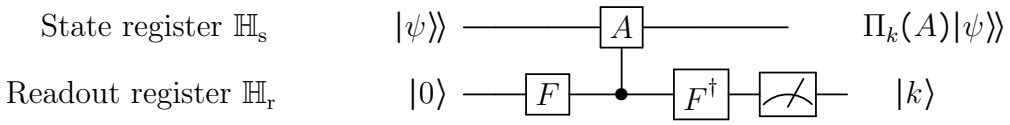


Figure 1.6: Circuit which performs an indirect measurement of a Weyl operation $A \in \mathbb{W}(\mathbb{H}_s)$, where the quantum Fourier transform is defined in eq. (1.43) and controlled-Weyl gates are defined in eq. (1.38). The measurement is performed in the computational basis.

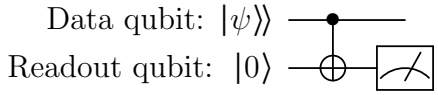


Figure 1.7: Circuit diagram of an indirect computational basis measurement.

We denote the indirect measurement of a Weyl operation $A \in \mathbb{W}(\mathbb{H}_s)$ as $\mathbb{M}_A$ so that

$$\mathbb{M}_A = (\mathcal{I} \otimes \mathbb{M}_{\text{comp}} \mathcal{F}^\dagger) \hat{\mathcal{A}} (\mathcal{I} \otimes \mathcal{F}). \tag{1.55}$$

The projective operators which describe this measurement are given by

$$\mathbb{M}_A = \{\pi_k(A) \otimes |k\rangle\langle k|\}, \tag{1.56}$$

where

$$\pi_k(A) = \frac{1}{d} \sum_{j \in \mathbb{Z}_d} \chi_k^*(j) A^j \tag{1.57}$$

projects onto the k^{th} eigenspace of A so that the final state on $\mathbb{H}_s$ is in the k^{th} eigenspace of A and the readout register is in $|k\rangle$ for $k \in \mathbb{Z}_d$.

Recalling eq. (1.44), an indirect computational basis measurement can be performed via the circuit given in fig. 1.7.

Recall that throughout this thesis, in the context of measurements, π will typically denote a projector and Π will be used as the map that applies π. Further, $\mathbb{M}$ will be used to denote a measurement as a set of operators or maps. In this notation, the ideal state after applying a PVM $\mathbb{M}$ to a system in the state ρ and observing outcome associated with projector π is

$$\rho' = \frac{\Pi[\rho]}{\text{Tr}\Pi[\rho]}, \tag{1.58}$$

where $\mathrm{Tr}\Pi[\rho]$ is the probability of observing the outcome $\pi \in \mathbb{M}$ conditioned on the system being in the state ρ.

In this thesis, we will often consider the average effect of a measurement, in which case we can treat measurements as a channel (more detail on this will be given in section 2.2.1 and we defer a more thorough discussion to that section). In this case, the channel that is applied when the measurement $\mathbb{M}$ is performed and the outcome is not recorded is

$$\mathcal{M} = \sum_{\substack{\pi_k \in \mathbb{M} \\ B \in \mathrm{Basis}(\mathbb{H})}} |\pi_k B \pi_k^\dagger\rangle\!\rangle\langle\!\langle B|. \tag{1.59}$$

1.7.2 Modeling noisy measurements

Studies which include noisy measurements typically model them as ideal projectors with probabilistically misreported outcomes. This model can be summarized by a *confusion matrix*, which contains the probability of measuring a given projector and reporting a given outcome under the model. The confusion matrix element in the i^{th} row and j^{th} column indicates the probability of having applied the ideal projector associated with the i^{th} outcome, given that the j^{th} outcome was reported. For the simplest example, consider a computational basis measurement on a single qubit. Because there are two possible outcomes, the confusion matrix will be 2×2, and would take the form given in fig. 1.8. The columns of a confusion matrix should sum to 1.

	$\|0\rangle$	$\|1\rangle$
$\langle 0\|$	$p(0\|0)$	$p(0\|1)$
$\langle 1\|$	$p(1\|0)$	$p(1\|1)$

Figure 1.8: A confusion matrix for a noisy measurement in the computational basis. Here, $p(A|B)$ is the probability of having measured an outcome A (i.e. the ideal projection onto A was applied) given a reported outcome B.

A noisy implementation of a measurement $\mathbb{M}$ will be denoted by $\Theta(\mathbb{M})$.

1.7.3 Weak measurements

We now include an overview of weak measurements for completeness. A weak measurement is an indirect measurement in which the unitary operation applied to the joint space prior

to measuring the readout space implements a weak coupling between the state space and the readout space. We therefore define a unitary operation on the joint space in terms of a Hamiltonian $H \in \mathbb{B}(\mathbb{K}_{\text{state}} \otimes \mathbb{K}_{\text{out}})$

$$V(t) = e^{-itH}, \tag{1.60}$$

where t is the interaction strength, and can be associated with the time for which the state is allowed to evolve under H. We assume since we want a weak interaction that t is small and perform a series expansion of V about $t = 0$:

$$V(t) = \sum_{n=0}^{\infty} \frac{(-itH)^n}{n!} \tag{1.61}$$

$$= I_{\mathbb{K}_{\text{state}}} \otimes I_{\mathbb{K}_{\text{state}}} - itH - \frac{1}{2}t^2H^2 - \frac{1}{6}it^3H^3 + \mathcal{O}(t^4). \tag{1.62}$$

Letting $H = S \otimes O$ for $S \in \mathbb{B}(\mathbb{K}_{\text{state}})$, $O \in \mathbb{B}(\mathbb{K}_{\text{out}})$, and recalling that a measurement on subspace $\mathbb{K}_{\text{out}}$ has Kraus operators of the form $I \otimes |k\rangle\langle k|$, we have Kraus operators $M_k \in \mathbb{B}(\mathbb{K}_{\text{state}})$ of the full weak measurement that take the form

$$M_k \approx I \langle k|\, \phi\rangle - itS \langle k|\, O\, |\phi\rangle - \frac{1}{2}t^2S^2 \langle k|\, O^2\, |\phi\rangle - \frac{1}{6}it^3S^3 \langle k|\, O^3\, |\phi\rangle\,, \tag{1.63}$$

where we have dropped higher order terms. The superoperator representation of such a measurement channel in terms of the Kraus operators is defined in the usual way (i.e. eq. (1.12)).

1.8 Error Correcting Codes

Errors are introduced into quantum systems when the system is not sufficiently isolated from the surrounding environment and when gates are not perfectly implemented. To correct these errors, it is necessary to use a protocol to reduce errors. This section reviews the basics of quantum error correcting codes, which form the backbone of one such method for reducing errors, namely quantum error correction.

1.8.1 Classical Repetition Code

In classical computing, hardware is sufficiently stable that errors due to interaction with the environment are rare. However, in situations where an error could be fatal (for example in an

aircraft), classical encodings are still used to protect computations and stored information. Classical encodings are also commonly used to reduce the number of bits used to transmit information. One common classical encoding used to protect against noise is the repetition code. The repetition code protects classical data by copying every bit of data onto additional bits to create redundancy. We distinguish encoded (logical) bits from physical bits by adding a bar, so a three bit encoding for a classical repetition code would be expressed as

$$0 \rightarrow \overline{0} = 000 \tag{1.64}$$

$$1 \rightarrow \overline{1} = 111. \tag{1.65}$$

For example, the string "010" would be encoded as "000111000". Errors are then corrected by measuring every physical bit in each logical bit and flipping any bit that doesn't agree with the majority. Introducing a bit flip error on the second last physical bit in the above example, we have "000111010". When we measure the third logical bit, we see that the second physical bit doesn't match the others and will accurately correct the error. If, however, two single bit errors were introduced in the same logical bit, e.g. "000111011", we would attempt to correct the wrong error so that the end result would be "000111111", thus introducing a logical fault, as this would then be decoded as "011" rather than the intended "010". However, this recovery method works the majority of the time because the error rate in classical systems is incredibly small [27][3].

1.8.2 Quantum Repetition Code

The classical error correction methods described in section 1.8.1 cannot be readily used in quantum systems because, by the no-cloning theorem [28], quantum states cannot be copied, so the classical encoding cannot be exactly replicated for a logical quantum state. Further, measuring a quantum state changes it, so observing the state directly to see errors directly is not advisable as it would result in a loss of encoded information, and errors are continuous, unlike errors in digitized classical data.

In quantum error correction, encoding creates redundancy in a similar way; data is protected using additional qudits, albeit in a different manner. Additionally, extra "readout" qudits are used to get information about errors in quantum error correcting codes via indirect measurements.

[3] This study reported one memory error on average in 41 hours on a node with 4GB of DRAM with no error correction in 2016.

Definition 6. *The set of codewords of a quantum error correcting code is the set of encoded computational basis states.*

The quantum repetition code is constructed by defining logical states $|\bar{0}\rangle = |000\rangle$ and $|\bar{1}\rangle = |111\rangle$. This encoding does not violate the no-cloning theorem because a general state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ maps to $|\bar{\psi}\rangle = \alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$, rather than $(\alpha|0\rangle + \beta|1\rangle)^{\otimes 3}$ and so the state is not copied in the encoding process. The encoding circuit for the 3 qubit repetition code is given in fig. 1.9.

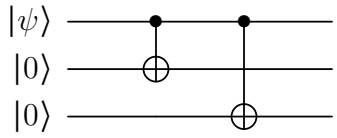


Figure 1.9: Encoding circuit for the quantum repetition code.

To detect bit flip errors on states encoded in the quantum repetition code, we can detect differences between any two pairs of physical qubits without directly measuring the encoded state as shown in fig. 1.10. An astute reader might recognize that this is similar to the indirect computational basis measurement circuit presented in fig. 1.7, which is a special case of fig. 1.6.

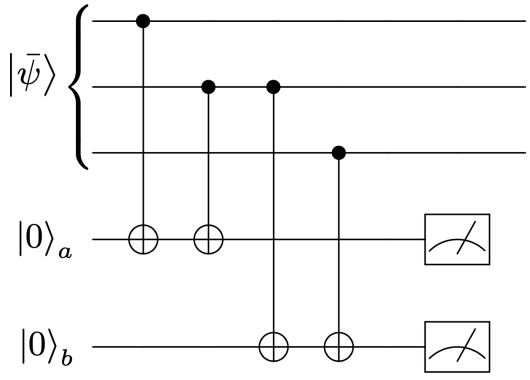


Figure 1.10: Circuit diagram that measures bit flip errors on the 3 qubit repetition code.

Measuring ancilla qubits a (b) tells us whether the first and second qubit (second and third) qubit in the encoded state match. The outcome of these measurements is a two bit string called a measurement syndrome, which provides insight into what errors might have occurred. Assuming that bit flip errors are the only source of noise in our system, we can

try to correct the noise based on the measured error syndrome. Error syndromes and the associated bit flip errors are listed in table 1.1.

Label	a	b	Errors
m_0	0	0	$\underline{I^{\otimes 3}}$, $X^{\otimes 3}$
m_1	0	1	$\underline{IIX}$, XXI
m_2	1	0	$\underline{XII}$, IXX
m_3	1	1	$\underline{IXI}$, XIX

Table 1.1: Syndromes and corresponding bit flip errors for the 3 bit repetition code.

Assuming a low probability of error (in a useful quantum system, this is a reasonable assumption), we generally choose to correct the underlined error associated with each syndrome in table 1.1; if each qubit has a low probability p of undergoing a bit flip error, then the underlined errors are significantly more likely to occur than the others. Correcting an error that did not occur can result in a logical error. For example, if an XXI error occurs and we measure the correct syndrome, $(0, 1)$, but we assume that IIX has occurred and apply IIX as a correction, we will then have the encoded state acted upon by XXX, thus producing a logical fault[4].

The indistinguishability of these errors is symptomatic of a pervasive issue in quantum error correction; the only information available to try to correct errors in a quantum error correcting code is which syndrome was measured, which could correspond to many different errors. This difficulty is well summarized by the Knill-Laflamme conditions for error correction, presented in ref. [20], which give necessary and sufficient conditions for a state to be recoverable after it has undergone some noise process.

The codewords of a code form an orthonormal basis of the codespace. In terms of the codewords, $\{|\overline{\psi}_i\rangle\} \in \mathbb{C}$, the first Knill-Laflamme condition for the correctability of a set of errors $\mathbb{E}$ is that the following two statements hold for all $E, F \in \mathbb{E}$ such that $E \neq F$:

1. $\langle\overline{\psi}_i| E^\dagger F |\overline{\psi}_i\rangle = \langle\overline{\psi}_j| E^\dagger F |\overline{\psi}_j\rangle$
2. $\langle\overline{\psi}_i| E^\dagger F |\overline{\psi}_j\rangle = 0$.

It should be noted that this condition applies for any orthonormal basis of the codespace, rather than just the set of codewords. The other conditions presented in [20] are equivalent and we restrict attention to this one for brevity.

[4] Applying the recovery operator introduces a logical fault because it plus the error maps $|\overline{0}\rangle \leftrightarrow |\overline{1}\rangle$.

We can construct a similar code which corrects phase errors by replacing the CNOT gates in fig. 1.9 with controlled-Z gates and applying Hadamard gates to the ancilla qubits before and after the CNOTs in the syndrome measurement.

1.8.3 Stabilizer Codes and their Error Correction Protocols

Stabilizer codes are a subclass of quantum error correcting codes for which the encoded space is specified as the space stabilized by an Abelian group $\mathbb{S}$, called the stabilizer group. That is, the codespace is the mutual +1 eigenspace of the stabilizers,

$$\{|\psi\rangle : S|\psi\rangle = |\psi\rangle \; \forall S \in \mathbb{S}\}. \tag{1.66}$$

The quantum repetition code introduced in the previous section is an example of a stabilizer code. Typically, the stabilizers are chosen to be phase multiples of Weyl operators, and for the code space to not be empty we require $\mathbb{S}$ to contain no nontrivial phase multiples of the identity. However, we will work with the operator space of the codespace, and so we define the codespace to be

$$\mathbb{O} = \{\rho : S\rho S^\dagger = \rho \forall S \in \mathbb{S}\}. \tag{1.67}$$

The projector onto the codespace is then

$$\pi_I = \mathbb{E}_{S\in\mathbb{S}}(S). \tag{1.68}$$

The stabilizer group for a code which encodes k logical qudits of dimension d in n physical qudits of dimension d has order $|\mathbb{S}| = d^{n-k}$ and can be specified by a minimal generating set $\mathbb{G}$ such that $\langle\mathbb{G}\rangle = \mathbb{S}$, where $\langle\mathbb{G}\rangle$ denotes the group generated by $\mathbb{G}$. The simultaneous eigenspaces of $\mathbb{G}$ form a set of mutually orthogonal subspaces of the physical space used to encode a logical state. We refer to the these eigenspaces as the cospaces of the code. There is a group $\mathbb{T} \subsetneq \mathbb{PW}_{d,n}$ called pure errors such that each pure error $T \in \mathbb{T}$ maps a state on one eigenspace to a state on a distinct eigenspace. Thus, we define the cospace $\mathbb{O}_T$ of the code to be

$$\mathbb{O}_T = \{T\rho T^\dagger : \rho \in \mathbb{O}\}. \tag{1.69}$$

Thus, the projector onto $\mathbb{O}_T$ is π_T where

$$\pi_T = T\pi_I T^\dagger = \mathbb{E}_{S\in\mathbb{S}}(\chi_T^*(S)S), \tag{1.70}$$

and the final equality follows from eq. (1.28). Each stabilizer code has logical Weyl operations $\mathbb{L}$ for which $[L, S] = [L, T] = 0 \forall L \in \mathbb{L}, T \in \mathbb{T}, S \in \mathbb{S}$ and which have the same commutation relations as the corresponding physical operations. Analogous to our notation for logical states, we denote a logical operation by putting a bar above, that is, a logical version of an operation L would be denoted by $\bar{L}$. As one might expect, a logical operation $\bar{L}$ acts on the codespace in the same manner as its physical counterpart, L, would act on an unencoded state space.

The existence of mutually orthogonal cospaces allows a us to distinguish between sets of errors that might have occurred by measuring the stabilizer generators of the code. Each pure error (and therefore cospace) is associated with a ditstring called the error syndrome. We define the syndrome space to be the union of the cospaces,

$$\cup_{T \in \mathbb{T}} \mathbb{O}_T. \tag{1.71}$$

Crucially, the syndrome space is a strict subspace of the whole physical space because superpositions of states in different cospaces are not in the syndrome space.

Relating this back to the repetition code introduced in section 1.8.2, the stabilizer generators of the repetition code are ZZI and IZZ. Figure 1.10 measures the stabilizer generators using an indirect measurement (recall from eq. (1.44) that a $\hat{X}$ operation can be expressed as a $\hat{Z}$ operation with Fourier transforms), with ZZI measured using the ancilla qubit a and IZZ measured with qubit b. The measurement outcomes specify which cospace of the code the post-measurement state is in, and are related to the commutation relations of the measured stabilizer generator with the error that has occurred (and, equivalently, the corresponding pure error). The logical Weyl (in this case Pauli) operators for the 3-qubit repetition code are $\bar{I} = III$, $\bar{X} = XXX$, $\bar{Y} = -YYY$, and $\bar{Z} = ZZZ$.

As part of the error correction process, a recovery operation R_T associated with the measured cospace is typically applied to map the state back to the codespace. The selection of effective recovery operations is imperative to the successful implementation of an error correcting code as selecting suboptimal recovery operations can introduce a logical fault, as we showed in section 1.8.2 for the repetition code.

The notion of code distance in quantum error correction is analogous to that used in classical error correcting codes, and there are many equivalent definitions. We will use the definition provided below herein.

Definition 7. *The distance of a quantum error correcting code is the minimum weight of an operator which maps one codeword to another.*

A QECC which encodes k logical qubits in n physical qubits and has distance d is denoted as an $[\![n, k, d]\!]$ code. An $[\![n, k, d]\!]$ code can correct the set of t-qubit errors, where $t \leq \lfloor (d-1)/2 \rfloor$[29]. When the distance is unknown or unspecified, a code is often referred to as an (n, k) code.

QECCs are often used in concatenation schemes to increase code distance. This is accomplished by re-encoding each physical qubit in an encoding with another (or the same) QECC, recursively. Re-encoding with the same QECC at each level of concatenation is often used to find thresholds for the performance of a given QECC [30–32]. It is worth noting that concatenating using different QECCs has been used to get a universal set of fault-tolerant gates on the full encoded state [33].

1.8.4 Popular Examples of Stabilizer Codes

The $[\![5, 1, 3]\!]$ code is the smallest code that can correct any single-qubit error. The stabilizer generators of the 5 qubit code are given in table 1.2. This code is referred to as a "perfect" code because each single qubit Pauli error (plus the trivial error) is associated with a unique syndrome and there are no syndromes leftover.

X	Z	Z	X	I
I	X	Z	Z	X
X	I	X	Z	Z
Z	X	I	X	Z

Table 1.2: Stabilizer generators for the $[\![5, 1, 3]\!]$ code.

The 9-qubit Shor code is created by concatenating the 3 qubit repetition (bit flip) code and the 3 qubit phase flip code to form a code which can correct all single qubit bit flip or phase flip errors. The Shor code is specified by the generators given in table 1.3.

Z	Z	I	I	I	I	I	I	I
I	Z	Z	I	I	I	I	I	I
I	I	I	Z	Z	I	I	I	I
I	I	I	I	Z	Z	I	I	I
I	I	I	I	I	I	Z	Z	I
I	I	I	I	I	I	I	Z	Z
X	X	X	X	X	X	I	I	I
I	I	I	X	X	X	X	X	X

Table 1.3: Stabilizer generators for the 9 qubit Shor code.

Calderbank-Steane-Shor (CSS) codes are created by mapping the parity check matrix of classical codes[5] to Pauli operators as follows: half of the stabilizer generators are constructed by taking each row of the parity check matrix for a classical code and replacing every 0 with I and every 1 with X. To construct the remaining stabilizer generators, this process is repeated, using a code that is the dual of the first code, but with $0 \to I$ and $1 \to Z$. The Steane code is an example of a CSS code, and is constructed from the classical 7 bit Hamming code, with stabilizer generators given in table 1.4. There are many other CSS codes; the $[\![15, 1, 3]\!]$ Reed-Muller is one of the more popular examples, and one of the larger codes commonly studied.

Z	Z	Z	Z	I	I	I
Z	Z	I	I	Z	Z	I
Z	I	Z	I	Z	I	Z
X	X	X	X	I	I	I
X	X	I	I	X	X	I
X	I	X	I	X	I	X

Table 1.4: Stabilizer generators for the 7 qubit Steane code.

For each of the 3-, 5-, 7-, and 9-qubit codes introduced above, the logical Pauli operators are specified by $\bar{X} = X^{\otimes n}$ and $\bar{Z} = Z^{\otimes n}$, with $\bar{Y} = i\bar{X}\bar{Z}$.

Some of the more popular stabilizer codes are motivated by topology. In particular, some surface codes are quite prominent for implementations because a planar chip layout lends

[5] We omit an introduction to classical error correcting codes for brevity; for a thorough introduction, see [34].

itself to a planar code construction. Often, topological codes allow stabilizer generators to be measured by low-weight local operations. In this thesis, we will restrict attention to a single type of topological code: the toric code. Unlike the previous examples of popular stabilizer codes, the toric code stores 2 logical qubits. As such, there are two of each of the single-qubit logical Pauli operations, and 2-qubit operations can be implemented on a single code block. Another difference is that "the" toric code is actually a class of codes, where an individual code is specified by a number of rows and columns, from which the number of physical qubits can be calculated. The distance of the code depends on which toric code is in use, and as usual corresponds to the lowest-weight non-trivial logical operator. We defer a complete description of stabilizers and logical operations to section 2.5.2 as it will be convenient to have the definition at hand when deriving the results in that section.

1.9 Quantifying Errors

In any computation, it is important to have some figure of merit which quantifies the reliability of the result; several methods have been developed to analyse the severity of noise in quantum systems.

Some specific noise channels have a form that lends itself easily to defining an error rate. A stochastic Pauli channel, for example, has some probability of applying a non-identity operation; that probability could easily be viewed as the error rate for that channel. For a channel which applies a unitary rotation, by contrast, it is not immediately obvious what figure of merit should be used to quantify the severity of the error.

The average gate infidelity[6] to the identity (hereafter infidelity) is one figure of merit developed to quantify error rate in quantum systems. The average gate infidelity is often referred to as the average error rate. It is the most commonly used figure of merit by experimentalists because it can be estimated efficiently in physical systems via randomized benchmarking (see ref. [9]). The infidelity, $r(\Phi)$, of a channel Φ, to the identity channel is given by eq. (1.72).

$$r(\Phi) = 1 - \int \langle\psi| \Phi(|\psi\rangle\langle\psi|) |\psi\rangle \, d\psi, \tag{1.72}$$

[6] We define infidelity as one minus the fidelity.

where the integration is over the uniform Haar measure on pure states. In terms of the Kraus operators, $\{A_k\}$, of the channel, Φ acting on a m^2-dimensional state space, the infidelity is[7] [26, 35]

$$r(\Phi) = \frac{\sum_k |\mathrm{Tr}(A_k)|^2 + m}{m(m+2)} \tag{1.73}$$

The diamond distance, $\diamond(\cdot)$, is another common figure of merit for error quantification, and is commonly used by theoreticians as a worst-case error analysis. It is defined relative to the identity channel, $\mathcal{I}$, by eq. (1.74),

$$\diamond(\Phi) = \sup_\psi \frac{1}{2} ||(\Phi \otimes \mathcal{I}_m - \mathcal{I}_{m^2})(\psi)||_1, \tag{1.74}$$

where the supremum is over all m^2-dimensional pure states to account for the effect of the channel on an entangled state.

The diamond distance is related to the infidelity by [36, 37]

$$\frac{m+1}{q} r(\Phi) \leq \diamond(\Phi) \leq \sqrt{m(m+1)}\sqrt{r(\Phi)}, \tag{1.75}$$

with

$$\diamond(\mathcal{P}) = \frac{m+1}{m} r(\mathcal{P}) \tag{1.76}$$

for Pauli noise $\mathcal{P}$ [38]. For unitary rotations, $\mathcal{U}$, $\diamond(\mathcal{U})$ is proportional to $\sqrt{r(\mathcal{U})}$, but does not necessarily saturate the upper bound in eq. (1.75)[39].

Both the fidelity and diamond distance extend easily to characterizing the severity of logical errors. However, comparing physical and logical error rates with these two figures of merit demonstrates some significant discrepancies in their characterization; the logical fidelity at the third level of concatenation arising from depolarizing vs. a rotation about the Z-axis differ by orders of magnitude when each has the same physical noise rate but one is characterized by the diamond distance and the other by the fidelity[40]. Figure 1.11 shows results from [40] comparing physical diamond distance and physical fidelity to logical fidelity

[7] In the interest of brevity, we omit the details of this derivation.

for randomly generated CPTP maps. Notice that the plots highlight that the depolarizing noise has the best logical fidelity when compared to the physical diamond distance, and the worst logical fidelity when compared to the physical fidelity. Conversely, Z rotation noise has the best logical fidelity when compared to the physical fidelity and the worst logical fidelity when compared to the physical diamond distance. It is thus difficult to predict the logical error rate (or even predict whether one type of noise will result in worse or better logical performance than another) in terms of a single figure of merit that quantifies the physical error rate.

Recent work, [41], developed a reliable way to predict the logical error rate in terms of the physical error rate for concatenated codes under randomized compiling (see section 1.12). However, we still do not have reliable methods for predicting the logical error rate for codes outside of this setting.

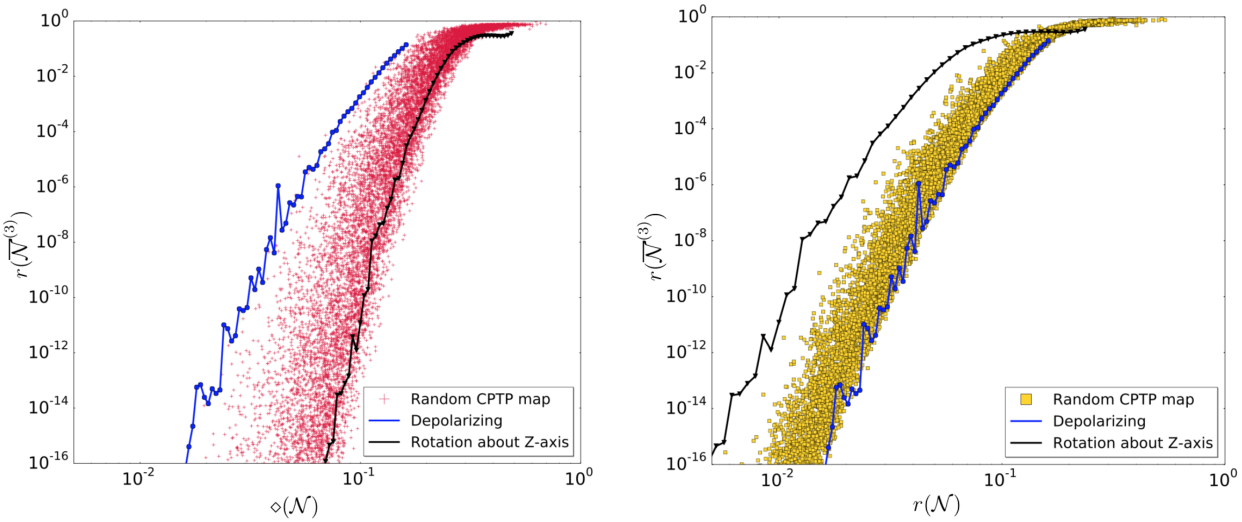


Figure 1.11: Figure reproduced from [40, Figure 1]. ©IOP Publishing. Reproduced with permission. All rights reserved. Plots of average logical fidelity for an encoded noisy state at the third level of concatenation, $r(\overline{\mathcal{N}}^{(3)})$, in the Steane code entangled with a perfect single qubit state as a function of physical diamond distance, $\diamond(\mathcal{N})$, (left) and physical infidelity, $r(\mathcal{N})$, (right) when recovery operations are selected to maximize fidelity with the noiseless state. Each point on each plot corresponds to a random CPTP map, which acts on each physical qubit comprising the state encoded in the Steane code. The blue lines show the behaviour of depolarizing noise, while the black show the scaling of errors for a coherent rotation about the Z axis.

1.10 Encoded implementation

When computations are implemented in an error corrected code, we refer to that implementation as an encoded implementation. The goal of an encoded implementation is to encode the operations in such a way that some errors can be detected by performing syndrome measurements and corrected between operations. We consider syndrome measurements as a fourth type of primitive operation for encoded implementations. Each of the encoded primitives (encoded state resets, unitary operations, measurements, and syndrome measurements) are themselves a sequence of the three primitive operations applied on the physical space.

We refer to an encoded version of a single operation as a *gadget*. A gadget is then a sequence of primitive operations whose composition acts on the codespace in the desired way. To implement an error-corrected circuit, each element of the unencoded circuit is replaced by the corresponding gadget. Additionally, error correction gadgets are added (e.g., between each gadget arising from encoding the circuit) that consist of a syndrome measurement followed by a recovery operator to correct the effect of any errors that occurred. An error corrected implementation is successful if it outperforms the unencoded implementation (as evaluated by some figure of merit). A mathematical description of encoded implementations follows.

Recall that a quantum circuit consists of a sequence of m operations $\mathcal{Q}_1, \ldots, \mathcal{Q}_m \in \mathbb{B}(\mathbb{H})$ on some physical space $\mathbb{H} = \mathbb{B}(\mathbb{C}^d)$. In a typical computation, $\mathcal{Q}_1$ resets the entire system to a fixed state and $\mathcal{Q}_m$ typically performs a measurement to return a ditstring result. The complete unencoded quantum computation is then

$$\prod_{j=m\to 1} \mathcal{Q}_j. \tag{1.77}$$

To implement these operations, we perform some physical operations that are intended to produce the desired map. That is, we have some noisy implementation map $\Theta : \mathbb{B}(\mathbb{H}) \to \mathbb{B}(\mathbb{H})$ such that the actual implemented map is

$$\prod_{j=m\to 1} \Theta(\mathcal{Q}_j). \tag{1.78}$$

The implementation map Θ is generally a non-linear function[8] (i.e., it is not an element of $\mathbb{B}(\mathbb{B}(\mathbb{H}))$) and is also not a representation of the unitary group since the implementation

[8] An implementation map can be non-linear because e.g. it could contain noise coming from a unitary operation acting on the joint system-environment state, in which case the evolution of the system can be non-linear.

of each ideal operation must be compiled into a sequence of calibrated operations, each of which is imperfectly implemented.

To obtain an improved implementation of a circuit, we implement quantum error correction protocols to embed the quantum space $\mathbb{H}$ into a larger space $\mathbb{K} = \mathbb{B}(\mathbb{C}^e)$ for some integer $e > d$. That is, we have some gadget map $\Gamma : \mathbb{B}(\mathbb{H}) \to \mathbb{B}(\mathbb{K})$, which will again not be linear, such that

$$\prod_{j=m\to 1} \Gamma(\mathcal{Q}_j). \tag{1.79}$$

acts on the larger system. The error correction protocol is successful if eq. (1.77) is closer to eq. (1.79) than eq. (1.78) under some appropriate figure of merit that depends upon the application. We can have multiple gadget maps and concatenate them appropriately to further improve performance at the cost of requiring a larger quantum system and longer processing time.

Encoded operations (also referred to as logical operations) act on encoded states, which are elements of a codespace $\mathbb{O} \subset \mathbb{K}$ that is isomorphic to $\mathbb{H}$, via gadgets. Specifically, for each operation A, a gadget map $\Gamma(\mathcal{A})$ returns a sequence of $q(A)$ physical operations $A_1, \ldots, A_{q(A)} \in \mathbb{B}(\mathbb{K})$ such that

$$\Gamma(\mathcal{A}) = \prod_{k=q(A)\to 1} \Theta(\mathcal{A}_k) \tag{1.80}$$

ideally performs the operation $\bar{A}$ on the encoded space $\mathbb{O}$. When operations are performed non-ideally, the effect of such an implementation can map encoded states outside of the codespace.

A gadget implementation of an operation A is ideal, $\Gamma(\mathcal{A}) = \Gamma_{\text{id}}(\mathcal{A}) \equiv \bar{\mathcal{A}}$, if the action of the gadget map on the codespace $\mathbb{O}$ is isomorphic to the ideal action $\mathcal{A}$ of A on the unencoded space $\mathbb{H}$. For convenience, when implementations are ideal, we will denote the action of such an implementation by the channel it implements, e.g. $\Gamma_{\text{id}}(\mathcal{A}) = \bar{\mathcal{A}}$. For a gadget map to be a fault-tolerant encoding, it must prevent the spread of errors beyond the ability of the code to correct [19]. See section 1.10.2 for more detail.

Because implementations of gadgets are noisy, we want to be able to correct errors that are introduced during the computation. We therefore introduce an error correction gadget, which leverages the redundancy introduced by encoding in a larger physical space to correct errors. The error correction gadget is a function of a set of recovery maps $\mathbb{V}$ applied to implement the

correction, as well as a measurement outcome T which indicates which recovery to apply. A QEC gadget is then denoted by $\Gamma_{\text{QEC}}(T, \mathbb{V})$. Because errors can be introduced by any noisy gadget, QEC gadgets are typically interleaved between every other gadget. An encoding of a circuit therefore typically takes the form

$$\Gamma(\mathbb{M}) \prod_{j=m-1\to 0} \Gamma_{\text{QEC}}(T_j, \mathbb{V}_j)\Gamma(\mathcal{A}_j), \tag{1.81}$$

where we terminate the computation with a measurement gadget $\Gamma(\mathbb{M})$ and allow the recovery sets $\mathbb{V}_j$ to depend on j rather than only T so that we can correct the different errors that are introduced by the different logical operations using distinct recovery maps. We will break down the form of a QEC gadget for Weyl stabilizer codes in section 1.10.1.

For circuit diagrams throughout this thesis, horizontal solid lines denote a logical system, dotted lines denote a physical system, and double lines denote a classical system. When the system could be either encoded or unencoded, the convention will be to use single solid lines. The first example of a circuit diagram requiring such a distinction is shown in the following section, in fig. 1.12, where the readout registers are unencoded, and an encoded register and classical register are also depicted.

1.10.1 QEC Gadgets

Because errors in a gadget implementation can map part or all of the encoded state out of the codespace $\mathbb{O}$, the first step in the quantum error correction gadget $\Gamma_{\text{QEC}}(T_{\text{out}}, \mathbb{V})$ is to perform a syndrome extraction which (ideally) projects the state onto a single cospace $\mathbb{O}_{T_{out}}$ via $\Pi_{T_{\text{out}}}$. Then a recovery operation $R_{T_{\text{out}}}$ from a predetermined set $\mathbb{V}$ is applied to the encoded space, which ideally maps the state from cospace $\mathbb{O}_{T_{\text{out}}}$ to the codespace $\mathbb{O}$. Figure 1.12 shows a circuit diagram of a QEC gadget, in which we include a classical register to show how the recovery operation is conditioned on the classical output of the syndrome measurement. For convenience, we assume that the readout and classical registers begin in their respective ground states and use an $\text{add}(0, s_{T_{\text{out}}}) = s_{T_{\text{out}}}$ function to write the syndrome to the classical register. This assumption does not remove generality, as modifying the classically controlled recovery allows us to permute the syndrome labels at our discretion.

Omitting the readout and classical spaces as the full map leaves them invariant, the effect of an ideal QEC gadget on the encoded register is

$$\Gamma_{\text{ideal, QEC}}(T_{\text{out}}, \mathbb{V}) = \mathcal{R}_{T_{\text{out}}}\Pi_{T_{\text{out}}}. \tag{1.82}$$

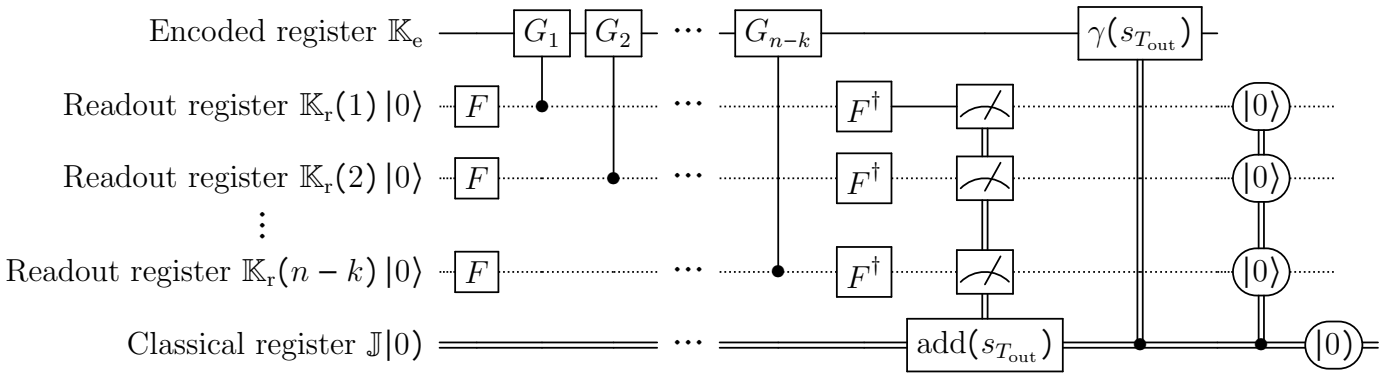


Figure 1.12: Quantum error correction gadget. At the end of the syndrome extraction circuit but prior to measuring the readout space (after the inverse Fourier transforms), the encoded space is in a superposition of the cospaces of the code, $\mathbb{O}_T$, each with a corresponding label state on the readout space, $|s_T\rangle$. The subsystem measurement of the readout space ideally projects onto a single syndrome $s_{T_{\text{out}}}$. To construct a QEC gadget, we follow the syndrome extraction by a classically conditioned unitary which applies recovery operations. State resets are applied on the readout register based on the state of the classical register, and the final step is to reset the classical register. Note that the reset is depicted as depending on the classical register but may be implemented differently in practice. Resets are included at the end of the circuit as the registers used for readout are often reused in the next round of syndrome extraction and error correction.

1.10.2 Fault Tolerance

While we have thus far mostly omitted an explicit discussion of quantum error correction protocols on multiple code blocks for simplicity, it is generally the case that a computation will require multiple logical qudits. In such a setting, with multi-qudit gates acting between physical qudits in different code blocks, it quickly becomes crucial to have methods in place to prevent the spread of errors. As an example of how errors might spread on a simple multi-qubit physical gate, consider the $\hat{X}$ operation. If an X error occurs on the control qubit prior to the application of $\hat{X}$, that error will spread, resulting in an X error on both the target and control qubits after the $\hat{X}$. Similarly, if a Z error occurs on the target qubit prior to the $\hat{X}$, the final state will have a Z error on both qubits.

Fault tolerance is a property of implementations (or gadgets, including error correction gadgets) which indicates that that implementation does not permit errors that are correctable

for that code to spread in a way that introduces errors that are not correctable by that code on any code block. In this way, a fault-tolerant implementation ensures that if errors introduced by any step of a computation is correctable, subsequent operations won't cause that error to become uncorrectable.

One class of fault-tolerant gate gadgets are transversal gates. A transversal gate is one in which each physical operation in the implementation acts only on a single physical qudit in a given code block, and no two physical operations act on the same physical qudit. More formally, a gadget $\Gamma(\cdot)$ implementation of an operation $\mathcal{A}$ for blocks of an n-qudit code is transversal if

$$\Gamma(\mathcal{A}) = \circ_{j=1}^{n} \mathcal{G}_i, \tag{1.83}$$

where each $\mathcal{G}_i$ acts non-trivially only on the i^{th} qudit in each code block. Note that while we have written this as a composition of operations, we could equally replace $\circ$ with $\otimes$ as the space that each $\mathcal{G}_i$ acts on is disjoint and they can therefore be implemented in parallel. This construction prevents any error of weight w from causing an error with weight $> w$ on the same or another code block.

Any gadget composed of a tensor product of Weyl operators is transversal. We show below how a transversal $\hat{X}$ operation might be implemented for the 3-qubit code.

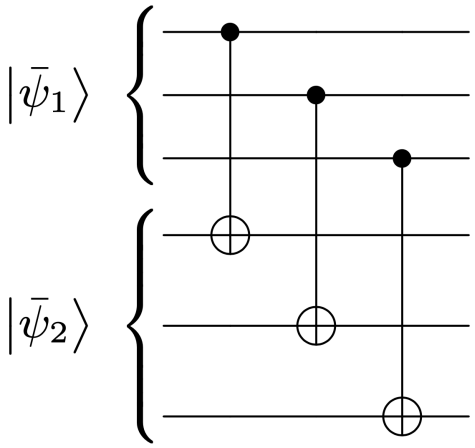


Figure 1.13: Transversal CNOT on the 3 qubit repetition code. Here, $|\bar{\psi}_1\rangle$ is the encoded input state on the control code block and

1.11 Twirling

Twirling is a process by which the average action of a map can be tailored to a desired form by applying additional operations sampled from a carefully chosen group. In this section, we show how to apply a twirl and how applying a twirl over Weyl operators to a channel results in a stochastic Weyl channel. A twirl over a group $\mathbb{G}$ is applied to a channel $\mathcal{A} : \mathbb{H} \to \mathbb{H}$ by taking the uniform average of the composition $\mathcal{G}^\dagger \mathcal{A} \mathcal{G}$ over elements G of $\mathbb{G}$. Recall that we denote the uniform average of a function f over a set $\mathbb{G}$ by

$$\mathbb{E}_{G \in \mathbb{G}}(f(G)) = \frac{1}{|\mathbb{G}|} \sum_{G \in \mathbb{G}} f(G). \tag{1.84}$$

We begin by defining a Kraus representation of $\mathcal{A}$ with Kraus operators A_k. Recall that a superoperator representation of $\mathcal{A}$ is given by

$$\mathcal{A} = \sum_{\substack{k \\ B \in \text{Basis}(\mathbb{H})}} |A_k B A_k^\dagger\rangle\!\rangle\langle\!\langle B|. \tag{1.85}$$

For convenience, we choose $\text{Basis}(\mathbb{H})$ to be the projective Weyl group with appropriate dimensions, which we will denote by simply $\mathbb{PW}$ as the dimensions are unspecifed. Applying the twirl over $\mathbb{PW}$ to $\mathcal{A}$, which we denote $\mathcal{A}^{\circlearrowleft \mathbb{PW}}$,

$$\begin{aligned}
\mathcal{A}^{\circlearrowleft \mathbb{PW}} &= \mathbb{E}_{P \in \mathbb{PW}}(\mathcal{P}^\dagger \mathcal{A} \mathcal{P}) && (1.86)\\
&= \frac{1}{|\mathbb{PW}|^2} \sum_{\substack{k \\ B,P \in \text{Basis}(\mathbb{H})}} \mathcal{P}^\dagger |A_k B A_k^\dagger\rangle\!\rangle\langle\!\langle B| \mathcal{P} && (1.87)\\
&= \frac{1}{|\mathbb{PW}|^2} \sum_{\substack{k \\ B,P \in \text{Basis}(\mathbb{H})}} |P^\dagger A_k B A_k^\dagger P\rangle\!\rangle\langle\!\langle P^\dagger B P|. && (1.88)
\end{aligned}$$

Now, because $\mathbb{PW}$ forms a basis for $\mathbb{H}$, we can decompose each A_k as a sum over elements of $\mathbb{PW}$, and we can pull out characters using eq. (1.28):

$$\begin{aligned}
\mathcal{A}^{\circlearrowleft \mathbb{PW}} &= \sum_{\substack{k \\ B,P \in \text{Basis}(\mathbb{H})}} |P^\dagger \left(\sum_{Q \in \mathbb{PW}} \alpha_{Q,k} Q \right) B \left(\sum_{W \in \mathbb{PW}} \alpha_{W,k} W \right)^\dagger P\rangle\!\rangle\langle\!\langle P^\dagger B P| && (1.89)\\
&= \frac{1}{|\mathbb{PW}|^2} \sum_{\substack{k \\ B,P,Q,W \in \text{Basis}(\mathbb{H})}} \chi_P(B) \alpha_{Q,k} \alpha_{W,k}^* |P^\dagger Q B W^\dagger P\rangle\!\rangle\langle\!\langle B| && (1.90)\\
&= \frac{1}{|\mathbb{PW}|^2} \sum_{\substack{k \\ B,P,Q,W \in \text{Basis}(\mathbb{H})}} \chi_P(B) \chi_P^*(Q) \chi_P^*(B) \chi_P^*(W^\dagger) \alpha_{Q,k} \alpha_{W,k}^* |Q B W^\dagger\rangle\!\rangle\langle\!\langle B|. && (1.91)
\end{aligned}$$

Noting that $\chi_A(B)\chi_A^*(B) = 1$, $\chi_A(B) = \chi_B^*(A)$, and $\chi_A^*(B^\dagger) = \chi_A(B)$, we can simplify further:

$$\mathcal{A}^{\circlearrowleft_{\mathbb{PW}}} = \frac{1}{|\mathbb{PW}|} \sum_{\substack{k \\ B,Q,W \in \text{Basis}(\mathbb{H})}} \chi_P^*(Q)\chi_P^*(W^\dagger)\alpha_{Q,k}\alpha_{W,k}^* |QBW^\dagger\rangle\!\rangle\langle\!\langle B| \tag{1.92}$$

$$= \frac{1}{|\mathbb{PW}|} \sum_{\substack{k \\ B,Q,W \in \text{Basis}(\mathbb{H})}} \chi_Q(P)\chi_W^*(P)\alpha_{Q,k}\alpha_{W,k}^* |QBW^\dagger\rangle\!\rangle\langle\!\langle B| \tag{1.93}$$

Using Schur's orthogonality relations, eq. (1.32),

$$\begin{aligned}
\mathcal{A}^{\circlearrowleft_{\mathbb{PW}}} &= \sum_{\substack{k \\ B,Q \in \text{Basis}(\mathbb{H})}} \alpha_{Q,k}\alpha_{Q,k}^* |QBQ^\dagger\rangle\!\rangle\langle\!\langle B| \\
&= \sum_{B \in \text{Basis}(\mathbb{H})} \left(\sum_{\substack{k \\ Q \in \text{Basis}(\mathbb{H})}} |\alpha_{Q,k}|^2 \mathcal{Q} \right) |B\rangle\!\rangle\langle\!\langle B| \\
&= \sum_{B \in \text{Basis}(\mathbb{H})} \left(\sum_{Q \in \text{Basis}(\mathbb{H})} \alpha'_Q \mathcal{Q} \right) |B\rangle\!\rangle\langle\!\langle B|
\end{aligned} \tag{1.94}$$

which is a stochastic Weyl channel for $\alpha'_Q = \sum_k |\alpha_{Q,k}|^2$. In the following section, we'll see an example of how a twirl can be leveraged to advantage in a quantum computation.

1.12 Randomized Compiling

Randomized compiling (RC) is a technique proposed in ref. [18] to tailor noise in quantum computations. The technique works by inserting random operations from the Weyl group at certain time steps in a circuit, and additional gates to undo the action of those operations, so that the full compiled circuit implements the same effective map as the bare circuit. The random gates introduced by RC can often be compiled into neighbouring gates (as in fig. 1.14) so that RC introduces little to no overhead in terms of number of gates. Averaging over these randomizations applies a twirl to the noise introduced by gates between the added random operations and corrections, thereby enforcing a stochastic Weyl noise model on those gates. In quantum computing, the results of an experiment are sampled from a distribution. Typically, we sample from the outcome distribution multiple times by implementing a computation multiple times, and each implementation is referred to as a *shot*. Because implementations of quantum computations require multiple shots, each randomization can

be added to a different shot, thereby allowing RC to be implemented without increasing the number of shots[9].

More concretely, before applying RC, circuits are expressed as alternating rounds of "easy" and "hard" gates, where we expect the easy gates to be significantly less noisy than hard gates. In most cases, the easy gates are taken to be single-qudit operations, and hard gates will include multi-qudit operations, as multi-qudit operations are often harder to implement with high precision. Correction gates are constructed by conjugating the random gates inserted before a round of hard gates by the operation implemented in the hard round. In chapter 4, we will extend this protocol to logical operations, and theorem 14 gives a concrete mathematical description of how such a randomization tailors noise in the general case. We therefore defer a mathematical treatment to that section.

[9] In chapter 4, we will discuss the case where only a single shot is taken to sample from a distribution.

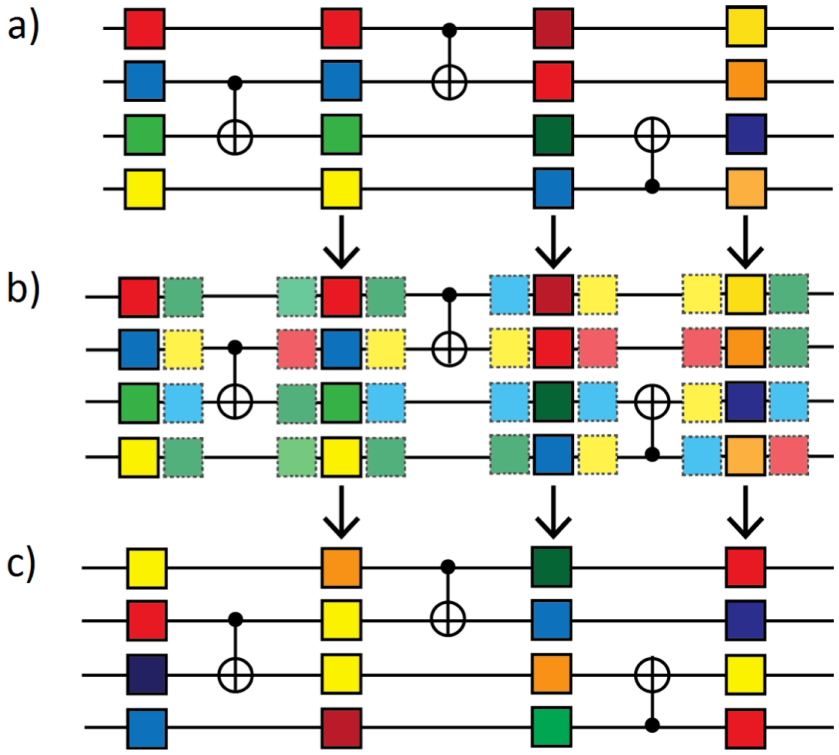


Figure 1.14: Reprinted figure with permission from [18, Figure 1] Copyright 2016 by the American Physical Society. Figure shows how to randomly compile a circuit. a) Shows a bare circuit with rounds of easy and hard gates (in this example, the easy gates are single-qudit gates and the hard gates are $\hat{X}$ gates). b) Shows the insertion of random operations before each hard gate, with corresponding corrections inserted after. c) shows the additional operations introduced in b) being compiled into the neighbouring easy gates so that the final circuit has the same depth as the bare circuit in a).

Chapter 2

Efficiently computing logical noise in quantum error-correcting codes

The results presented in this chapter were first presented in [2], with the exception of section 2.5.3, which is as-yet unpublished. This chapter presents work which was begun in my Masters thesis on using symmetries in QECs to calculate the effective logical noise more efficiently by leveraging degeneracies in the logical noise that arise as a result of these symmetries [1]. The results presented in my Masters thesis restricted attention to perfect syndrome extraction, dealt only with stabilizer codes, and considered only degeneracies, with no consideration of logical degeneracies, that is, attention was restricted to maps that are exactly identical rather than identical up to logical operations. The symmetries presented for the 3-qubit, 5-qubit, and Steane codes were presented in [21] and/or [1]. My Masters thesis utilized symmetries in treatment of a hybrid soft/hard decoder for a concatenated 5-qubit code, but did not consider general concatenated codes. This chapter generalizes these results to arbitrary quantum error correcting codes, including concatenation, considers noisy measurement, extends treatment to consider equivalence up to logical operations, and presents new results. New results include a no-go theorem that states that there is no global symmetry group for any code which applies to all noise processes, symmetries of the toric code, symmetries for multiple rounds of error correction, and generalized symmetries for non-IID noise.

2.1 Chapter overview

Quantum error correction protocols have been developed to offset the high sensitivity to noise inherent in quantum systems. However, much is still unknown about the behaviour of a quantum error-correcting code under general noise, including noisy measurements. This lack of knowledge is largely due to the computational cost of simulating quantum systems large enough to perform nontrivial encodings. In this chapter, we develop general methods for incorporating noisy measurement operations into simulations of quantum error-correcting codes and show that measurement errors on readout qubits manifest as a renormalization on the effective logical noise. We also derive general methods for reducing the computational complexity of calculating the exact effective logical noise by many orders of magnitude. This reduction is achieved by determining when different recovery operations produce equivalent logical noise. These methods could also be used to better approximate soft decoding schemes for concatenated codes or to reduce the size of a lookup table to speed up the error correction step in implementations of quantum error-correcting codes. We give examples of such reductions for the 3-qubit, 5-qubit, Steane, concatenated, and toric codes.

2.2 Introduction

In recent years, there has been a resurgence of interest in studying the behaviour of general and specific noise in QECCs [21, 30, 42–48]. There has also been interest in studying symmetries in quantum error correcting codes for several purposes, including noise tailoring [47], code construction via classical cyclic codes [49], applying logical operations [50, 51], and reducing the complexity of simulations [21]. In this chapter, we extend the previous formalism used to study quantum memories [52] to include noisy measurements. We then derive general conditions under which two syndromes will result in equivalent logical noise in a QECC. These degeneracies can reduce the computational cost of studying QECCs by orders of magnitude, and once degeneracies are found for a given QECC, they can be re-used for any future simulation of that code under any applicable noise model without any additional computation. Moreover, part of the motivation for simulating QECCs is to determine good choices of recovery maps for each sydrome. By establishing general and simple conditions under which recovery maps associated with different syndromes are degenerate, we can reduce the number of syndromes for which good choices need to be cached or have complex calculations performed. We anticipate that this will enable faster implementations of decoders, especially in memory-constrained environments (such as cryogenic control computers or decoders built

into FPGAs), so that errors can be corrected before they cascade.

In section 2.2.1 of this chapter, we show how measurements can be treated like a channel and, conversely, how noise can be treated like a measurement channel. In section 2.4, we derive general results showing when the effective noise conditioned upon two different measurement outcomes is equivalent. In section 2.5, we demonstrate that noisy readout measurements in stabilizer codes leave the effective logical noise invariant up to a renormalization and show how our results can be applied to reduce the simulation cost by applying our results to several small stabilizer codes, the general toric code, and concatenated codes. Even at the first level of concatenation, our results can reduce the exact simulation cost of a soft decoder (see [32] for a description of soft decoding) by a factor of $64^7/34,992 \approx 10^8$ for depolarizing noise in the Steane code. Similar results were obtained for some of these codes under a more restricted noise model (namely, noise models with a single Kraus operator) in ref. [21], however, the results here are broader (they identify more degenerate syndrome maps), more general (they apply to general codes and noise maps, including noisy readout measurements), and allow new symmetry operations to be verified with ease.

2.2.1 Measurements as channels and noise channels as measurements

Experimental imperfections cause realistic measurements to deviate from an ideal projective measurement in a number of ways. First, noise may result in superoperators being applied to the system before and/or after the ideal measurement. Second, the measurement procedure may not be describable by a PVM, but rather by a Positive-Operator-Valued Measure (POVM), which is a subset $\mathbb{M}' \subset \mathrm{Pos}(\mathbb{H})$ such that $\sum_{\pi' \in \mathbb{B}'} \pi' = I_{\mathbb{H}}$ (that is, we relax the assumption that the elements of a PVM are projectors). The state update rule for PVMs introduced in eq. (1.58) (and included in eq. (2.1) for convenience) can be directly generalized to POVMs using the decomposition $\pi' = \pi^\dagger \pi \; \forall \pi' \in \mathbb{M}' \subset \mathrm{Pos}(\mathbb{H})$ to define an operator π (which need not be positive and is not unique).

So for a PVM $\{\pi\}$ or POVM $\{\pi'\}$ with decompositions $\pi' = \pi^\dagger \pi$ (where the π used to decompose each π' are not the elements of the PVM but use the same symbol for the purpose of writing eq. (2.1) in a way that applies to both types of measurement), the post-measurement state when outcome π is observed is

$$\rho'_\pi = \frac{\Pi[\rho]}{\mathrm{Tr}[\Pi[\rho]]}. \tag{2.1}$$

Crucially, for both PVMs and POVMs the post-measurement state is not a linear function of the input state. However, the non-linearity is of a benign form, namely, the non-linearity

from conditioning upon the output state that arises from Bayes' rule. With some abuse of notation (namely, interpreting 0/0 as 1), the expected post-measurement state is

$$\sum_{\pi \in \mathbb{M}} \rho'_\pi \mathrm{Tr}\Pi[\rho] = \sum_{\pi \in \mathbb{M}} \Pi[\rho]. \tag{2.2}$$

To accomodate additional noise processes, we can simply replace the implicit linear maps $\Pi[\rho] = \pi\rho\pi^\dagger$ by general linear maps that can include pre- and post-measurement control operations and noise maps. That is, measurements can be defined as either a set of operators (as introduced in section 1.7) or as a set of superoperators. For this chapter, we define a general measurement $\mathbb{M}$ to be a set of superoperators, that is, $\mathbb{M} \subset \mathbb{B}(\mathbb{B}(\mathbb{H}), \mathbb{B}(\mathbb{K}))$, although we will only consider the case $\mathbb{K} = \mathbb{H}$. To be physical, each $\mathcal{M} \in \mathbb{M}$ must map valid quantum states to valid quantum states (that is, be completely positive but generally not trace-preserving). To be a complete measurement, some outcome must always occur if the input state is a density matrix and so

$$\sum_{\mathcal{M} \in \mathbb{M}} \mathcal{M} \tag{2.3}$$

must be a trace-preserving map. Note that any completely positive map can be represented in Kraus operator form as a channel in the form of eq. (2.3), and so noise processes can formally be regarded as a measurement where the outcome is not recorded.

2.3 Encoded memory channels

In this chapter, we focus on what is perhaps the simplest example of an implementation of a quantum error corrected channel. Specifically, we deal with encoded memory channels, comprised of the following steps:

1. Apply an encoding map $\mathcal{U}$.
2. Apply a noise map $\mathcal{N}$.
3. Perform a syndrome measurement $\mathbb{M}$.
4. Perform a recovery operation $\mathcal{R}$ conditioned on the outcome of the measurement.
5. Apply a decoding map $\mathcal{U}^\dagger$.

Recall that an encoding of $\mathbb{H}$ into $\mathbb{K}$ is an isometry $U \in \mathbb{U}(\mathbb{H}, \mathbb{K})$, which can be expressed as $\sum_{j \in \mathbb{Z}_d} |\bar{j}\rangle\langle j|$. For any encoding, we can choose a set of recovery maps $\mathbb{V} \subset \mathbb{U}(\mathbb{K})$ such that the set of projectors $\{\Pi_R = \mathcal{R}\mathcal{U}\mathcal{U}^\dagger\mathcal{R}^\dagger : R \in \mathbb{V}\}$ associated with the cospaces of the code, form a projective measurement. That is, the recovery maps satisfy $U^\dagger Q^\dagger R U = \delta_{Q,R} 1_{\mathbb{H}}$ for all $Q, R \in \mathbb{V}$ and $\sum_{R \in \mathbb{V}} RUU^\dagger R^\dagger = 1_{\mathbb{K}}$.

For example, the 3-qubit repetition code encodes 1 logical qubit into 3 physical qubits via the isometry $U = |000\rangle\langle 0| + |111\rangle\langle 1|$. We can extend this isometry into a projective measurement on the encoded space by choosing, for example, $\mathbb{V} = \{III, XII, IXI, IIX\}$. The elements of $\mathbb{V}$ can be regarded as the most likely errors to occur. Thus, if a system is prepared in the encoded state $U|\psi\rangle$ and a specific error $E \in \mathbb{V}$ occurs, the ideal PVM $\{\Pi_R = \mathcal{R}\mathcal{U}\mathcal{U}^\dagger\mathcal{R}^\dagger : R \in \mathbb{V}\}$ will result in the outcome E, which can then be corrected by applying $\mathcal{E}^\dagger$.

In this chapter, because we are dealing exclusively with channels of the form described above, we fold the recovery maps $\mathcal{R}^\dagger$ for each cospace into the measurement for convenience and define the ideal measurement to be $\mathbb{M} = \{\mathcal{M}\}$, where $\mathcal{M} = \mathcal{R}^\dagger \Pi_R$ so that $\mathbb{M} = \{\mathcal{U}\mathcal{U}^\dagger\mathcal{R}^\dagger : R \in \mathbb{V}\}$. A QECC is then a pair $(U, \mathbb{M})$[1], and can be used to protect a logical qubit against a noisy physical process $\mathcal{N}$ (typically a completely positive, trace-preserving map), thereby reducing the above steps to

1. Apply an encoding map $\mathcal{U}$.
2. Apply a noise map $\mathcal{N}$.
3. Perform a syndrome measurement $\mathbb{M}$ (including recovery).
4. Apply a decoding map $\mathcal{U}^\dagger$.

While this procedure includes applying the encoding and decoding maps, in practice one would typically prepare a logical state directly and measure the expectation values of encoded operators or treat the output as an encoded input into a subsequent round of error correction, where subsequent operations can be conditioned upon the observed outcome of $\mathbb{M}$. Often, a gate will be applied before the correction step (step 3); in this case, $\mathcal{N}$ is replaced with a possibly noisy implementation of a gate or a fault-tolerant gadget.

[1] Note that because we have folded the recovery operations into $\mathbb{M}$, this pair implicitly defines the set of corrected errors as those that are equivalent to a recovery operation up to trivial operations on the logical space.

When the outcome $\mathcal{M} \in \mathbb{M}$ is observed, the above process results in the conditional map

$$\bar{\mathcal{N}}_U(\mathcal{M}) = \mathcal{U}^\dagger \mathcal{M} \mathcal{N} \mathcal{U}, \tag{2.4}$$

where we do not divide by the probability with which the outcome occurs. We define the average logical channel as the average over the conditional maps,

$$\bar{\mathcal{N}}_U = \frac{1}{|\mathbb{M}|} \sum_{\mathcal{M} \in \mathbb{M}} \bar{\mathcal{N}}_U(\mathcal{M}). \tag{2.5}$$

The average logical channel is often used to benchmark the performance of a given QECC $(U, \mathbb{M})$ against a given noise model $\mathcal{N}$.

For ideal measurements, the ideal recovery operator uniquely specifies the measurement outcome. Therefore we define the effective map conditioned on the recovery map R to be the effective map conditioned on $\mathcal{M}_R = \mathcal{U}\mathcal{U}^\dagger \mathcal{R}^\dagger$, following our convention of folding the recovery map into the measurement. That is, we define

$$\dot{\mathcal{N}}_U(R) = \bar{\mathcal{N}}_U(\mathcal{M}_R) = \mathcal{U}^\dagger \mathcal{R}^\dagger \mathcal{N} \mathcal{U}, \tag{2.6}$$

as in ref. [52], where we have used the fact that $U^\dagger U U^\dagger = U^\dagger$ for the isometry U and we use $\dot{\mathcal{N}}$ and Roman font rather than $\bar{\mathcal{N}}$ and caligraphic font to differentiate between conditioning on a measurement or a recovery operation.

As with the usual model for noisy measurements, to model noisy readout measurements in quantum error correcting codes, we assume that the effective measurement on the encoded space can be represented as a probabilistic sum over ideal projectors onto the cospaces of the code. That is, for a measurement outcome associated with a recovery operation R that is applied based on a reported outcome, the noisy measurement channel $\Theta(\Pi_R)$ is given by

$$\Theta(\Pi_R)[\rho] = \sum_{Q \in \mathbb{R}} \bar{C}_{Q,R} \Pi_Q[\rho], \tag{2.7}$$

where $\bar{C}_{Q,R}$ is an element of the logical $2^{n-k} \times 2^{n-k}$ confusion matrix, $\bar{C}$. One method used to protect against measurement errors of this form is to measure each stabilizer generator multiple times and correct according to the outcome that occurs most frequently. We prove that this assumption is valid for stabilizer codes with ideal syndrome extraction circuits and noisy measurement of the readout qubits in section 2.5. To be consistent with the effective logical map defined for the case where measurement is ideal, we fold the recovery operation into the noisy measurement map, to define

$$\tilde{\mathcal{M}}_R = \mathcal{R}^\dagger \Theta(\Pi_R). \tag{2.8}$$

Note that because ideal measurements are a special case of noisy measurements where $\bar{C}_{j,m} = \delta_{j,m}$, any results derived using noisy measurements of the form $\tilde{\mathcal{M}}_R$ throughout this chapter hold equally for ideal measurements. We denote the effective logical channel with noisy measurements the same as that for ideal measurements but with the addition of a $\tilde{}$ over $\dot{\mathcal{N}}$ or $\bar{\mathcal{N}}$. We can then write

$$\begin{aligned}
\tilde{\dot{\mathcal{N}}}_U(R) &= \tilde{\bar{\mathcal{N}}}_U(\tilde{\mathcal{M}}_R) && (2.9)\\
&= \mathcal{U}^\dagger \tilde{\mathcal{M}}_R \mathcal{N} \mathcal{U} && (2.10)\\
&= \mathcal{U}^\dagger \mathcal{R}^\dagger \Theta(\Pi_R) \mathcal{N} \mathcal{U} && (2.11)\\
&= \sum_{Q \in \mathbb{R}} \bar{C}_{Q,R} \mathcal{U}^\dagger \mathcal{R}^\dagger \Pi_Q \mathcal{N} \mathcal{U} && (2.12)\\
&= \bar{C}_{R,R} \dot{\mathcal{N}}_U(R), && (2.13)
\end{aligned}$$

The simplification from eq. (2.12) to eq. (2.13) is possible because we have the decoding map, $\mathcal{U}^\dagger$ after the measurement so that any terms not in the codespace after the measurement are removed by $\mathcal{U}^\dagger$. In the case where the state is not decoded after the error correction step, the cospaces will remain populated from cases where $\delta_{Q,R} = 0$. Examining eq. (2.13), we see that the effective logical channel with noisy measurements is identical to the effective logical channel with ideal measurement, up to a renormalization. For the remainder of this chapter, when we refer to noisy measurements we mean measurements with noise of the form described in section 1.7.2, that is, ideal projectors with probabilistically misreported outcomes.

2.4 Symmetries in encoded memory channels

The above error correction procedure is defined to be a pair $(U, \mathbb{M})$, where $\mathbb{M}$ is the measurement including recovery operators and we often label measurement outcomes by the associated recovery map. To simulate the error correction procedure, we need to compute the conditional maps $\bar{\mathcal{N}}_U(\mathcal{M})$. There are many maps and each conditional map is typically expensive to compute. However, as observed in [21, 30], many of the conditional maps are

related, which can be exploited to reduce the computation time. We define two superoperators $\mathcal{N}, \mathcal{N}' \in \mathbb{B}(\mathbb{B}(\mathbb{H}))$ to be degenerate if $\mathcal{N}' = c\mathcal{N}$ for some $c \in \mathbb{R}$ and to be logically degenerate if there exist invertible superoperators $\bar{\mathcal{A}}, \bar{\mathcal{B}} \in \mathbb{B}(\mathbb{B}(\mathbb{H}))$ such that

$$\mathcal{N}' = c\bar{\mathcal{A}}\mathcal{N}\bar{\mathcal{B}} \tag{2.14}$$

for some $c \in \mathbb{R}$. We are primarily interested in the superoperators associated to measurement outcomes, and so we say that two outcomes $\mathcal{Q}, \mathcal{R} \in \mathbb{M}$ are (logically) degenerate for a fixed noise process $\mathcal{N}$ if the corresponding superoperators $\bar{\mathcal{N}}_U(\mathcal{Q})$ and $\bar{\mathcal{N}}_U(\mathcal{R})$ are (logically) degenerate. We say that two measurement outcomes are nondegenerate for a fixed noise process $\mathcal{N}$ if they are not known to be degenerate under $\mathcal{N}$. We make this particular distinction because we may not know all degeneracies for a class of noise processes. We call a set of degenerate measurement outcomes a degeneracy class and a set of logically degenerate measurement outcomes a logical degeneracy class.

Some degeneracy relations are easily established for a noise process using stabilizers and logical operators. Stabilizers and logical operators of an encoding U can be expressed as invertible superoperators $\mathcal{S}, \mathcal{L} \in \mathbb{B}(\mathbb{B}(\mathbb{K}))$ such that $\mathcal{SU} = \mathcal{U}$ and $\mathcal{LU} = \mathcal{U}\bar{\mathcal{L}}$ for some $\bar{\mathcal{L}} \in \mathbb{B}(\mathbb{B}(\mathbb{H}))$ respectively. Note that the set of logical operators refers to operations on the logical space, rather than restricting to Pauli or Weyl logical operations. A stabilizer is a special case of a logical operator where $\bar{\mathcal{L}} = 1_{\mathbb{B}(\mathbb{H})}$. For example, for the 3-qubit code, the superoperators corresponding to ZZI and ZZZ are a stabilizer and a nontrivial logical operator of the encoding $U = |000\rangle\langle 0| + |111\rangle\langle 1|$ respectively. The (general) stabilizer and logical groups are the groups $\mathbb{GS}(U)$ of stabilizer and $\mathbb{GL}(U)$ of logical operations respectively. Note that the "general" stabilizer group is distinct from the stabilizer group of a stabilizer code as it includes any operation for which the above conditions hold. The general stabilizer group can have e.g. permutation operations which relabel the physical qubits. The stabilizer group, $\mathbb{S}$, of a stabilizer code is a subset of the general stabilizer group, that is $\mathbb{S} \subset \mathbb{GS}(U)$.

For any stabilizer $\mathcal{S} \in \mathbb{GS}(U)$ and logical operator $\mathcal{L} \in \mathbb{GL}(U)$, we have $\bar{\mathcal{N}}_U(\mathcal{RS}) = \bar{\mathcal{N}}_U(\mathcal{R})$ and $\bar{\mathcal{N}}_U(\mathcal{LR}) = \bar{\mathcal{L}}\bar{\mathcal{N}}_U(\mathcal{R})$. Therefore recovery maps in the same left coset of $\mathbb{GS}(U)$ will be degenerate and recovery maps in the same left coset of $\mathbb{GL}(U)$ will be logically degenerate. The above observation can be used to change a single measurement outcome $\mathcal{M}$ to make $\bar{\mathcal{N}}_U(\mathcal{M})$ closer to a given logical operation (in particular, the identity operation). However, two distinct elements $Q, R \in \mathbb{V}$ cannot be related by a logical operation, as otherwise they would violate the assumption that $U^\dagger Q^\dagger R U = 0$, that is, that the associated measurement operators are orthogonal. Therefore the relationship $\bar{\mathcal{N}}_U(\mathcal{LR}) = \bar{\mathcal{L}}\bar{\mathcal{N}}_U(\mathcal{R})$ can-

not speed up the computation of the full set of conditional maps for a fixed measurement. The results presented in the remainder of this section, however, do enable a reduction in computational complexity that allows a speed up. Before proceeding any further, we prove that no distinct outcomes are logically degenerate for all noise processes under ideal or noisy measurements, and hence any degeneracies can only hold for restricted noise models.

Theorem 1. *For any encoding $U \in \mathbb{U}(\mathbb{H})$ and any two recovery maps $Q, R \in \mathbb{V}$ associated to different (possibly noisy) measurement outcomes, there exist noise models $\mathcal{N}$ such that $\mathcal{Q}$ and $\mathcal{R}$ are not degenerate under $\mathcal{N}$.*

Proof. Let $\mathcal{N} = \mathcal{Q}$, so

$$\tilde{\dot{\mathcal{Q}}}_U(Q) = \bar{C}_{Q,Q}\dot{\mathcal{Q}}_U(Q) = \bar{C}_{Q,Q}\mathcal{U}^\dagger\mathcal{Q}^\dagger\mathcal{Q}\mathcal{U} = \bar{C}_{Q,Q}\bar{\mathcal{I}} \tag{2.15}$$

Recalling that $U^\dagger R^\dagger Q U = \delta_{Q,R} I_{\mathbb{H}}$, we have

$$\tilde{\dot{\mathcal{Q}}}_U(R) = \bar{C}_{R,R}\dot{\mathcal{Q}}_U(R) = \bar{C}_{R,R}\mathcal{U}^\dagger\mathcal{R}^\dagger\mathcal{Q}\mathcal{U} = \bar{\mathbf{0}}. \tag{2.16}$$

Therefore for any invertible superoperators $\bar{\mathcal{A}}, \bar{\mathcal{B}} \in \mathbb{B}(\mathbb{B}(\mathbb{H}))$,

$$\bar{\mathcal{A}}\tilde{\dot{\mathcal{Q}}}_U(R)\bar{\mathcal{B}} = \bar{\mathbf{0}} \neq \tilde{\dot{\mathcal{Q}}}_U(Q) \tag{2.17}$$

as required. □

Despite the apparently strong statement of theorem 1, it has been observed that different measurement outcomes can be degenerate for broad families of noise processes [21, 30]. ref. [21] gave conditions based on code symmetries to identify degenerate syndrome maps for independent and identically distributed (IID) unitary noise in stabilizer codes, where a noise channel $\mathcal{N}$ is IID if it can be written as $\mathcal{N} = \mathcal{N}_1^{\otimes n}$ for some noise process, $\mathcal{N}_1$, acting on a single system. We present a more general result with a trivial and constructive proof, giving conditions under which different recovery operations or measurements are *logically* degenerate for general QECCs. These conditions hold for more general (including non-IID) noise. Our results also show how logical operations can be factored into the error correction step by updating ideal recovery operations to other operations which are logically degenerate to their ideal counterparts. The following proposition follows directly from the definitions laid out above. This result seems trivial in light of our notation, however, arriving at such conclusions in less abstract settings (e.g., for stabilizer codes) is quite challenging.

Proposition 2. *Let $U \in \mathbb{U}(\mathbb{H}, \mathbb{K})$, $\mathcal{M} \in \mathbb{M}$ be a measurement outcome, and $\mathcal{N} \in \mathbb{B}(\mathbb{B}(\mathbb{K}))$. For any $\mathcal{A}, \mathcal{B} \in \mathbb{GL}(U)$, the maps $\mathcal{U}^\dagger\mathcal{B}^\dagger\tilde{\mathcal{M}}\mathcal{N}\mathcal{A}\mathcal{U}$ and $\mathcal{U}^\dagger\tilde{\mathcal{M}}\mathcal{N}\mathcal{U}$ are logically degenerate.*

Note that the the symmetry operators $\mathcal{A}, \mathcal{B} \in \mathbb{GL}(U)$ are not required to be unitary. From proposition 2, we can immediately identify some degenerate measurement outcomes when there are logical operators that commute with the noise. Specifically, we define the logical symmetry group of an encoding U under a noise process $\mathcal{N}$ to be the group $\mathbb{GL}(U, \mathcal{N}) = \{\mathcal{L} \in \mathbb{GL}(U) : [\mathcal{N}, \mathcal{L}] = 0\}$, and the stabilizer symmetry group to be $\mathbb{GS}(U, \mathcal{N}) = \mathbb{GL}(U, \mathcal{N}) \cap \mathbb{GS}(U)$.

Corollary 3. *Let $U \in \mathbb{U}(\mathbb{H}, \mathbb{K})$, $\tilde{\mathcal{M}} \in \tilde{\mathbb{M}}$, and $\mathcal{N} \in \mathbb{B}(\mathbb{B}(\mathbb{K}))$. The maps $\{\tilde{\tilde{\mathcal{N}}}_U(\mathcal{B}^\dagger \tilde{\mathcal{M}} \mathcal{A}) : \mathcal{A} \in \mathbb{GL}(U, \mathcal{N}), \mathcal{B} \in \mathbb{GL}(U)\}$ are logically degenerate. Furthermore, the maps $\{\tilde{\tilde{\mathcal{N}}}_U(\mathcal{B}^\dagger \tilde{\mathcal{M}} \mathcal{A}) : \mathcal{A} \in \mathbb{GS}(U, \mathcal{N}), \mathcal{B} \in \mathbb{GS}(U)\}$ are degenerate.*

Proof. Let $\mathcal{A} \in \mathbb{GL}(U, \mathcal{N})$ and $\mathcal{B} \in \mathbb{GL}(U)$. By assumption, there exist invertible $\bar{\mathcal{A}}, \bar{\mathcal{B}} \in \mathbb{B}(\mathbb{B}(\mathbb{H}))$ such that $\mathcal{A}\mathcal{U} = \mathcal{U}\bar{\mathcal{A}}$ and $\mathcal{B}\mathcal{U} = \mathcal{U}\bar{\mathcal{B}}$. Therefore by eqs. (2.4) and (2.13) we have

$$\begin{aligned}
\tilde{\tilde{\mathcal{N}}}(\mathcal{B}^\dagger \tilde{\mathcal{M}} \mathcal{A}) &= \bar{C}_m \bar{\mathcal{N}}_U(\mathcal{B}^\dagger \mathcal{M} \mathcal{A}) = \bar{C}_m \mathcal{U}^\dagger \mathcal{B}^\dagger \mathcal{M} \mathcal{A} \mathcal{N} \mathcal{U} \\
&= \bar{C}_m \mathcal{U}^\dagger \mathcal{B}^\dagger \mathcal{M} \mathcal{N} \mathcal{A} \mathcal{U} = \bar{C}_m \bar{\mathcal{B}}^\dagger \mathcal{U}^\dagger \mathcal{M} \mathcal{N} \mathcal{U} \bar{\mathcal{A}} \\
&= \bar{C}_m \bar{\mathcal{B}}^\dagger \bar{\mathcal{N}}_U(\mathcal{M}) \bar{\mathcal{A}},
\end{aligned}$$

where $\bar{C}_m$ is the diagonal element of the logical confusion matrix associated with measurement outcome $\mathcal{M}$, as required. Note that the logical group and the logical symmetry group are groups, and so all pairs in the first set are logically degenerate. The final statement holds because $\bar{\mathcal{A}} = \bar{\mathcal{I}} = \bar{\mathcal{B}}$ if $\mathcal{A}, \mathcal{B} \in \mathbb{GS}(U)$ by definition. □

Measurement is often assumed to be noiseless in studies of quantum computing, and, in fact, this assumption was made in each of the papers which observed symmetries in QECCs before the work presented in this chapter, namely refs [1, 21, 30]. The symmetry conditions for QEC are given by corollary 4, where the order and daggers differ because of the $\mathcal{R}^\dagger$ in eq. (2.6). We also denote the subset of a set of superoperators $\mathbb{A} \subset \mathbb{B}(\mathbb{B}(\mathbb{H}))$ that have a single Kraus operator by $\mathbb{A}_K$ so that we can use the bijection $A \leftrightarrow \mathcal{A}$ in the following statement.

Corollary 4. *Let $U \in \mathbb{U}(\mathbb{H}, \mathbb{K})$, $R \in \mathbb{V}$, and $\mathcal{N} \in \mathbb{B}(\mathbb{B}(\mathbb{K}))$. The maps $\{\tilde{\tilde{\mathcal{N}}}_U(A^\dagger R B) : \mathcal{A} \in \mathbb{GL}_K(U, \mathcal{N}), \mathcal{B} \in \mathbb{GL}_K(U)\}$ are logically degenerate. Furthermore, the maps $\{\tilde{\tilde{\mathcal{N}}}_U(A^\dagger R B) : \mathcal{A} \in \mathbb{S}_K(U, \mathcal{N}, \mathcal{B} \in \mathbb{S}_K(U)\}$ are degenerate.*

A simple application of corollary 4 is to independent and identically distributed (IID) noise, that is, to superoperators $\mathcal{N}$ that can be written as $\mathcal{N} = \mathcal{N}_1^{\otimes n}$ for some noise process $\mathcal{N}_1 \in \mathbb{B}(\mathbb{B}(\mathbb{J}))$ acting on a single system with Hilbert space $\mathbb{J}$. For such noise, $\mathbb{GL}(U, \mathcal{N}_1^{\otimes n})$

will typically contain permutation operators. Moreover, the recovery maps R are typically chosen to be tensor products of elements of some "nice error basis"[53] $\mathbb{E} \subset \mathbb{U}(\mathbb{J})$ that contains the identity and can be used to construct the ideal PVM, which, for qubits, is taken to be the set of single-qubit Pauli matrices. We define the weight of some recovery map $R \in \mathbb{E}^{\otimes n}$ to be the number of subsystems on which it acts nontrivially. Recall that a group of permutations of n objects is k-transitive if every ordered subset of k objects can be mapped to every other ordered subset of k objects. When a group is 1-transitive, we say that it is transitive. We then have the following.

Corollary 5. *Let n be a positive integer, $U \in \mathbb{U}(\mathbb{H}, \mathbb{K}^{\otimes n})$ be an encoding, $\mathbb{E} \subset \mathbb{U}(\mathbb{K})$ be a nice error basis, and $\mathcal{N} \in \mathbb{B}(\mathbb{B}(\mathbb{K}^{\otimes n}))$. Then if $\mathbb{GS}(U, \mathcal{N})$ ($\mathbb{GL}(U, \mathcal{N})$) contains a k-transitive group, the set of weight k errors under ideal or noisy measurements will be partitioned into at most $\binom{|\mathbb{E}|+k-2}{k}$ (logical) degeneracy classes.*

Proof. From corollary 4, for any permutation $\mathcal{P} \in (\mathbb{GL}(U, \mathcal{N}))\ \mathbb{GS}(U, \mathcal{N})$ of the tensor factors of $\mathbb{K}^{\otimes n}$ and any $R \in \mathbb{E}^{\otimes n}$, the conditional maps $\dot{\mathcal{N}}_U(P^\dagger R P)$ and $\tilde{\mathcal{N}}_U(R)$ are (logically) degenerate. The number of distinct unordered combinations of length k from s items is given by $\binom{s+k-1}{k}$. Then corollary 5 follows directly from corollary 4 and the fact that there are $\binom{|\mathbb{E}|+k-2}{k}$ distinct unordered combinations of k of the $|\mathbb{E}| - 1$ nontrivial errors. □

For example, let $\mathbb{E} = \mathbb{P} = \{I, X, Y, Z\}$ be the set of single-qubit Pauli operators. Then if $\mathbb{GL}(U, \mathcal{N})$ contains a 1-transitive group, the weight-1 Pauli operators for IID noise are partitioned into at most 3 degeneracy classes. If $\mathbb{GL}(U, \mathcal{N})$ contains a 2-transitive group, then there will be at most $\binom{4}{2} = 6$ degeneracy classes of weight 2 Pauli errors.

2.5 Symmetries of stabilizer codes

As an application of section 2.4, we now consider how symmetries can be used to accelerate simulations of Pauli stabilizer codes under IID noise. We focus on symmetries $\mathcal{A}$ and $\mathcal{B}$ that correspond to conjugation by unitary operators $A, B \in \mathbb{U}(\mathbb{H})$ with $A = B$ and typically set $\bar{A} = \bar{I}$. As we are considering IID noise, any permutation of the qubits in $\mathbb{GL}(U)$ is an element of the symmetry group of U under $\mathcal{N}$. We can then use logical operations and elements of the symmetry group to find sets of degenerate recovery maps.

For a stabilizer code with generators $\{G_i\}$ and logical Pauli operations $\mathbb{L}$, any state in

	3-Qubit Code	5-Qubit Code	Steane Code
Permutations	(0 1 2)	(0 1 2 3 4) (0 4)(1 3)	(3 4)(5 6) (0 3 1)(2 4 5)
Transitivity	1	1	2
$\lvert\mathbb{V}\rvert$	4	16	64
$\lvert\mathbb{V}_{\text{IID}}\rvert$	2	4	5

Table 2.1: Permutations that leave the code space invariant for common stabilizer codes [1, 21]. We also list the transitivity of the symmetry groups generated by these permutations, the number $\lvert\mathbb{V}\rvert$ of recovery maps, the number $\lvert\mathbb{V}_{\text{IID}}\rvert$ of logically nondegenerate Pauli recovery maps for generic IID noise under the permutation group formed by the listed permutations.

the code space can be written as

$$\rho = \left(\sum_{L\in\mathbb{L}} \mu_L L\right) \prod_{i=0}^{n-k-1} \tfrac{1}{2}(I + G_i), \tag{2.18}$$

where $\mu_L \in [-1, 1]$. Stabilizer generators and $\bar{X}$ and $\bar{Z}$ operators for common $(n, 1)$-stabilizer codes and the toric code are listed in section 1.8.4.

From eq. (2.18), we see that $\mathbb{L}(U) = \langle \mathbb{S}, \mathbb{L} \rangle$ forms a basis for the codespace of a stabilizer code with logical Pauli operations $\mathbb{L}$ and stabilizer group $\mathbb{S}$. Then any permutation operator that permutes the elements of the stabilizer group, $\mathbb{S}$, and leaves the elements of $\mathbb{L}$ invariant will be an element of the (general) stabilizer group $\mathbb{GS}(U)$. Similarly, any permutation operator that permutes the elements of the stabilizer group, $\mathbb{S}$, and permutes the elements of the logical Pauli group $\mathbb{L}(U)$ will be an element of the (general) logical group $\mathbb{GL}(U)$.

Therefore we can find permutation operators in the symmetry group of the corresponding code for IID noise and so partition the recovery operators into degeneracy classes using corollary 4 by considering only the action of permutations on the stabilizer $\mathbb{S}$ and logical $\mathbb{L}$ groups. In table 2.1 we list permutation operators that generate transitive groups for each code, and a 2-transitive group for the Steane code. As $\bar{X}$ and $\bar{Z}$ are permutationally invariant for these codes, the permutation operators are in the general stabilizer group $\mathbb{GS}(U)$.

For (n, k)-stabilizer codes, it is common to consider only Pauli recovery maps. Pauli recovery maps for an (n, k)-stabilizer code with encoding U can be written as $\{TL_T : T \in \mathbb{P}^{\otimes n}/\mathbb{L}(U)\}$, where any choice of Pauli operator $L_T \in \mathbb{L}(U)$ for each T will define a valid set of recovery maps. The set $\mathbb{P}^{\otimes n}/\mathbb{L}(U)$ is sometimes referred to as the set of pure errors [32]. For an (n, k)-stabilizer code, there are 2^{n-k} recovery maps, where typically $k \ll n$. Using

Code	Symmetry Operation	Logical Operation	$\lvert\mathbb{V}_{\text{dep}}\rvert$
5-Qubit	$\mathcal{Q}^{\otimes 5}$	$\overline{Q}$	2
Steane	$\mathcal{Q}^{\otimes 7}$	$\overline{Q}$	3
	$\mathcal{H}^{\otimes 7}$	$\overline{\mathcal{H}}$	

Table 2.2: An incomplete list of non-trivial operations which induce symmetries in some of the more popular quantum error correcting codes and the number $\lvert\mathbb{V}_{\text{dep}}\rvert$ of logically nondegenerate Pauli recovery maps for local depolarizing noise under the listed symmetry operations and the permutation symmetries listed in table 2.1. Note that these symmetries are valid under the conditions of corollary 3, and therefore require that the physical noise acting on the system commute with the symmetry operator. For example, we can have physical noise $\mathcal{N} = \mathcal{D}_p^{\otimes n}$, where $\mathcal{D}_p$ is a single qubit depolarizing channel with parameter p because $[\mathcal{Q}, \mathcal{D}_p] = [\mathcal{H}, \mathcal{D}_p] = 0$.

the permutation operators, we can reduce the number of distinct conditional maps that need to be computed for IID noise using corollary 4. Note that the choice of L_T will not affect the number of logically degenerate recovery maps, however, it will change the number of degenerate recovery maps. This is because applying L_T to a recovery map only alters the corresponding conditional map by a logical operation and keeps the same syndrome, so toggling L_T will change the logical relation between elements of the same logical degeneracy class.

For example, one could choose the recovery maps for the 3-qubit code to be $\mathbb{V}_1 = \{III, XII, IYI, IIY\}$, in which case there are 3 degeneracy classes under IID noise. We could also select $\mathbb{V}_2 = \{III, XII, IXI, IIX\}$ which only has 2 degeneracy classes under IID noise. This apparent discrepancy occurs because IYI and IIY are logically degenerate to IXI and IIX respectively, which are degenerate to XII under IID noise by corollary 4. Both $\mathbb{V}_1$ and $\mathbb{V}_2$ have 2 logical degeneracy classes. A similar situation arises for the other codes, which explains the discrepancy between the 7 nondegenerate recovery maps observed in ref. [30] for the Steane code, and the 5 nondegenerate recovery maps proven for IID unitary noise in ref. [21]. Ref. [21] speculated that the discrepancy was due to the restriction to noise models with a single Kraus operator. Instead, the discrepancy is due to the selection of recovery operations; in [30], elements from 5 different equivalence classes were present in the set of recovery operations, whereas in [21] the set of recovery operations chosen contained elements of 7 equivalence classes.

If the noise commutes with additional logical operators, the number of logically nonde-

generate recovery maps is further decreased. A common example is IID depolarizing noise $\mathcal{N} = \mathcal{D}_p^{\otimes n}$, where $\mathcal{D}_p(\rho) = p\rho + (1-p)I/2$ is the single-qubit depolarizing channel with parameter p. For any single-qubit unitary (or unital[2]) channels $\mathcal{U}_0, \ldots, \mathcal{U}_{n-1}$, the composite channel $\otimes_j \mathcal{U}_j$ commutes with $\mathcal{D}_p^{\otimes n}$. In particular, let $Q = \sqrt{Z}\sqrt{X}$, which maps $X \to Y \to Z \to X$. For the 5-qubit and Steane codes, we have $Q^{\otimes n} \in \mathbb{GL}(U)$, where $\bar{Q}$ implements the same operation on the logical space, that is, it maps $\bar{X} \to \bar{Y} \to \bar{Z}$. The set $\mathbb{V}_{\text{IID}}$ of representative elements for the degeneracy classes under IID noise can be chosen to be $\{I, X_0, Y_0, Z_0\}$ for the 5-qubit code and $\{I, X_0, Y_0, Z_0, X_0Z_1\}$ for the Steane code. We can map Y_0 and Z_0 to X_0 by applying powers of $Q^{\otimes n}$ and so the representative elements of the logical degeneracy classes under IID depolarizing noise can be chosen to be $\{I, X_0\}$ for the 5 qubit code and $\{I, X_0, X_0Z_1\}$ for the Steane code. For the 5 qubit code, $\dot{\mathcal{N}}_U(R) = \bar{\mathcal{D}}_p$ for any $R \in \{X_0, Y_0, Z_0\}$, so that the logical operations (that is, $\bar{Q}$ and $\bar{Q}^\dagger$) commute with $\dot{\mathcal{N}}_U(R)$ and the representatives of the *degeneracy* classes can also be chosen to be $\{I, X_0\}$. The same property does not hold for the Steane code. Some examples of symmetry operators for the 3-, 5-, and 7-qubit codes are given in tables 2.1 and 2.2. Note that these operators can be used to generate equivalence classes whenever they commute with the physical noise map, that is, they do not necessarily require IID noise. For example, the $(0\,4)(1\,3)$[3] operator can be used to partition the recovery operators in the 5-qubit code when the physical noise has the form $\mathcal{N} = \mathcal{E} \otimes \mathcal{F} \otimes \mathcal{G} \otimes \mathcal{F} \otimes \mathcal{E}$, where $\mathcal{E}$, $\mathcal{F}$ and $\mathcal{G}$ are arbitrary single-qubit noise maps. We explore symmetries for non-IID noise in more detail in section 2.5.4.

2.5.1 Noisy readout measurements in stabilizer codes

Recall that syndrome measurements are used to project onto cospaces of stabilizer codes. A syndrome measurement is typically performed by coupling the encoded system to a readout system via a syndrome extraction circuit and then measuring the state of the readout system to implement an indirect measurement of a stabilizer generator, as in fig. 1.6. The measurement outcome for a complete syndrome measurement is the bit string $m \in \mathbb{Z}_2^{n-k}$ formed from the bits retrieved from each stabilizer generator measurement. In this section, we show that for syndrome measurements where the physical measurement undergoes measurement noise,

[2] A unital channel is one which leaves the identity matrix invariant.

[3] Many of these operators are permutations, which we denote using cycle notation wherein orbits created by the permutation operators are grouped in rounded brackets with qubit labels ordered to convey the applied permutation, e.g. $(1\,4\,3)$ maps the 1^{st} qubit to the 4^{th} qubit, the 4^{th} qubit to the 3^{rd} qubit, and the 3^{rd} qubit to the 1^{st} qubit. When a permutation contains multiple orbits, they can be written consecutively to denote the composition of the permutations that apply each orbit.

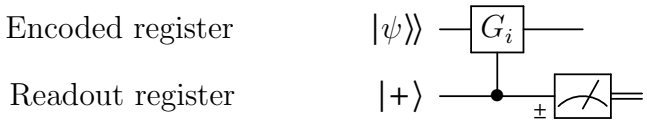


Figure 2.1: Indirect Weyl measurement using the $\{|+\rangle, |-\rangle\}$ basis.

the effective noise on the syndrome extraction is to probabilistically misreport syndromes according to a logical confusion matrix. The logical confusion matrix can be constructed as a product of elements of the confusion matrices that describe the individual measurements on the readout space. Folding the Fourier transforms into the state preparation and measurement, we obtain the syndrome extraction circuit for a single syndrome bit with measurement in the $\{|+\rangle, |-\rangle\}$ basis, i.e. measurement of an X operator on the readout space.

	$\|+\rangle$	$\|-\rangle$
$\langle+\|$	a	$1-b$
$\langle-\|$	$1-a$	b

Table 2.3: A confusion matrix for a noisy measurement in the X-basis where a is the probability of correctly measuring $|+\rangle$ and b is the probability of correctly measuring $|-\rangle$.

A destructive noisy X-basis measurement $\Theta(\mathbb{M}_{\pm})$ described by a confusion matrix C can be expressed as a set of Kraus operators, $\{\tilde{\pi}(X)_{i,j}\}$,

$$\tilde{\pi}(X)_{i,j} = \frac{1}{2}\sqrt{C_{i,j}}\,\langle j|\,, \tag{2.19}$$

where $i, j \in \{+, -\}$ with i the observed outcome and j the correct outcome. We retrieve Kraus operators for the effective map on the encoded qubits by noting that the readout qubit starts in the state $|+\rangle\langle+|$ so that we can write the Kraus operators $\{\tilde{\pi}(G_i)_{m_i,j}\}$ of the noisy indirect measurement $\Theta(\Pi(G_i)_{m_i}) : \mathbb{K} \to \mathbb{K}$ of generator G_i with observed outcome m_i and correct outcome j as

$$\tilde{\pi}(G_i)_{m_i,j} = \frac{1}{2}\sqrt{C_{m_i,j}}\,(I \otimes \langle j|)\,\hat{G}_i\,(I \otimes |+\rangle)\,. \tag{2.20}$$

Recalling that $\hat{G}_i = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes G_i$, we can simplify to

$$\tilde{\pi}(G_i)_{m_i,j} = \frac{1}{2}\sqrt{C_{m_i,j}}(I + (-1)^{\gamma(j)}G_i), \tag{2.21}$$

where $\gamma(+) = 0$ and $\gamma(-) = 1$. From here on, we use $\{0,1\}$ to label outcomes $\{+,-\}$ according to the mapping defined by $\gamma(\cdot)$. An ideal measurement is a special case of the above derivation, with $C_{i,j} = \delta_{i,j}$ so that the ideal measurement operators for a single syndrome bit are $\pi(G_i)_0 = (I + G_i)/2$ and $\pi(G_i)_1 = (I - G_i)/2$. Then we can express the Kraus operators for a noisy measurement in terms of the Kraus operators of an ideal measurement as follows

$$\tilde{\pi}(G_i)_{m_i,j} = \frac{1}{2}\sqrt{C_{m_i,j}}\pi(G_i)_j. \tag{2.22}$$

The noisy channel for the measurement of a single syndrome bit with observed outcome m_i is the sum over a correct and incorrect measurement with outcome m_i,

$$\Theta(\Pi(G_i)_{m_i})[\rho] = \sum_{j\in\mathbb{Z}_2} \tilde{\pi}(G_i)_{m_i,j}\rho\tilde{\pi}(G_i)_{m_i,j} \tag{2.23}$$

$$= \sum_{j\in\mathbb{Z}_2} C_{m_i,j}\pi(G_i)_j\rho\pi(G_i)_j \tag{2.24}$$

$$= \sum_{j\in\mathbb{Z}_2} C_{m_i,j}\Pi(G_i)_j[\rho]. \tag{2.25}$$

A complete noisy syndrome measurement with outcome m is implemented by $\Theta(\Pi_m) = \Theta(\Pi_{m_{n-k}}) \circ ... \circ \Theta(\Pi_{m_2}) \circ \Theta(\Pi_{m_1})$, so that

$$\Theta(\Pi_m)[\rho] = \sum_{j\in\mathbb{Z}_2^{n-k}} \left(\prod_{i=1}^{n-k} \tilde{\pi}(G_i)_{m_i,j_i}\right)\rho\left(\prod_{i=1}^{n-k} \tilde{\pi}(G_i)_{m_i,j_i}\right) \tag{2.26}$$

$$= \sum_{j\in\mathbb{Z}_2^{n-k}} \left(\prod_{i=1}^{n-k} C^{(i)}_{m_i,j_i}\pi(G_i)_{j_i}\right)\rho\left(\prod_{i=1}^{n-k} \pi(G_i)_{j_i}\right), \tag{2.27}$$

$$\tag{2.28}$$

where $C^{(i)}$ is the confusion matrix for the i^{th} readout qubit. Examining eq. (2.28), we can see that the noisy projector onto a cospace associated with syndrome m can be expressed as a probabilistic sum over ideal projections onto the cospaces of the code,

$$\Theta(\Pi_m)[\rho] = \sum_{j\in\mathbb{Z}_2^{2^{n-k}}} \bar{C}_{j,m}\Pi_j[\rho], \tag{2.29}$$

where

$$\bar{C}_{j,m} = \prod_{i=1}^{n-k} C^{(i)}_{m_i,j_i} \tag{2.30}$$

forms a confusion matrix for the logical measurement noise.

2.5.2 Symmetries of the toric code

We now show how our results can be applied to surface codes by considering the toric code as described in fig. 2.2. We do not fully specify an upper bound on the number of logically nondegenerate Pauli recovery maps because it depends on the number of rows and columns and involves high-weight recovery maps for large codes. Instead, we focus on weight 1 and weight 2 recovery maps.

The torus is constructed from a rectangular lattice by identifying the top and bottom edges and then the left and right edges, which imposes periodic boundary conditions. Therefore $\bar{Z}_1$ ($\bar{Z}_2$) can be moved to be any dark row (column) by multiplying the illustrated choice by Z stabilizers along the rows (columns). Similarly, $\bar{X}_1$ ($\bar{X}_2$) can be moved to any light column (row) by multiplying the illustrated choice by X stabilizers along the columns (rows). Therefore rotating or twisting the torus by 2 units to map dark lines to dark lines (horizontal or vertical translations with periodic boundary conditions) will permute the elements of $\mathbb{GS}(U)$ and $\mathbb{L}(U)$, where the logical operation will simply be an identity. That is, twists that map $(i,j) \to (i+2,j)$ and rotations that map $(i,j) \to (i,j+2)$ are in the symmetry group of the $r \times c$ Toric code. We include below a proof. Using rotations and twists, we can map any weight 1 Pauli to a weight 1 Pauli that acts on one of the qubits at $(0,1)$ or $(1,0)$. Therefore, by corollary 4, there will be at most 6 degeneracy classes of weight 1 Pauli errors under IID noise instead of 3n.

Lemma 6. *Vertical translation via* $VertT : (i,j) \to (i+2,j)$ *and horizontal translation via* $HorizT : (i,j) \to (i,j+2)$ *each preserve the toric code under IID noise.*

Proof. By definition, a vertical translation operation, $VertT : (i,j) \to (i+2,j)$, is a permutation operator, making it a unitary operation which commutes with noise that acts in the same way on every physical qubit. Then if $VertT[\mathbb{S}\overline{A}] = \mathbb{S}\overline{A}$ for all logical operations $\overline{A}$, lemma 6 holds as the codespace is preserved.

Applying $VertT$ to an X stabilizer has the following effect:

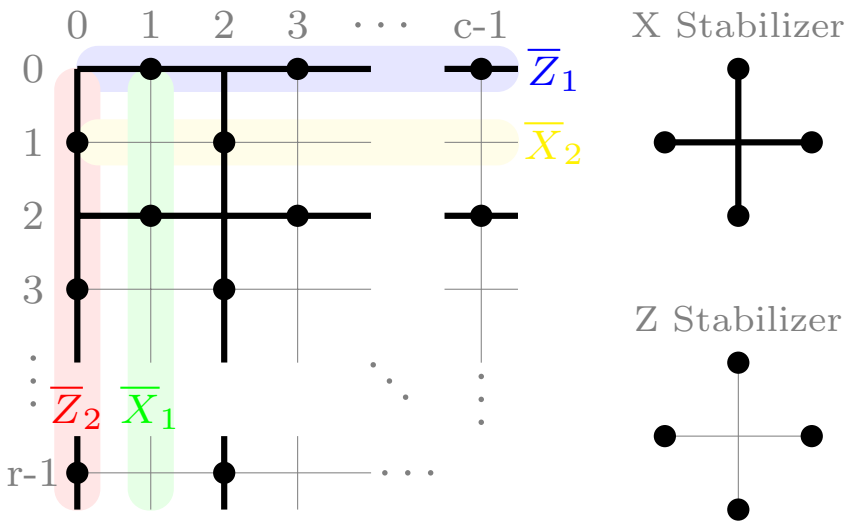


Figure 2.2: A toric code is specified by a number of rows, $r \in 2\mathbb{Z}$, and a number of columns, $c \in 2\mathbb{Z}$, with qubits lying on the edges of an $r \times c$ grid indexed by $\{(i,j) : i \neq j, i \in \mathbb{Z}_r, j \in \mathbb{Z}_c\}$, as depicted above. The qubit in row i and column j is indexed by (i,j). The circles represent physical qubits, and the diagrams of X and Z stabilizers show which qubits these operators act on, with every qubit on the edge of an X (Z) stabilizer acted on by an X (Z) Pauli operator. The X stabilizers for the toric code are given by $\mathbb{X} = \{X(i,j) : i \in 2\mathbb{Z}_{r/2}, j \in 2\mathbb{Z}_{c/2}\}$ and the Z stabilizers by $\mathbb{Z} = \{Z(i,j) : i \in 2\mathbb{Z}_{r/2}+1, j \in 2\mathbb{Z}_{c/2}+1\}$, where $A(i,j) = A_{i-1,j} \otimes A_{i+1,j} \otimes A_{i,j-1} \otimes A_{i,j+1}$. The logical operators are $\bar{Z}_1 = \bigotimes_{j \in 2\mathbb{Z}_{c/2}+1} Z_{0,j}$, $\bar{Z}_2 = \bigotimes_{i \in 2\mathbb{Z}_{r/2}+1} Z_{i,0}$, $\bar{X}_1 = \bigotimes_{j \in 2\mathbb{Z}_{r/2}} X_{i,1}$, and $\bar{X}_2 = \bigotimes_{j \in 2\mathbb{Z}_{c/2}} X_{1,j}$. Note that all operations on the row (column) index are taken modulo r (c), corresponding to periodic boundary conditions. The stabilizers are generated by $\mathbb{X} \cup \mathbb{Z}$, and a minimal generating group can be achieved by removing one X stabilizer and one Z stabilizer from $\mathbb{X} \cup \mathbb{Z}$.

Code	Map Name	Symmetry Operation	Logical Operation
$r \times c$ Toric	Vertical Translation	$(i,j) \to (i+2,j) \forall i,j$	NA
	Horizontal Translation	$(i,j) \to (i,j+2) \forall i,j$	NA
	Vertical Reflection	$(i,j) \to (-i,j) \forall i,j$	NA
	Horizontal Reflection	$(i,j) \to (i,-j) \forall i,j$	NA
	$\mathcal{J}$	$\mathcal{F}^{\otimes n} \circ (i,j) \to (i+1,j+1) \forall i,j$	$\overline{F}^{\otimes 2} \circ \overline{\text{SWAP}}$
$r \times r$ Toric	Diagonal Reflection	$(i,j) \to (j,i) \forall i,j$	$\overline{\text{SWAP}}$

Table 2.4: Symmetry operations for the toric code, along with the associated logical operations. These operations can be applied for general IID noise, with the exception of $\mathcal{J}$, which requires noise that commutes with the n-qubit Hadamard gate as well as the permutation, for example, $\mathcal{J}$ can be applied to a toric code undergoing IID depolarizing noise. Note that a vertical translation is also called a twist and a horizontal translation is also called a rotation.

$$VertT[X(i,j)] = X_{m-1,j} \otimes X_{m+1,j} \otimes X_{m,j-1} \otimes X_{m,j-1}, \tag{2.31}$$

where $m = i + 2 \pmod r$, and the modulo r arises due to the periodic boundary of the toric code. Noting that $\{m = (i+2) \bmod r : i \in 2\mathbb{Z}_{r/2}\} = \{i : i \in 2\mathbb{Z}_{r/2}\}$ and comparing $VertT[X(i,j)]$ to the form of $X(i,j)$, we see that $VertT$ leaves the X stabilizers invariant. A similar exercise confirms that the Z stabilizers are similarly preserved, and therefore $VertT$ leaves the stabilizer group invariant.

Applying $VertT$ to $\overline{X_1}$ and $\overline{Z_2}$ leaves them invariant; $VertT$ therefore leaves $\mathbb{S}\overline{X_1}$ and $\mathbb{S}\overline{Z_2}$ invariant.

For the other logical operations,

$$VertT[\overline{X_2}] = \bigotimes_{j \in 2\mathbb{Z}_{c/2}} X_{3,j} \tag{2.32}$$

and

$$VertT[\overline{Z_1}] = \bigotimes_{j \in 2\mathbb{Z}_{c/2}+1} X_{2,j}. \tag{2.33}$$

Then, $\overline{X_2} = M\, VertT[\overline{X_2}]$, and $\overline{Z_1} = N\, VertT[\overline{Z_1}]$ for

$$M = \bigotimes_{u \in 2\mathbb{Z}_{c/2}} X(2,u), \tag{2.34}$$

$$N = \bigotimes_{w \in 2\mathbb{Z}_{c/2}+1} Z(1,w). \tag{2.35}$$

Because $M \in \mathbb{S}$,

$$\begin{aligned} VertT[\mathbb{S}\overline{X_2}] &= \{ VertT[S\overline{X_2}] : S \in \mathbb{S}\} \\ &= \{ VertT[S]\, VertT[\overline{X_2}] : S \in \mathbb{S}\} \\ &= \{SM\overline{X_2} : S \in \mathbb{S}\} \\ &= \mathbb{S}\overline{X_2}. \end{aligned} \tag{2.36}$$

A similar procedure using N rather than M shows that $VertT$ preserves $\mathbb{S}\overline{Z_1}$, and for a horizontal translation, an analogous proof demonstrates that $HorizT : (i,j) \to (i,j+2)$ preserves the toric code when every physical qubit undergoes the same noise process.

□

We can also combine a rotation by 1, a twist by 1 (i.e. mapping all physical qubits by $\mathcal{H} : (i,j) \to (i+1, j+1)$), and $\mathcal{F}^{\otimes n}$, where F is the Hadamard gate. This operation $\mathcal{J} = \mathcal{F}^{\otimes n}\mathcal{H}$ preserves the stabilizer group by mapping X stabilizers to Z stabilizers and vice versa, $\bar{X}_1 \leftrightarrow \bar{Z}_2$, and $\bar{X}_2 \leftrightarrow \bar{Z}_1$, and so implements a logical SWAP combined with a Hadamard gate on each logical qubit. A proof is included below. Therefore for noise that commutes with $\mathcal{J}$ (e.g., IID depolarizing noise), there are at most 4 logically nondegenerate weight 1 Pauli recovery maps, namely, $\{X_{0,1}, Z_{0,1}, Y_{0,1}, Y_{1,0}\}$.

Lemma 7. *A logical Hadamard gate can be implemented simultaneously on both logical qubits in a toric code by applying $\mathcal{J} = \mathcal{F}^{\otimes n}\mathcal{H}$.*

Proof. Applying $\mathcal{J}$ has the following effect on the X and Z stabilizers:

$$\begin{aligned} \mathcal{J}[\mathbb{X}] &= \{Z(i+1, j+1) : i \in 2\mathbb{Z}_{r/2}, j \in 2\mathbb{Z}_{c/2}\} \\ &= \{Z(i,j) : i \in 2\mathbb{Z}_{r/2}+1, j \in 2\mathbb{Z}_{c/2}+1\} \\ &= \mathbb{Z} \end{aligned} \tag{2.37}$$

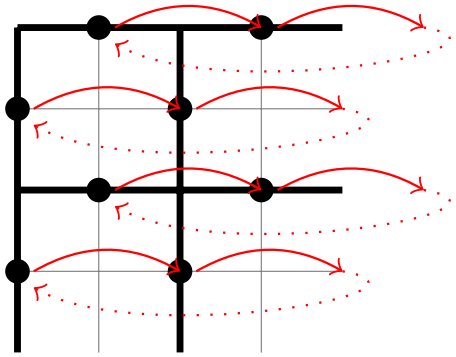


Figure 2.3: Horizontal translation symmetry in the toric code.

$$\begin{aligned}\mathcal{J}[\mathbb{Z}] &= \{X(i+1,j+1) : i \in 2\mathbb{Z}_{r/2}+1, j \in 2\mathbb{Z}_{c/2}+1\} \\ &= \{X(i,j) : i \in 2\mathbb{Z}_{r/2}+2, j \in 2\mathbb{Z}_{c/2}+2\} \\ &= \{X(i,j) : i \in 2\mathbb{Z}_{r/2}, j \in 2\mathbb{Z}_{c/2}\} \\ &= \mathbb{X} \end{aligned} \tag{2.38}$$

so the stabilizers are preserved by $\mathcal{J}$. Applying $\mathcal{J}$ to $\overline{Z}_1$,

$$\begin{aligned}\mathcal{J}[\overline{Z}_1] &= \bigotimes_{j \in 2\mathbb{Z}_{c/2}+1} X_{1,j+1} \\ &= \bigotimes_{j \in 2\mathbb{Z}_{c/2}} X_{1,j} \\ &= \overline{X}_2. \end{aligned} \tag{2.39}$$

An analogous procedure shows that $\mathcal{J}[\overline{Z}_2] = \overline{X}_2$. Applying $F^{\otimes n}H$ to $\overline{X}_2$,

$$\begin{aligned}\mathcal{J}[\overline{X}_2] &= \bigotimes_{j \in 2\mathbb{Z}_{c/2}} Z_{2,j+1} \\ &= \bigotimes_{j \in 2\mathbb{Z}_{c/2}+1} Z_{2,j} \\ &= S\overline{Z}_1 \end{aligned} \tag{2.40}$$

for some $S \in \mathbb{S}$. A similar procedure yields $\mathcal{J}[\overline{X}_2] = S\overline{Z}_2$ for some $S \in \mathbb{S}$. Then $\mathcal{J}$ maps $\overline{X}_2 \leftrightarrow \overline{Z}_1$ and $\overline{X}_1 \leftrightarrow \overline{Z}_2$ modulo stabilizers, thereby implementing a Hadamard on both logical qubits. □

From the periodic boundary conditions, we can also reflect vertically or horizontally by mapping $(i,j) \to (-i,j)$ or $(i,j) \to (i,-j)$, respectively. These reflections will permute the X stabilizers and the Z stabilizers and will either leave the logical operators invariant or map them to a different row/column, which is equivalent to the original logical operator up to a product of stabilizers. Therefore these reflections are elements of the (general) stabilizer group $\mathbb{GS}(U)$.

Lemma 8. *Vertical and horizontal reflection preserve the toric code under IID noise.*

Proof. Applying a vertical reflection, $VertR : (i,j) \to (-i,j)$, to an X stabilizer, $X(i,j)$,

$$VertR[X(i,j)] = X_{m-1,j} \otimes X_{m+1,j} \otimes X_{m,j-1} \otimes X_{m,j+1}, \tag{2.41}$$

where $m = r - i \pmod r$. Noting that $\{m = r - i \text{ (mod r)} : i \in 2\mathbb{Z}_{r/2}\} = \{i : i \in 2\mathbb{Z}_{r/2}\}$, we see that a vertical reflection preserves the X stabilizers. A similar exercise shows that vertical reflection preserves the Z stabilizers, and thereby preserves the stabilizer group.

Applying a vertical reflection to the logical operators leaves all but $\overline{X_2}$ invariant, so that $VertR[\mathbb{S}\overline{X_1}] = \mathbb{S}\overline{X_1}$, $VertR[\mathbb{S}\overline{Z_1}] = \mathbb{S}\overline{Z_1}$, and $VertR[\mathbb{S}\overline{Z_2}] = \mathbb{S}\overline{Z_2}$. Applying a vertical reflection to $\overline{X_2}$,

$$\begin{aligned} VertR[\overline{X_2}] &= \bigotimes_{j \in 2\mathbb{Z}_{c/2}} X_{r-1,j} \\ &= P\overline{X_2}, \end{aligned} \tag{2.42}$$

where

$$P = \bigotimes_{\substack{u \in 2\mathbb{Z}_{r/2} \setminus 0 \\ w \in 2\mathbb{Z}_{c/2}}} X(u,w). \tag{2.43}$$

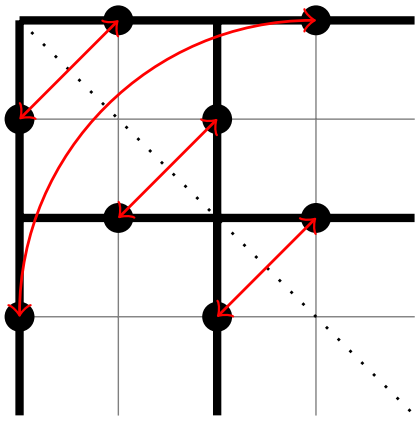


Figure 2.4: Diagonal reflection symmetry in the toric code.

Because $P \in \mathbb{S}$,

$$\begin{aligned} \mathit{VertR}[\mathbb{S}\overline{X_1}] &= \{\mathit{VertR}[S\overline{X_1}] : S \in \mathbb{S}\} \\ &= \{\mathit{VertR}[S]\,\mathit{VertR}[\overline{X_1}] : S \in \mathbb{S}\} \\ &= \{SP\overline{X_1} : S \in \mathbb{S}\} \\ &= \mathbb{S}\overline{X_1}. \end{aligned} \tag{2.44}$$

A similar procedure shows that the toric code is preserved by a horizontal reflection, HorizR : $(i, j) \to (i, -j)$. □

An $r \times r$ toric code can also be reflected across a diagonal axis via $(i, j) \to (j, i)$. This reflection permutes the stabilizers and maps $\bar{X}_1 \leftrightarrow \bar{X}_2$ and $\bar{Z}_1 \leftrightarrow \bar{Z}_2$ up to a product of stabilizers. Diagonal reflection is therefore an element of the logical symmetry group of an $r \times r$ toric code, which implements a logical SWAP gate.

Lemma 9. *A logical SWAP gate can be implemented in an $r \times r$ toric code by permuting the physical qubits by rotating clockwise by 90°or via a reflection through an axis 45°clockwise from the horizontal.*

Proof. Applying a clockwise rotation by 90°, Rot90: $(i, j) \to (j, -i)$, to an X stabilizer yields

$$\mathrm{Rot90}[X(i,j)] = X_{j,m-1} \otimes X_{j,m+1} \otimes X_{j-1,m} \otimes X_{j+1,m}, \tag{2.45}$$

where $m = r - i$; because we apply operations to the row index modulo r, $-i = r - i = m$. Then $\mathbb{X} = \{\mathrm{Rot90}[X(i,j)] : i,j \in \mathbb{Z}_r\}$. A similar argument shows that Rot90 preserves $\mathbb{Z}$. Applying Rot90 to the logical operators,

$$\mathrm{Rot90}[\overline{Z}_1] = \bigotimes_{j \in 2\mathbb{Z}_{r/2}+1} Z_{j,0} = \overline{Z}_2 \tag{2.46}$$

$$\mathrm{Rot90}[\overline{Z}_2] = \bigotimes_{i \in 2\mathbb{Z}_{r/2}+1} Z_{0,r-i} = N\overline{Z}_1 \tag{2.47}$$

$$\mathrm{Rot90}[\overline{X}_2] = \bigotimes_{j \in 2\mathbb{Z}_{r/2}} X_{j,r-1} = M\overline{X}_1 \tag{2.48}$$

$$\mathrm{Rot90}[\overline{X}_1] = \bigotimes_{j \in 2\mathbb{Z}_{r/2}} X_{1,r-j} = \overline{X}_2, \tag{2.49}$$

where $N, M \in \mathbb{S}$

□

For any Pauli recovery map of a fixed weight w, we can use rotations and twists to map one of the qubits P acts nontrivially on to either $(0,1)$ or $(1,0)$. Therefore the number of nondegenerate recovery maps is reduced by a factor of approximately $n/2$, although the exact reduction factor introduced by translational symmetry depends upon w. We can further reduce the number of nondegenerate Pauli recovery maps using another symmetry of the toric code, namely, horizontal and vertical reflections about any row or column. Translational symmetries combined with horizontal and vertical reflection reduce the $9\binom{n}{2}$ weight 2 Pauli recovery maps to at most $9(\frac{n+c+r}{2} - 2)$ nondegenerate recovery maps, which is approximately $9(n\text{-}1)/2$ up to edge effects. The following explains where this number comes from. Using translation, we can map any weight 2 Pauli recovery map to have weight on either $(0,1)$ or $(1,0)$. There are $n-1$ coordinate pairs containing each of these origins. Consider the coordinate pairs of the form $\{(0,1),(i,j)\}$. Without loss of generality, we can use a vertical reflection—that is, $(i,j) \to (r-i,j)$—so that $i \in [0, r/2]$. We can then use a horizontal reflection and a rotation—that is, $(i,j) \to (i, 2-j)$—so that $j \in [1, c/2+1]$. There are then $\frac{(r/2+1)\times(c/2+1)}{2} - 1$ coordinate pairs containing $(0,1)$ as we have reduced to a $(c/2+1) \times (r/2+1)$ grid with qubits on half of the locations, and we subtract the

location of $(0,1)$. A similar reduction of coordinate pairs containing $(1,0)$ using horizontal reflection and vertical reflection with a twist of the form $(i,j) \to (2-i,j)$ produces pairs in $\{\{(1,0),(i,j)\} : i \in [1, r/2+1], j \in [0, c/2]\}$. Adding the sizes of these two sets of coordinate pairs, we get $\frac{rc/2+r+c}{2} - 1$. Recall that $rc/2 = n$ to get $\frac{n+r+c}{2} - 1$. There is one additional symmetry that occurs using these operations which is not covered by the above counting argument: there is a coordinate pair in each reduced set which, by undoing some of the operations used, maps to $\{(0,1),(1,0)\}$. as such, we can subtract one case. Then we multiply by 9 for the selection of a weight 2 element in $\mathbb{P}_2$. Further reductions are possible using, e.g., a diagonal reflection for general IID noise in a square lattice or $\mathcal{J}$ for depolarizing noise.

2.5.3 Symmetries for multiple rounds of QEC

For a memory channel in which a state is stored for a prolonged period, we expect to need to apply multiple rounds of error correction as the noise continues to act on the state. A memory channel with m QEC steps and corresponding m recovery operations R_i and noise channels $\mathcal{N}_i$ can be expressed as

$$\dot{\mathcal{N}}_U(\{(R_i, \mathcal{N}_i)\}) = \mathcal{U}^\dagger \left(\prod_{i=1}^{m} \mathcal{R}_i^\dagger \mathcal{N}_i \right) \mathcal{U}. \tag{2.50}$$

Because we have allowed the noise acting at each time step to be distinct for generality, we also assume that the set of recovery maps used at each time step will be distinct so that each R_i is drawn from a set $\mathbb{V}_i$ of recovery operations for that time step. We now show how to get degeneracies and logical degeneracies in the case where a channel has the above form.

Corollary 10. *Let $U \in \mathbb{U}(\mathbb{H}, \mathbb{K})$, $R_i \in \mathbb{V}_i$, and $\mathcal{N}_i \in \mathbb{B}(\mathbb{B}(\mathbb{K}))$ and define $\mathbb{C}(\mathcal{N}) = \{\mathcal{A} \in \mathbb{U}(\mathbb{B}(\mathbb{K})) : [\mathcal{A}, \mathcal{N}] = 0\}$ The maps $\{\dot{\mathcal{N}}_U(\{(A_i^\dagger R_i A_{i+1}, \mathcal{N}_i)\}) : \mathcal{A}_1 \in \mathbb{GL}_K(U, \mathcal{N}_1), \mathcal{A}_{i \notin \{1,m+1\}} \in \mathbb{C}(\mathcal{N}_i), \mathcal{A}_{m+1} \in \mathbb{GL}_K(U)\}$ are logically degenerate. Furthermore, the maps $\{\dot{\mathcal{N}}_U(\{(A_i^\dagger R_i A_i, \mathcal{N}_{i+1})\}) : \mathcal{A}_1 \in \mathbb{S}_K(U, \mathcal{N}_1), \mathcal{A}_{i \notin \{1,m+1\}} \in \mathbb{C}(\mathcal{N}_i), \mathcal{A}_{m+1} \in \mathbb{S}_K(U)\}$ are degenerate.*

Proof.

$$\bar{\mathcal{A}}_{m+1}^{\dagger}\dot{\mathcal{N}}_U(\{(R_i,\mathcal{N}_i)\})\bar{\mathcal{A}}_1 = \mathcal{U}^{\dagger}\mathcal{A}_{m+1}^{\dagger}\left(\prod_{i=1}^{m}\mathcal{R}_i^{\dagger}\mathcal{N}_i\right)\mathcal{A}_1\mathcal{U} \tag{2.51}$$

$$= \mathcal{U}^{\dagger}\mathcal{A}_{m+1}^{\dagger}\left(\prod_{i=1}^{m}\mathcal{R}_i^{\dagger}\mathcal{N}_i\mathcal{A}_i\mathcal{A}_i^{\dagger}\right)\mathcal{A}_1\mathcal{U} \tag{2.52}$$

$$= \mathcal{U}^{\dagger}\mathcal{A}_{m+1}^{\dagger}\left(\prod_{i=1}^{m}\mathcal{R}_i^{\dagger}\mathcal{A}_i\mathcal{N}_i\mathcal{A}_i^{\dagger}\right)\mathcal{A}_1\mathcal{U} \tag{2.53}$$

$$= \mathcal{U}^{\dagger}\left(\prod_{i=1}^{m}\mathcal{A}_{i+1}^{\dagger}\mathcal{R}_i^{\dagger}\mathcal{A}_i\mathcal{N}_i\right)\mathcal{U} \tag{2.54}$$

$$= \dot{\mathcal{N}}_U(\{(A_i^{\dagger}R_iA_{i+1},\mathcal{N}_i)\}) \tag{2.55}$$

□

2.5.4 Symmetries for Non-IID Noise

Though we have restricted attention in the examples thus far to IID noise in stabilizer codes, it should be noted that proposition 2, corollary 3, corollary 4, and corollary 10 apply equally to codes undergoing non-IID noise. In the case of non-IID noise, logically degenerate or strictly degenerate recovery operations can be calculated just as in previous sections. However, the form of the noise must be taken into account when finding (logically) degenerate maps, as commutation relations are not quite as simple for non-IID noise. If the non-IID noise commutes with a symmetry operator, it can be applied in the same way as in all of the examples thus far. For example, in a 5-qubit code undergoing noise of the form $\mathcal{N} = \mathcal{N}_0 \otimes \mathcal{N}_1 \otimes \mathcal{N}_0 \otimes \mathcal{N}_1 \otimes \mathcal{N}_0$, the permutation operator $(0\,4)(1\,3)$ commutes with the noise and can thus be used to partition recovery operations, while $(0\,1\,2\,3\,4)$ does not preserve the noise and therefore cannot be used to generate symmetries. Then, using single-qubit X operators as an example, $X_0 \cong X_4 \neq X_1 \cong X_3$, where we use $\cong$ to denote operators in the same degeneracy class, and $\neq$ to denote a separation between degeneracy classes.

To find degeneracies when the physical noise does not commute with a symmetry operator, we must examine the effects of symmetries on the joint map $\mathcal{M}\mathcal{N}$, as in proposition 2. For perfect measurements, this corresponds to finding degeneracies in maps of the form $\mathcal{R}^{\dagger}\mathcal{N}$, where $\mathcal{R}^{\dagger}$ is a recovery map. To find recovery maps which produce the same effective noise under a permutation operator, for example, we must simultaneously permute $\mathcal{N}$ and the recovery operation in question. This will not necessarily help in reducing computational

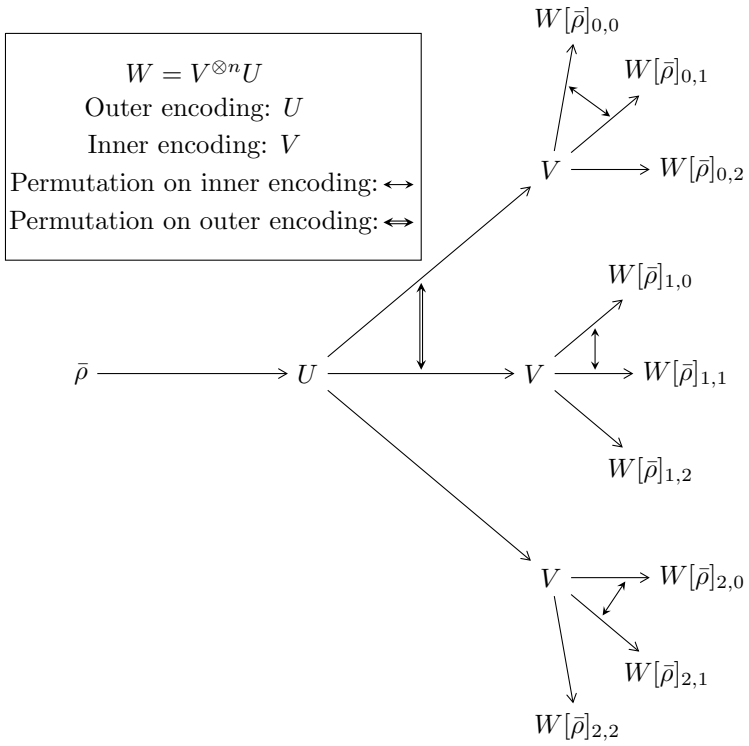


Figure 2.5: Tree structure of a concatenated code with outer encoding U and inner encoding V. The leaves on the right represent physical qubits. The subscripts in $W[\bar{\rho}]_{i,j}$ denote that the physical qubit is the jth qubit in the inner encoding (V) of the ith qubit in the outer encoding (U). Double-ended arrows depict the action of a permutation which swaps the 0^{th} and 1^{st} qubit on the inner and outer encodings.

complexity for a fixed noise model, but comes in handy when handling, for example, permutations of noise occurring for different error paths in a concatenated code. Section 2.5.5 gives a more complete example of how this can be applied, using a concatenated 5-qubit code to illustrate, where the second level of concatenation sees non-IID noise even if the physical noise is IID. Note, however, that by the nature of concatenated codes, there are several permutations of noise at the second level of concatenation, allowing permutations to be applied more liberally to the noise model to find degeneracies between permutations of the noise as well as between recovery operations, whereas for a single noise model, any symmetry operator must preserve that single noise model.

2.5.5 Symmetries of concatenated stabilizer codes

A common method of improving the logical error rate is to concatenate QECCs. Let $U \in \mathbb{U}(\mathbb{H}, \mathbb{H}^{\otimes n})$ and $V \in \mathbb{U}(\mathbb{H}, \mathbb{H}^{\otimes m})$ be two encodings. Then $W = V^{\otimes n}U \in \mathbb{U}(\mathbb{H}, \mathbb{H}^{\otimes mn})$ is a concatenated encoding with inner (outer) encoding V (U), as illustrated in fig. 2.5. The number of recovery operators increases doubly exponentially in the number of levels of concatenation. However, we now show how corollary 4 and proposition 2 can be applied directly to concatenated codes to reduce the number of nondegenerate (or logically nondegenerate) recovery maps by concatenating symmetries of the inner and outer encodings. We apply corollary 4 at the first level as in previous examples, and in higher levels, use proposition 2 to apply symmetry operations. The reason proposition 2 is more applicable after the first encoding is that the effective "physical" noise for higher levels is no longer IID so the symmetry operations we've explored for IID noise no longer commute with the effective "physical" noises. We therefore must permute the noise and symmetry operations together to find degeneracy classes for different recovery operation/noise map pairs at each level.

To analyze concatenated codes, we first need to construct a set of recovery maps and the logical group. Let $\mathbb{V}_U$ and $\mathbb{V}_V$ be sets of recovery maps for U and V respectively. Then $\mathbb{V}_W = \{(\otimes_j F_j)\mathcal{V}^{\otimes n}[G] : F_0, \ldots, F_{n-1} \in \mathbb{V}_V, G \in \mathbb{V}_U\}$ is a set of recovery maps for W. To see this, let $R = (\otimes_j F_j)\mathcal{V}^{\otimes n}[G]$ and $R' = (\otimes_j F'_j)\mathcal{V}^{\otimes n}(G')$ be two elements of $\mathbb{V}_W$ and recall that the recovery maps for, e.g., U satisfy $U^\dagger Q^\dagger R U = \delta_{R,Q} 1_{\mathbb{H}}$. Then as $\mathbb{V}_U$ and $\mathbb{V}_V$ are sets of recovery maps for U and V, we have

$$\begin{aligned} W^\dagger R^\dagger R' W &= U^\dagger G^\dagger (\otimes_j V^\dagger F_j^\dagger F'_j V) G' U \\ &= U^\dagger B^\dagger (\otimes_j \delta_{F_j,F'_j} 1_{\mathbb{H}}) G' U \\ &= U^\dagger G^\dagger (1_{\mathbb{H}})^{\otimes n} G' U \prod_j \delta_{F_j,F'_j} \\ &= 1_{\mathbb{H}} \delta_{G,G'} \prod_j \delta_{F_j,F'_j} \\ &= 1_{\mathbb{H}} \delta_{R,R'} \end{aligned} \tag{2.56}$$

as required. Any recovery maps for U that are tensor products of logical operators for V (e.g. Pauli recovery maps in concatenated stabilizer codes) can be commuted through $V^{\otimes n}$.

We now show how to build some elements of the logical group $\mathbb{GL}(W)$, focusing on tensor product operations and the symmetry group under IID noise. Suppose that $\mathcal{A}_0, \ldots, \mathcal{A}_{n-1} \in \mathbb{GL}(V)$ and let $\bar{\mathcal{A}}_j$ be such that $\mathcal{A}_j \mathcal{V} = \mathcal{V} \bar{\mathcal{A}}_j$. Then, provided $(\otimes_j \bar{\mathcal{A}}_j) \in \mathbb{GL}(U)$ with

$(\otimes_j \bar{\mathcal{A}}_j)\mathcal{U} = \mathcal{U}\breve{\mathcal{A}}$, we have

$$(\otimes_j \mathcal{A}_j)\mathcal{W} = (\otimes_j \mathcal{A}_j \mathcal{V})\mathcal{U} = (\otimes_j \mathcal{V}\bar{\mathcal{A}}_j)\mathcal{U} \tag{2.57}$$

$$= \mathcal{V}^{\otimes n}\mathcal{U}\breve{\mathcal{A}} = \mathcal{W}\breve{\mathcal{A}}, \tag{2.58}$$

and so $(\otimes_j \mathcal{A}_j) \in \mathbb{GL}(W)$ for $\mathcal{A}_j \in \mathbb{GL}(V) \forall j$ such that $\mathcal{A}_j \mathcal{V} = \mathcal{V}\bar{\mathcal{A}}_j$ and $\otimes_j \bar{\mathcal{A}}_j \in \mathbb{GL}(U)$.

The permutation group of a concatenated code can be generated from the permutation groups of the inner and outer encodings by labeling each qubit as a pair of indices and permuting one index with the inner code's permutation group and the other index by the outer code's permutation group. For example, let $\mathcal{A} \in \mathbb{GL}(U)$ be a permutation of n qubits. Then we can commute $\mathcal{A}$ through $\mathcal{V}^{\otimes n}$ by permuting the tensor factors as illustrated in fig. 2.5.

If the permutation groups of both the inner and outer code in a concatenated code are transitive, then the concatenated code will also be transitive. This becomes evident when envisioning the action of permutation operators on the branches in fig. 2.5; if U is transitive then each major branch (those to the right of U but left of V) can be mapped to any other major branch. If V is transitive, each subbranch of the major branches (those to the right of each V) can be mapped to any other subbranch of the same branch. Then it follows that any physical qubit can be mapped to any other physical qubit when the codes being concatenated are transitive, that is, the concatenation of transitive codes will also be transitive, although the concatenation of 2-transitive codes will typically not be 2-transitive because the individual permutations cannot separate errors that act on the same inner code block.

For example, consider the 9-qubit Shor code, which can be regarded as a concatenated code with $U = |000\rangle\langle 0| + |111\rangle\langle 1|$ and $V = UF = |000\rangle\langle +| + |111\rangle\langle -|$. The Hadamard gate is introduced so that weight 1 Z errors, which are logical operators in the inner repetition code without the Hadamard, are mapped to X errors and so can be detected and corrected by the outer code. There are $2^8 = 256$ recovery maps for the 9-qubit Shor code. Any permutation of the 3 qubits for either U or V is in both $\mathbb{GS}(U)$ and $\mathbb{S}(V)$. Labelling the qubits by (i, j) where i labels the qubits in the inner encoding and j labels the qubits in the outer encoding, we can apply any permutation to j for each i independently (i.e., permutations on the inner encoding) and any permutation on j for all i (i.e., permutations on the outer encoding, which affect all corresponding qubits in the inner encoding). Because both the inner and outer encodings are transitive, the weight 1 Pauli recovery maps for the concatenated code fall into 3 degeneracy classes under IID noise with representative elements $X_{0,0}$, $Y_{0,0}$, and $Z_{0,0}$. The weight 2 Pauli recovery maps can be divided into two types. For the first type,

both nontrivial Pauli terms act on a qubit in the same inner encoding. By permuting the outer encoding, we can map the nontrivial Pauli terms to act on the first inner encoding. We can then permute the qubits in the inner encoding so that the nontrivial Pauli terms act on qubits $(0,0)$ and $(0,1)$. Multiplying by $Z_{0,0}Z_{0,1}$ (a stabilizer) and permuting qubits $(0,0)$ and $(0,1)$ as necessary, there are 3 nondegenerate weight 2 recovery maps of this type, namely, $P_{0,0}X_{0,1}$ for $P \in \{X,Y,Z\}$, where some of the other weight 2 recovery maps are in the same degeneracy class as I or weight 1 errors. For the second type, one nontrivial Pauli term acts on a qubit in each of two distinct inner encodings. By permuting the inner encodings, we can map any such Pauli recovery map to $P_{0,0}Q_{1,0}$ for $P,Q \in \mathbb{P}$. There are 9 such terms; however, we can swap the inner encodings to reduce to 5 terms (i.e., ZX and XZ are degenerate). Therefore there are at most 5 nondegenerate weight 2 recovery maps of this type. Then there are at most 12 nondegenerate Pauli recovery maps under IID noise (1 of weight 0, 3 of weight 1, and 8 of weight 2), compared to the total of 256 Pauli recovery maps selected for a given implementation.

The reduction in the number of recovery maps is more dramatic for larger concatenated codes and higher levels of concatenation. For the 5 qubit and Steane codes, there are $16^5 \approx 10^6$ and $64^7 \approx 4 \times 10^{12}$ recovery maps at the first level of concatenation respectively. To quickly remove many degenerate recovery maps, it is more convenient to use the recursive structure typically used in numerical studies. For noise with $\mathcal{N} = \mathcal{N}_1^{\otimes m}$ and a recovery map $R = (\otimes_j F_j)\mathcal{V}^{\otimes n}[G]$ where $F_0, \ldots, F_{n-1} \in \mathbb{V}_V$ and $G \in \mathbb{V}_U$, we then have

$$\begin{aligned}\dot{\mathcal{N}}_W(R) &= \mathcal{W}^\dagger \mathcal{R}^\dagger \mathcal{N}_1^{\otimes m} \mathcal{W} \\ &= \mathcal{U}^\dagger \mathcal{G}\left(\otimes_j \mathcal{V}^\dagger \mathcal{F}_j^\dagger \mathcal{N}_1 \mathcal{V}\right)\mathcal{U} \\ &= \mathcal{U}^\dagger \mathcal{G}\left(\otimes_j \bar{\mathcal{N}}_{1,V}(\mathcal{F}_j)\right)\mathcal{U}. \end{aligned} \tag{2.59}$$

That is, the logical noise map via eq. (2.4) for the concatenated code is simply the logical map for the outer code where the effective "physical" noise map is the effective logical noise for the inner code conditioned on the F_j. We then see that the number of logical nondegeneracy classes at the first level of a concatenated code will be at most $|\mathbb{V}_{V,\mathcal{N}_1}|^n \times |\mathbb{V}_U|$, where $\mathbb{V}_{V,\mathcal{N}_1}$ is the set of degeneracy classes under the noise process $\mathcal{N}_1$. For the 5 qubit and Steane codes with IID depolarizing noise, there are then at most $2^5 \times 16 = 512$ and $3^7 \times 64 = 34,992$ logical degeneracy classes.

We can further reduce the number of logical degeneracy classes by using the symmetry of the outer encoding and choosing the decoder correctly. Importantly, even if the physical noise is IID, the effective noise for the outer encoding conditioned on observed syndromes (i.e., the F_j in eq. (2.59)) will not be IID. Nevertheless, the set of effective noise processes

for the outer encoding is permutationally invariant with respect to the outer code. Consider two recovery maps $(\bigotimes_j F_j)\mathcal{V}^{\otimes n}[G]$ and $(\bigotimes_j F'_j)\mathcal{V}^{\otimes n}[G']$. If $\bigotimes_j F'_j$ can be obtained from $\bigotimes_j F_j$ by an element of the symmetry group, then an optimal decoder will set G' to be the same permutation of G as then the two recovery maps will have the same (and optimal) logical noise by corollary 4. Therefore instead of all the elements of $\mathbb{V}_{V,\mathcal{N}_1}^{\otimes n}$, we need only consider the representative elements under the symmetry group of U.

For example, for the 5-qubit code under IID depolarizing noise there are 2^5 effective noise processes $\otimes_j \bar{\mathcal{N}}_{1,V}(F_j)$ corresponding to the choices of $F_j \in \{I, X_{(j,0)}\}$, where we denote the operator that acts as X_j on the ith code block and identity on the other blocks by $X_{(i,j)}$. With this notation, because of the (0 1 2 3 4) symmetry, we only need to consider operations that act as $X_{(i,0)}$, resulting in 8 effective noise processes for an optimal decoder, namely those conditioned on the following recovery operators applied to the inner encoding:

$$\begin{aligned}
\mathbb{F}_0 &= \{I\} && \text{(no errors)}\\
\mathbb{F}_1 &= \{X_{(2,0)}\} && \text{(one error)}\\
\mathbb{F}_2 &= \{X_{(0,0)}X_{(4,0)}, X_{(1,0)}X_{(3,0)}\} && \text{(two errors)}\\
\mathbb{F}_3 &= \{X_{(1,0)}X_{(2,0)}X_{(3,0)}, X_{(0,0)}X_{(2,0)}X_{(4,0)}\} && \text{(three errors)}\\
\mathbb{F}_4 &= \{X_{(0,0)}X_{(1,0)}X_{(3,0)}X_{(4,0)}\} && \text{(four errors)}\\
\mathbb{F}_5 &= \{X_{(0,0)}X_{(1,0)}X_{(2,0)}X_{(3,0)}X_{(4,0)}\}, && \text{(five errors)}
\end{aligned}$$

where we have grouped the effective noise processes by weight and chosen as representative elements the combinations that are invariant under reflection about qubit 2 (that is, the permutation $(0\,4)(1\,3)$, which is in the symmetry group of the 5-qubit code). Furthermore, for IID depolarizing noise at the physical level (i.e., $\mathcal{N}_1 = \mathcal{D}_p^{\otimes 5}$), each of the $\bar{\mathcal{N}}_{1,V}(\mathcal{F}_j)$ is a depolarizing channel and so commutes with $\mathcal{Q}$, so that we need only consider the 6 choices of $G \in \{I, X_0, \ldots, X_4\}$. Moreover, by the reflection symmetry of the 5-qubit code and the chosen combinations about qubit 2, we need only consider $G \in \{I, \bar{X}_0, \bar{X}_1, \bar{X}_2\}$, where we use a $\bar{X}$ because the recovery map acts on a single code block as a logical X. That is, interpreting (F, G) as specifying the recovery map $R = F\mathcal{V}^{\otimes n}[G]$, there are at most 32 degeneracy classes whose representative elements are specified by the recovery maps

$$(\mathbb{F}_0 \cup \mathbb{F}_1 \cup \mathbb{F}_2 \cup \mathbb{F}_3 \cup \mathbb{F}_4 \cup \mathbb{F}_5) \times \{I, \bar{X}_0, \bar{X}_1, \bar{X}_2\}. \tag{2.60}$$

For the no error or five error cases, we can permute G by arbitrary cyclic permutations because the effective noise for the outer encoding is IID and the outer encoding is invariant under cyclic permutations, so we need only consider $G \in \{I, \bar{X}_1\}$ and so the elements in the

two sets

$$\begin{aligned}&\mathbb{F}_0 \times \{\bar{X}_0, \bar{X}_1, \bar{X}_2\}\\&\mathbb{F}_5 \times \{\bar{X}_0, \bar{X}_1, \bar{X}_2\}\end{aligned} \tag{2.61}$$

are all degenerate, reducing to single representative elements,

$$\begin{aligned}&\mathbb{F}_0 \times \bar{X}_0\\&\mathbb{F}_5 \times \bar{X}_0,\end{aligned} \tag{2.62}$$

and leaving only 28 degeneracy classes. We note that there is a further degeneracy, as explicit calculations show that there are in fact only 20 degeneracy classes as any two choices of G that are related by a permutation that leaves $\otimes_j F_j$ invariant are degenerate. We denote degeneracy by $(F, G) \cong (F', G')$. From explicit calculations, we know that

$$\begin{aligned}(X_{(2,0)}, \bar{X}_0) &\cong (X_{(2,0)}, \bar{X}_1)\\(X_{(0,0)}X_{(4,0)}, I) &\cong (X_{(1,0)}X_{(3,0)}, I)\\(X_{(0,0)}X_{(4,0)}, \bar{X}_1) &\cong (X_{(0,0)}X_{(4,0)}, \bar{X}_2)\\(X_{(1,0)}X_{(3,0)}, \bar{X}_0) &\cong (X_{(1,0)}X_{(3,0)}, \bar{X}_2)\\(X_{(1,0)}X_{(2,0)}X_{(3,0)}, I) &\cong (X_{(0,0)}X_{(2,0)}X_{(4,0)}, I)\\(X_{(0,0)}X_{(2,0)}X_{(4,0)}, \bar{X}_0) &\cong (X_{(0,0)}X_{(2,0)}X_{(4,0)}, \bar{X}_2)\\(X_{(1,0)}X_{(2,0)}X_{(3,0)}, \bar{X}_1) &\cong (X_{(1,0)}X_{(2,0)}X_{(3,0)}, \bar{X}_2)\\(X_{(0,0)}X_{(1,0)}X_{(3,0)}X_{(4,0)}, \bar{X}_0) &\cong (X_{(0,0)}X_{(1,0)}X_{(3,0)}X_{(4,0)}, \bar{X}_1).\end{aligned} \tag{2.63}$$

However, we have not been able to identify an explicit symmetry to show that these cases are degenerate.

2.6 Conclusion

In this chapter, we introduced notation which facilitates the exploration of the effective noise arising from a QECC, including the effects of noisy measurements. In section 2.5.1 we demonstrated that noisy readout measurements leave the effective logical noise invariant up to a renormalization for stabilizer codes. We showed how the computational cost of calculating the effective logical noise in a QECC can be reduced by orders of magnitude by identifying measurement outcomes and recovery maps which result in the same (or logically

degenerate) noise in section 2.4. This reduction in computational complexity does not reduce the accuracy of the simulation. In section 2.5, we demonstrated the usefulness of the reduction by presenting degeneracies for the 3-, 5-, and 7-qubit codes (tables 2.1 and 2.2) as well as concatenated codes (section 2.5.5) and the toric code (table 2.4) for independent and identically distributed noise. Identifying additional symmetries in these and other quantum error correcting codes is an open question. We anticipate that our results can be used to construct better soft decoders for concatenated codes, since logically degenerate recovery maps should simply be altered to make them degenerate.

Furthermore, a significant barrier to the successful implementation of a quantum error correcting protocol in a large system is the time that the error correction step takes. The methods in this chapter can be used to simplify the decoding step, since optimal recovery maps for a small number of nondegenerate syndromes can be pre-computed and cached. This allows other syndromes to be straightforwardly reduced to the nondegenerate syndromes and the recovery maps can be altered accordingly. While we focused on the toric code due to its translational symmetry, we expect that similar reductions will also hold for surface codes [45, 54] and color codes [55, 56] even without translational symmetry, in part because these codes have more logical gates with simple (e.g., tensor product) structures.

Chapter 3

Randomized compiling for subsystem measurements

In the previous chapter, we made the standard assumption about the physical noise and showed that it produces syndrome measurement noise of the commonly assumed form. In this chapter, we propose a noise tailoring technique for measurement noise that enforces a similar form for physical measurements.

The results presented in this chapter were first presented in [3].

3.1 Chapter overview

Measurements are a vital part of any quantum computation, whether as a final step to retrieve results, as an intermediate step to inform subsequent operations, or as part of the computation itself (as in measurement-based quantum computing). However, measurements, like any aspect of a quantum system, are highly error-prone and difficult to model. In this chapter, we introduce a new technique based on randomized compiling to transform errors in measurements into a simple form that removes particularly harmful effects and is also easy to analyze. In particular, we show that our technique reduces generic errors in a computational basis measurement to act like a confusion matrix, i.e. to report the incorrect outcome with some probability, and as a stochastic channel that is independent of the measurement outcome on any unmeasured qudits in the system. We further explore the impact of errors on indirect measurements and demonstrate that a simple and realistic noise model can cause errors that are harmful and difficult to model. Applying our technique in conjunction with

randomized compiling to an indirect measurement undergoing this noise results in an effective noise which is easy to model and mitigate.

3.2 Introduction

Measurements are the sole method of retrieving information from quantum systems; as such, they are integral to any quantum computation. In circuit-based models of quantum computation, most measurements are destructive[1], terminal measurements in the computational basis, used at the end of a circuit to retrieve results. Noise on these measurements is typically modeled by assuming that the measurement outcomes are permuted according to a *confusion matrix*, that is, a matrix containing probabilities of observing a given outcome given that another outcome occurred (recall section 1.7.2). When noise can accurately be modeled by a confusion matrix, it can be mitigated in post-processing with an overhead that, under reasonable assumptions, scales polynomially with the number of qubits [57].

However, measurement outcomes can also be used to choose subsequent operations as in quantum error correction [19] and measurement-based quantum computation [58]. For these applications, the assumption that measurements are terminal and destructive is insufficient, and the noise model that applies a confusion matrix is not well motivated. The effect of realistic noise when the measured state persists after measurement can be harmful and difficult to model. This same problem of realistic errors being difficult to model also arises for unitary channels. For such channels, the effects of noise can be reduced to a stochastic model which is easier to predict and mitigate or correct by applying randomized compiling (RC) [18].

In this chapter, we generalize RC to tailor the effects of measurement errors to a stochastic form that is easier to model. Like RC for unitary channels, our generalization does not introduce significant overhead in number of shots or number of gates. We show that our protocol reduces fully general noise on a destructive computational basis measurement to a channel which acts stochastically on the unmeasured qudits in a way that is independent of the measurement outcome, and permutes the outcomes observed on the measured qudits.

[1] In theoretical treatments of quantum computing, destructive measurements are effectively just measurements for which the measured qudits are not re-used. In experimental implementations, a distinction might be made based on the platform as measurements in e.g. photonic quantum computation destroy the measured qudits, whereas superconducting qudits persist after measurement; however this distinction is not relevant here.

We further show how harmful noise can occur from a simple and realistic perturbation of an indirect measurement and that our protocol reduces this noise to a more desirable form.

We begin in section 3.3 by defining the concept of unnormalized stochastic channels to cover the form of noise returned by our technique. In section 3.4 we give an explicit example where a simple and well-motivated noise model on a standard implementation of an indirect computational basis measurement results in noise that does not follow the standard assumptions on measurement noise, and which is harmful and difficult to model when subsequent operations are applied. In section 3.5 we show how randomized compiling can be applied to non-destructive measurements to tailor harmful errors into a form which is easier to model and less harmful. Section 3.6 generalizes section 3.4 to general indirect measurements, showing how randomized compiling can be applied to indirect measurements to tailor noise effectively, and we return to the example from section 3.4 to show the effects of randomized compiling explicitly. We finally include in section 3.6.2 an explicit derivation of a noisy implementation of a general indirect measurement of a Weyl operation which shows that the harmful noise which is removed by randomized compiling is realistic for noise on higher dimensional systems and for a common subclass of measurements.

3.3 Subsystem measurements

Stochastic channels are easier to analyze because the worst-case error rate is approximately equal to the average error rate, and are accurate as a model because general errors can be converted into stochastic channels using randomized compiling [18]. Although ref. [23] defined stochastic channels to be trace-preserving maps, the proof that the twirled channel

$$\mathbb{E}_{G\in\mathbb{G}}\mathcal{G}^\dagger\Lambda\mathcal{G} \tag{3.1}$$

is a stochastic channel for any channel Λ and unitary 1-design $\mathbb{G}$ [59] in [23, Theorem 3] can be trivially generalized to completely positive maps, where the twirled channel admits the same Kraus operator decomposition without the trace-preserving condition. We refer to such channels as unnormalized stochastic channels. Similarly, measurements which undergo a probabilistic bit-flip noise are easy to predict and model.

A case of interest is a type of measurement that we refer to as *subsystem measurements*, wherein a subset of m qudits are non-destructively measured in the computational basis. Formally, let $\mathbb{H}_{d^n} \equiv \mathbb{H}_{\text{rest}} \otimes \mathbb{H}_{\text{meas}}$ where $\mathbb{H}_{\text{meas}} \equiv \mathbb{H}_{d^m}$ is the space of states for the m qudits being measured. Then a subsystem measurement with side-evolution U applied to

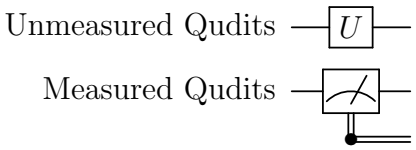


Figure 3.1: A circuit diagram showing a subsystem measurement, which consists of a computational basis measurement of a subset of m qudits of dimension d and some evolution U on the rest of the system. The corresponding channel is given in eq. (3.3).

the unmeasured qudits corresponds to measuring the m qudits and returning the outcome $k \in \mathbb{Z}_d^m$ with probability

$$\operatorname{Tr}(\pi_k \rho), \tag{3.2}$$

where $\pi_k = U \otimes |k\rangle\langle k|$, and we allow $U \in \mathbb{H}_{d^{n-m}}$ to not be the identity to allow other computation to take place during the measurement.

We can return the outcome k by "creating" a new m-qudit system that always has a definite state. This gives us the Kraus operators $\pi_k \otimes |k\rangle$ and so we obtain the overall channel

$$\mathcal{M}_{U,\mathbb{H}_{\text{meas}}} = \mathcal{U} \otimes \sum_{k \in \mathbb{Z}_d^m} |k\rangle\langle k| \otimes |k\rangle, \tag{3.3}$$

where the second tensor factor corresponds to the returned outcome. Note that this channel does not violate the no-cloning theorem because it is only cloning orthogonal states.

A subsystem measurement includes the case where $m = n$, that is, where the entire system is measured. While there are in principle more general types of measurements, in practice more complex measurements are typically implemented by applying an entangling unitary and then performing a subsystem measurement. We will return to this in more detail in section 3.6.

We now define uniform stochastic instruments, which correspond to a specific type of imperfect subsystem measurement.

Definition 8. *A uniform stochastic instrument is a channel that can be written as*

$$\sum_{a,b,k \in \mathbb{Z}_d^m} \mathcal{U}\mathcal{T}_{a,b} \otimes |k+b\rangle\langle k+a| \otimes |k\rangle \tag{3.4}$$

where $\mathcal{T}_{a,b}$ is an unnormalized stochastic channel that acts only on $\mathbb{H}_{\text{rest}}$.

A uniform stochastic instrument is a particularly simple error model in that there are only three error modes:

1. A state may be misreported, that is, the system is in the state $|k+a)$ but is reported as $|k)$;

2. The measured systems are left in the wrong computational basis state, that is, they are left in the state $|k+b)$ when the outcome k is reported; and

3. A stochastic error is applied to the unmeasured systems that is independent of the observed outcome.

We refer to such channels as "uniform" because the error applied to the rest of the system is uniform over the observed outcome. Moreover, the measured and unmeasured qudits are disentangled by such a measurement even if they were entangled before the measurement. We can see this by noting that the post-measurement state is a linear combination of tensor products of arbitrary states of the unmeasured qudits with pure states of the measured qudits. The states of the measured and unmeasured qudits after the measurement are typically classically correlated, unless only one $\mathcal{T}_{a,b}$ is nonzero in eq. (3.4). Note that the probability of observing the outcome k and leaving the system in the state $|k+b)$ given an initial state ρ is

$$\sum_{a \in \mathbb{Z}_d^m} \mathrm{Tr}\left(\mathcal{T}_{a,b}\right) \otimes (k+a|[\rho], \tag{3.5}$$

and so is determined by the unnormalized stochastic channels $\{\mathcal{T}_{a,b}\}$. Generally, some outcome will always occur and so

$$\sum_{a,b \in \mathbb{Z}_d^m} \mathcal{T}_{a,b} \tag{3.6}$$

is a trace-preserving map.

3.4 Example: indirect computational basis measurement

We now illustrate the complexity of modelling errors in subsystem measurements without randomized compiling by considering an indirect subsystem measurement, wherein the measurement is performed by coupling the system be measured to another quantum system that

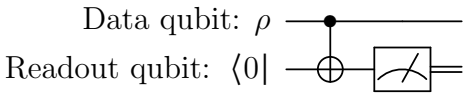


Figure 3.2: Circuit diagram of an indirect computational basis measurement. A data qubit in an initial state ρ is coupled to a readout qubit initialized in the state $\langle 0|$ by a CNOT gate. The readout qubit is then measured in the computational basis.

can be directly measured. For simplicity, we will restrict to an indirect computational basis measurement of a single data qubit by coupling it to another readout qubit, and we include as a reminder the circuit diagram for such a measurement in fig. 3.2. Typically, V would correspond to performing a controlled-X (CNOT) gate, which can be written as

$$V = |0\rangle\langle 0| \otimes I + e^{i\pi/2} |1\rangle\langle 1| \otimes e^{-i\pi X/2}. \tag{3.7}$$

The Kraus operators describing the action of an ideal implementation of fig. 3.2 on the data qubit are

$$K_j = (I_2 \otimes |j\rangle\langle j|)V(I_2 \otimes |0\rangle) = |j\rangle\langle j| \otimes |j\rangle\,, \tag{3.8}$$

for $j = 0, 1$, which corresponds to an ideal computational basis measurement of the data qubit as desired, where the second tensor factor is the observed measurement outcome.

Now suppose that the CNOT gate is implemented with an over-rotation error, which we will take to be

$$V_\phi = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes e^{-i\phi X}, \tag{3.9}$$

where $\phi = \pi/2$ corresponds to the ideal case and we ignore the $e^{i\pi/2}$ factor as it only appears as an overall phase on K_1. This results in the Kraus operators

$$\begin{aligned} \tilde{K}_0 &= |0\rangle\langle 0| \otimes |0\rangle + \cos\phi\, |1\rangle\langle 1| \otimes |0\rangle \\ \tilde{K}_1 &= -i \sin\phi\, |1\rangle\langle 1| \otimes |1\rangle\,. \end{aligned} \tag{3.10}$$

While this is an extremely simple error model representing just one likely error in an implementation, it already complicates the behavior. Consider an initial state $\rho = |+\rangle\langle +|$ for the data qubit. The ideal unnormalized output after observing the outcome $|0\rangle$ is $\frac{1}{2} |0\rangle\langle 0|$, where the trace of the unnormalized output gives the probability with which it occurs. However, with the Kraus operators in eq. (3.10) we instead obtain

$$\tilde{K}_0 |+\rangle\langle +| \tilde{K}_0^\dagger = \tfrac{1}{2} \left(|0\rangle\langle 0| + \cos\phi\, |0\rangle\langle 1| + \cos\phi\, |1\rangle\langle 0| + \cos^2\phi\, |1\rangle\langle 1| \right) \otimes |0\rangle\langle 0|\,. \tag{3.11}$$

This incorrect state cannot be modelled by a confusion matrix, as it is not a mixture of $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$. We will return to this example in section 3.6.1 to show how randomized compiling results in a simpler error model.

3.5 Randomized compiling for non-destructive quantum measurements

In this section, we introduce randomized compiling for non-destructive quantum measurements and analyze the impact on a general noisy measurement. A general noisy implementation of a subsystem measurement $\mathcal{M}_{U,\mathbb{H}_{\text{meas}}}$ as illustrated in fig. 3.1 can be expressed as

$$\Theta(\mathcal{M}_{U,\mathbb{H}_{\text{meas}}}) = \sum_{k\in\mathbb{Z}_d^n} \mathcal{M}_k \otimes |k), \tag{3.12}$$

where the $\mathcal{M}_k$ are completely positive maps from $\mathbb{H}$ to itself and the $|k)$ corresponds to the outcome. That is, a known classical outcome is returned but the corresponding evolution of the system can be arbitrary provided that the state of the system conditioned on an outcome is a valid quantum state. As some outcome is always observed, $\sum_k \mathcal{M}_k$ is a trace-preserving map, however, we do not need to formally impose this restriction.

Our goal is to transform a noisy implementation of a subsystem measurement of the form in eq. (3.12) to be a uniform stochastic measurement as in eq. (3.4) by adding gates drawn uniformly at random from the following three sets as illustrated in fig. 3.3:

1. diagonal gates on the measured qudits before and after the measurement to diagonalize the measurement;
2. basis permutations on the measured qudits to permute the expected outcome and undo the permutation after the measurement; and
3. a unitary 1-design on the unmeasured qudits before and after the measurement to reduce the errors on the unmeasured qudits to stochastic errors.

For clarity, we will first prove the well-known fact that the average effect of applying uniformly random diagonal gates as introduced before and after the measurement produces a dephasing channel before analyzing the effect of all three steps together. Throughout this chapter, we will assume that the implementations of the random gates are ideal.

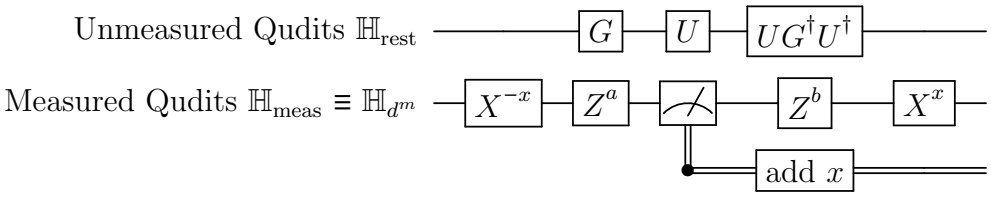


Figure 3.3: A circuit diagram showing a randomly compiled computational implementation of fig. 3.1, where $a, b, x \in \mathbb{Z}_d^m$ are m-digit strings with entries modulo d and G is an element of a unitary 1-design. In theorem 12, we prove that the uniform average over a, b, x, and G is a uniform stochastic instrument.

This assumption can be trivially generalized to errors where the implementation of each random gate U can be written as $\mathcal{LUR}$ for some constant maps $\mathcal{L}$ and $\mathcal{R}$ by redefining $\Theta(\mathcal{M}_{U,\mathbb{H}_{\text{meas}}}) \to \mathcal{R}\Theta(\mathcal{M}_{U,\mathbb{H}_{\text{meas}}})\mathcal{L}$ and similarly for other imperfectly implemented operations [60], which would correspond to the "hard" cycles in randomized compiling [18]. To both minimize the number of additional physical gates that are introduced and to justify the above redefinition of the noisy implementations of measurements and other hard cycles, adjacent random gates should be compiled into each other and into neighbouring single-qudit gates from the bare circuit where possible. As such, when measurements are randomly compiled in addition to other hard cycles following ref. [18], the number of physical gates to be implemented does not change. Moreover, in the limit where random gates are chosen independently for each execution of a circuit, no additional executions are required [61].

Lemma 11. *For any positive integers d and n with $d \geq 2$,*

$$\mathbb{E}_{a\in\mathbb{Z}_d^n}\left(\mathcal{Z}^a\right) = \sum_{z\in\mathbb{Z}_d^n} |Z^z\rangle\!\rangle\langle\!\langle Z^z| = \sum_{j\in\mathbb{Z}_d^n} |j\rangle\!\rangle\langle\!\langle j| \,.$$

Proof. Using eq. (1.13) with $\mathbb{PW}_{d,n}$ as the basis, we have

$$\begin{aligned}\mathcal{Z}^a &= \sum_{x,z\in\mathbb{Z}_d^m} |Z^a X^x Z^z Z^{-a}\rangle\!\rangle\langle\!\langle X^x Z^z| \\ &= \sum_{x,z\in\mathbb{Z}_d^m} \chi_a(x) |X^x Z^z\rangle\!\rangle\langle\!\langle X^x Z^z|, \end{aligned} \tag{3.13}$$

where the second equality follows from eq. (1.28). Averaging over a and using eq. (1.32)

gives

$$\begin{aligned}\mathbb{E}_{a\in\mathbb{Z}_d^n}\left(\mathcal{Z}^a\right) &= \sum_{x,z\in\mathbb{Z}_d^n} |X^x Z^z\rangle\!\rangle\langle\!\langle X^x Z^z| \mathbb{E}_{a\in\mathbb{Z}_d^m}\left(\chi_a(x)\right) \\ &= \sum_{x,z\in\mathbb{Z}_d^n} |X^x Z^z\rangle\!\rangle\langle\!\langle X^x Z^z| \mathbb{E}_{a\in\mathbb{Z}_d^n}\left(\chi_x(a)\chi_0^*(a)\right) \\ &= \sum_{z\in\mathbb{Z}_d^n} |Z^z\rangle\!\rangle\langle\!\langle Z^z|, \end{aligned} \tag{3.14}$$

establishing the first equality.

We can use eq. (1.32) to invert eq. (1.25) and obtain

$$|j\rangle\langle j| = \mathbb{E}_{z\in\mathbb{Z}_d^n}\left(\chi_j^*(z) Z^z\right). \tag{3.15}$$

Substituting eq. (3.15) into eq. (3.14) and using linearity gives

$$\begin{aligned}\mathbb{E}_{a\in\mathbb{Z}_d^n}\left(\mathcal{Z}^a\right) &= \sum_{j,k\in\mathbb{Z}_d^m} |j\rangle\!\rangle\langle\!\langle k| \, \mathbb{E}_{a\in\mathbb{Z}_d^m}\left(\chi_j(a)\chi_k^*(a)\right) \\ &= \sum_{j\in\mathbb{Z}_d^m} |j\rangle\!\rangle\langle\!\langle j|, \end{aligned} \tag{3.16}$$

establishing the second equality. □

Having analyzed the effect of the random diagonal gates, we now prove that averaging over all the random gates in fig. 3.3 produces a uniform stochastic instrument.

Theorem 12. *If the implementation of a noisy measurement as in fig. 3.1 is described by the channel eq. (3.12) and the additional gates in fig. 3.3 are ideal, then the average of fig. 3.3 over $a, b, x \in \mathbb{Z}_d^m$ and G over a unitary 1-design $\mathbb{G}$ is a uniform stochastic instrument.*

Proof. Under the stated assumptions, we can write the channel implemented by fig. 3.3 for fixed $a, b, x \in \mathbb{Z}_d^m$ and $G \in \mathbb{G}$ as

$$\Theta_{a,b,x,G}(\mathcal{M}_{U,\mathbb{H}_{\text{meas}}}) = \sum_{k\in\mathbb{Z}_d^m} \Lambda_{k,a,b,x,g} \otimes |k\rangle\!\rangle \tag{3.17}$$

where we have relabeled the classical outcomes to account for adding x so that the measurement acts as $\mathcal{M}_{k-x} \otimes |k\rangle\!\rangle$ and we define

$$\Lambda_{k,a,b,x,G} = (\mathcal{U}\mathcal{G}^\dagger\mathcal{U}^\dagger \otimes \mathcal{X}^x\mathcal{Z}^b)\mathcal{M}_{k-x}(\mathcal{G} \otimes \mathcal{Z}^a\mathcal{X}^{-x}). \tag{3.18}$$

The average over $a, b, x \in \mathbb{Z}_d^m$ and $G \in \mathbb{G}$ is then

$$\Theta_{\mathrm{RC}}(\mathcal{M}_{U,\mathbb{H}_{\mathrm{meas}}}) = \mathbb{E}_{a,b,x\in\mathbb{Z}_d^m,G\in\mathbb{G}}\left(\Theta_{a,b,x,G}(\mathcal{M}_{U,\mathbb{H}_{\mathrm{meas}}})\right). \tag{3.19}$$

We will evaluate this average by first averaging over the diagonal gates, that is, over $a, b \in \mathbb{Z}_d^m$, then over the permutation gates, that is, over $x \in \mathbb{Z}_d^m$, and finally over the unitary 1-design, that is, over $G \in \mathbb{G}$. By lemma 11, the independent uniform average over $a, b \in \mathbb{Z}_d^m$ gives

$$\mathbb{E}_{a,b}\left(\Lambda_{k,a,b,x,G}\right) = \sum_{\alpha,\beta\in\mathbb{Z}_d^m} (\mathcal{U}\mathcal{G}^\dagger\mathcal{U}^\dagger \otimes \mathcal{X}^x\,|\beta)(\beta|)\mathcal{M}_{k-x}(\mathcal{G}\otimes|\alpha)(\alpha|\,\mathcal{X}^{-x}). \tag{3.20}$$

Now note that the $\mathcal{X}$ channels permute the ideal projectors, since

$$X^x\,|j\rangle\langle j|\,X^{-x} = |j+x\rangle\langle j+x|\,. \tag{3.21}$$

Therefore for any fixed x we have $\mathcal{X}^x|j) = |j+x)$ and so averaging $\Lambda_{k,a,b,x,G}$ over $a, b, x \in \mathbb{Z}_d^m$ for fixed values of k and G gives

$$\begin{aligned}\mathbb{E}_{a,b,x\in\mathbb{Z}_d^m}\left(\Lambda_{k,a,b,x,G}\right) &= \sum_{\alpha,\beta\in\mathbb{Z}_d^m} \mathbb{E}_{x\in\mathbb{Z}_d^m}\left((\mathcal{U}\mathcal{G}^\dagger\mathcal{U}^\dagger\otimes|\beta+x)(\beta|)\mathcal{M}_{k-x}(\mathcal{G}\otimes|\alpha)(\alpha+x|)\right)\\ &= \sum_{\alpha,\beta\in\mathbb{Z}_d^m} \mathcal{U}\mathcal{G}^\dagger\mathcal{M}_{k,\alpha,\beta}\mathcal{G}\otimes|k+\beta)(k+\alpha|\,,\end{aligned} \tag{3.22}$$

where we relabel the terms in the sum by setting $\alpha \to \alpha + k - x$ and $\beta \to \beta + k - x$ and define

$$\mathcal{M}_{k,\alpha,\beta} = \mathbb{E}_{x\in\mathbb{Z}_d^m}\left((\mathcal{U}^\dagger\otimes(\beta+k-x|)\mathcal{M}_{k-x}(\mathcal{I}_{\mathrm{rest}}\otimes|\alpha+k-x))\right), \tag{3.23}$$

which only acts on $\mathbb{H}_{\mathrm{rest}}$. Therefore to complete the proof we need to show that $\mathbb{E}_G\mathcal{G}^\dagger\mathcal{M}_{k,a,b}\mathcal{G}$ is an unnormalized stochastic channel that is independent of k. To see that it is independent of k, note that if we relabel $x \to k + x$ in eq. (3.23), then the right-hand side is manifestly independent of k. As $\mathcal{M}_{k,\alpha,\beta}$ only acts on $\mathbb{H}_{\mathrm{rest}}$, we set

$$\mathcal{T}_{\alpha,\beta} = \mathbb{E}_{G\in\mathbb{G}}\left(\mathcal{G}^\dagger\mathcal{M}_{k,\alpha,\beta}\mathcal{G}\right) = \mathbb{E}_{G\in\mathbb{G}}\left(\mathcal{G}^\dagger\mathcal{M}_{0,\alpha,\beta}\mathcal{G}\right), \tag{3.24}$$

which is an unnormalized stochastic channel [23], completing the proof. □

3.6 General indirect measurements

We now analyze how randomly compiling subsystem measurements can be used to perform effective indirect measurements with simple error models. Indirect measurements are one of

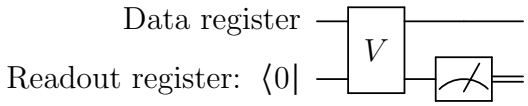


Figure 3.4: Circuit diagram of a general indirect measurement, which can be used to perform indirect measurements such as syndrome measurements.

the fundamental techniques in quantum error correction as they are used to detect if the system is in the code space without disturbing the state of the system if it is in the code space and to determine the correction to apply if it is not in the code space. Such measurements are performed by a circuit as in fig. 3.4, which generalizes fig. 3.2. The corresponding Kraus operators are

$$K_k = (I \otimes \langle k|)V(I \otimes |0\rangle) \otimes |k\rangle\,, \tag{3.25}$$

where the $|k\rangle$ represents the observed outcome. Typically, we choose

$$V = \sum_{j,k \in \mathbb{Z}_d^m} \pi_k \otimes |k+j\rangle\langle j| \tag{3.26}$$

for a projective measurement $\{\pi_k : k \in \mathbb{Z}_d^m\}$. To see that V is unitary, note that for any projective measurement we have $\sum_k \pi_k = I$ and so

$$V^\dagger V = \sum_{j,k \in \mathbb{Z}_d^m} \pi_k \otimes |j\rangle\langle j| = I \otimes \sum_{j \in \mathbb{Z}_d^m} |j\rangle\langle j| = I, \tag{3.27}$$

and similarly for $VV^\dagger$. Thus the ideal Kraus operators are

$$K_k = \pi_k \otimes |k\rangle\,, \tag{3.28}$$

which is equivalent to performing the ideal projective measurement on the state register and encoding the result in the state of the readout register. As in section 3.4, simple error models will result in Kraus operators that are not proportional to the ideal projectors and thus leave coherences, arising from cross-terms between the orthogonal subspaces corresponding to different ideal outcomes. For example, in section 3.6.2, we show that an indirect measurement projecting onto the eigenspaces of a Weyl operator with a simple error leaves such coherences. These coherences are assumed to not exist in many analyses of quantum error correction and can cause particularly harmful accumulation of errors as different cospaces and combinations thereof are allowed to interfere in subsequent operations.

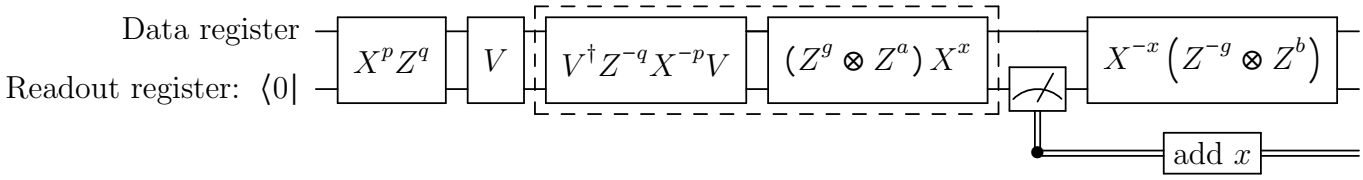


Figure 3.5: Randomly compiled implementation of a stabilizer measurement where V is a multi-qudit Clifford operation. As V is a multi-qudit Clifford operation, $V^\dagger Z^{-q}X^{-p}V$ is a tensor product of single-qudit Weyl operators. The random Weyl operators are specified by $p, q, x \in \mathbb{Z}_d^n$, $g \in \mathbb{Z}_d^{n-m}$, and $a, b \in \mathbb{Z}_d^m$.

While fig. 3.4 is valid for any unitary V, we typically require some structure to ensure that the random gates that are added are simpler than the gate being twirled, and thus do not significantly add to the complexity (and typically the total error) of the circuit. We now consider the special case where V is a Clifford unitary, that is, when $VWV^\dagger$ is proportional to a Weyl operator for any $W \in \mathbb{PW}_{d,n}$. In fig. 3.5, we show how to implement fig. 3.4 with randomized compiling so that Markovian implementations of a Clifford unitary operation and the measurement are reduced to the ideal Clifford channel followed by a uniform stochastic measurement. Considering first the Clifford operation V, from ref. [18], averaging $p, q \in \mathbb{Z}_d^n$ gives the effective channel

$$\Theta_{\mathrm{RC}}(\mathcal{V}) = \Lambda\mathcal{V}, \tag{3.29}$$

where Λ is a stochastic Weyl channel, that is, a channel of the form [62]

$$\Lambda = \sum_{x,z \in \mathbb{Z}_d^n} \mu(x,z)\mathcal{X}^x\mathcal{Z}^z \tag{3.30}$$

where $\mu : \mathbb{Z}_d^{2n} \to [0,1]$ is a probability distribution. Combining this with theorem 12, when we average over all the random gates in fig. 3.5, we obtain the channel

$$\begin{aligned}\Theta_{\mathrm{RC}}(\mathcal{M}_{U,\mathbb{H}_{\mathrm{meas}}}\mathcal{V}) &= \Theta_{\mathrm{RC}}(\mathcal{M}_{U,\mathbb{H}_{\mathrm{meas}}})\Theta_{\mathrm{RC}}(\mathcal{V}) \\ &= \left(\sum_{a,b,k \in \mathbb{Z}_d^m} \mathcal{T}_{a,b} \otimes |k+b\rangle\langle k+a| \otimes |k\rangle\right)\Lambda\mathcal{V}\end{aligned} \tag{3.31}$$

where the $\mathcal{T}_{a,b}$ are unnormalized stochastic channels. Moreover, as the $\mathcal{T}_{a,b}$ are twirled by the Weyl group, they are also of the form

$$\mathcal{T}_{a,b} = \sum_{\alpha,\beta \in \mathbb{Z}_d^{n-m}} \nu_{a,b}(\alpha,\beta)\mathcal{X}^\alpha\mathcal{Z}^\beta \tag{3.32}$$

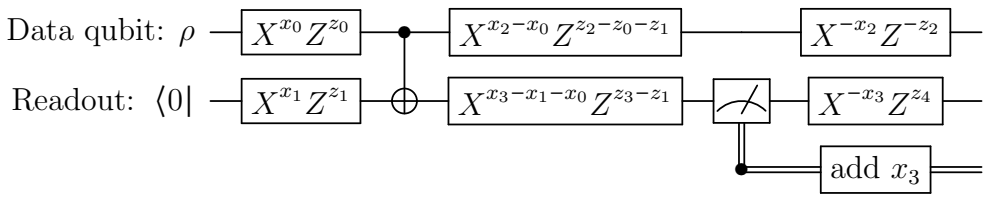
where each $\nu_{a,b} : \mathbb{Z}_d^{n-m} \to [0,1]$ is a subnormalized probability distribution that together add to a normalized probability distribution. Substituting eqs. (3.30) and (3.32) into eq. (3.31) and using the fact that the product of Weyl channels is a Weyl channel gives

$$\Theta_{\mathrm{RC}}(\mathcal{M}_{U,\mathbb{H}_{\mathrm{meas}}}\mathcal{V}) = \sum_{a,b,k\in\mathbb{Z}_d^m} \big(\Lambda_{a,b} \otimes |k+b\rangle\langle k+a|\big)\,\mathcal{V} \otimes |k\rangle, \tag{3.33}$$

for some subnormalized stochastic channels $\Lambda_{a,b}$. When V is a Clifford operation that is also of the special form in eq. (3.26), eq. (3.33) simplifies to performing the ideal projective measurement and then applying a uniform stochastic channel. While we focused on measurements in the computational basis throughout this chapter, these results can be generalized to measurements in another basis by re-defining the random Z and X gates relative to the measurement basis.

3.6.1 Two-qubit example

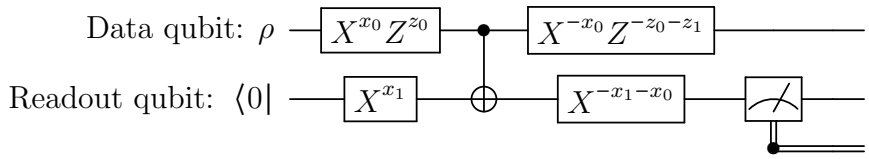
For concreteness, we return to the simple two-qubit example in section 3.4 and show how it is transformed by randomized compiling. Using the fact that Z is diagonal and using the identities $CX(X \otimes I)CZ = X \otimes X$ and $CX(I \otimes Z)CX = Z \otimes Z$ to determine the correction gates, so that the randomly compiled implementation of section 3.4 is as follows, where the x_j and z_j are bits.



(3.34)

As we will only consider a noisy implementation of the CNOT gate, we will simplify the above by compiling adjacent single-qubit operations, however, this compilation would result in a different noise process if the single-qubit operations are implemented imperfectly. We simplify eq. (3.34) by applying the first Z^{z_1} to the input state and canceling the random gates where possible by commuting them through single-qubit circuit elements, noting that $\mathcal{X}$ and $\mathcal{Z}$ commute as channels. Because the computational basis measurement is ideal, the random X and Z gates applied before and after it leave it unchanged. These gates should not generally be omitted when the measurement itself is noisy, but in our example the noise is coming solely from a multi-qubit gate, so they have no effect. These simplifications give

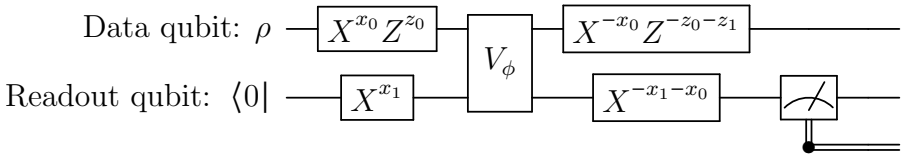
the following circuit.



(3.35)

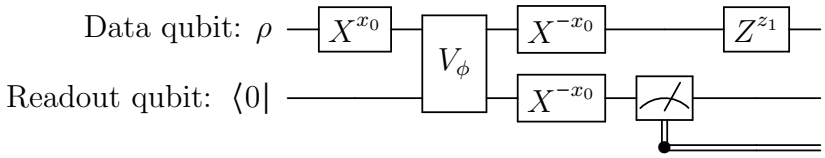
While eq. (3.35) is equivalent to simply randomly compiling the CNOT, randomized compiling has not been analyzed in the context of non-destructive measurements. Theorem 12 establishes that the complete indirect measurement with general errors will be reduced to a uniform stochastic measurement as in eq. (3.33), however, the only simple error models for measurements are confusion matrices, which do not fully capture the dynamics of a non-destructive measurement.

We therefore restrict attention to the well-motivated example of an over-rotation gate during an indirect measurement and show that RC helps for this explicit example. When the CNOT in eq. (3.35) is implemented imperfectly using the V_ϕ gate defined in eq. (3.9), the overall channel is equivalent to the following.



(3.36)

As V_ϕ commutes with $Z \otimes I$ and $I \otimes X$, we obtain the reduced circuit



(3.37)

Setting aside the Z^{z_1} gate (which will average to a dephasing channel by lemma 11) to begin, we have four Kraus operators arising from two measurement outcomes and two values of x_0 These Kraus operators are given by

$$L_{k,x_0} = \begin{cases} (|x_0\rangle\langle x_0| + \cos\phi\,|1+x_0\rangle\langle 1+x_0|) \otimes |k\rangle & k = x_0 \\ -i\sin\phi\,|1+x_0\rangle\langle 1+x_0| \otimes |k\rangle & k \neq x_0. \end{cases} \tag{3.38}$$

Averaging over $z_1 \in \mathbb{Z}_2$ removes any off-diagonal terms introduced by conjugating by these Kraus operators, so that effectively we have 6 Kraus operators for the final channel. We can combine like terms to reduce this down to 4 Kraus operators of the form

$$\tilde{K}_{k',k} = \sqrt{|p(k,k')|}\,|k'\rangle\langle k'| \otimes |k\rangle\,, \tag{3.39}$$

where k' indicates which ideal projector acts on the state space, k is the result written to the readout register, and $p(k, k')$ is a coefficient representing the probability of observing k and applying k'. After this simplification, the final set of Kraus operators can be written

$$\begin{aligned}
\tilde{K}_{0,0} &= (1 - i \sin\phi) \, |0\rangle\langle 0| \otimes |0\rangle \\
\tilde{K}_{0,1} &= \cos\phi \, |1\rangle\langle 1| \otimes |0\rangle \\
\tilde{K}_{1,0} &= \cos\phi \, |0\rangle\langle 0| \otimes |0\rangle \\
\tilde{K}_{1,1} &= (1 - i \sin\phi) \, |0\rangle\langle 0| \otimes |0\rangle \, .
\end{aligned} \tag{3.40}$$

This noise can thus be condensed into a confusion matrix representation

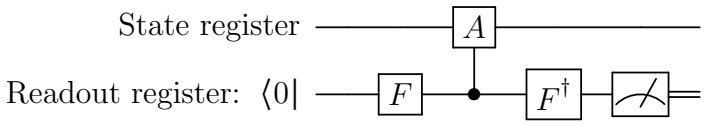
$$\frac{1}{2} \begin{pmatrix} 1 + \sin^2\phi & \cos^2\phi \\ \cos^2\phi & 1 + \sin^2\phi \end{pmatrix}. \tag{3.41}$$

Adding additional errors to the CZ gate may change the confusion matrix and/or add a stochastic Pauli error channel acting on the state space. In section 3.6.2, we show how this example generalizes to higher dimensions and also indirect measurements of multi-qudit operators.

3.6.2 Indirect measurements with noisy Clifford operations

Noise can arise in any or all of the component operations used to implement an indirect measurement. In this subsection, we look specifically at the effect of an over- or under-rotation of the coupling operation used to implement an indirect Weyl measurement.

Recall the explicit form for a circuit which performs an indirect measurement of a Weyl operator $A \in \mathbb{PW}_{d,n}$ given in fig. 1.6,



We assume that the Fourier and inverse Fourier operations are performed perfectly and implement the controlled-A operation as an imperfect rotation. Recall that the ideal controlled-A operation is

$$\hat{A} = \sum_{j \in \mathbb{Z}_d} |j\rangle\langle j| \otimes A^j. \tag{3.42}$$

We now define an operation which can represent an over- or under-rotation of the applied operation,

$$\hat{A}(t) \equiv \hat{A}^t = \sum_{j \in \mathbb{Z}_d} |j\rangle\langle j| \otimes A^{jt}, \tag{3.43}$$

where the ideal operation is applied when $t = 1$, that is $\hat{A}(1) = \hat{A}$. We will assume that t is close to 1 and let $t = 1 + \epsilon$. Then

$$A^{jt} = A^{j(1+\epsilon)} = A^j (A^j)^\epsilon. \tag{3.44}$$

For small ϵ, we can expand B^ϵ as

$$B^\epsilon = I + \epsilon \ln(\mathrm{B}) + \mathcal{O}(\epsilon^2). \tag{3.45}$$

Applying this expansion to eq. (3.44),

$$A^{jt} = A^j [I + \epsilon \ln(\mathrm{A^j}) + \mathcal{O}(\epsilon^2)] \tag{3.46}$$

$$\approx A^j + \epsilon j A^j \ln(\mathrm{A}). \tag{3.47}$$

To get the complete expression for the Kraus operators of an indirect measurement with noise of the described form, we fold the Fourier transform into the input state to get an initial state

$$F|0\rangle = \frac{1}{\sqrt{d}} \sum_{a \in \mathbb{Z}_d} |a\rangle \equiv |+\rangle, \tag{3.48}$$

and fold the inverse Fourier transform into the computational basis measurement, to get an effective destructive measurement on the readout space with Kraus operators

$$\mathcal{M}_{I,\mathbb{H}_r} F^\dagger = \{\langle k| F^\dagger : k \in \mathbb{Z}_d\} \tag{3.49}$$

$$= \{\frac{1}{\sqrt{d}} \sum_{b \in \mathbb{Z}_d} \chi_b^*(k) |k\rangle\langle b| : k \in \mathbb{Z}_d\} \tag{3.50}$$

acting on the state register. Combining the input state (eq. (3.48)), interaction operation, $\hat{A}^t$, and effective measurement (eq. (3.50)), we get Kraus operators for the noisy measurement of the form

$$M_k(t) = \frac{1}{d} \sum_{j \in \mathbb{Z}_d} \chi_k^*(j) A^{jt}. \tag{3.51}$$

Note that for the remainder of this section we omit the readout register, treating this as a destructive measurement, for ease of book-keeping. However, the following analysis generalizes trivially to a non-destructive measurement. Substituting eq. (3.47) into the Kraus operators in eq. (3.51), we get

$$\begin{aligned} M_k(1+\epsilon) &\approx \frac{1}{d}\sum_{j\in\mathbb{Z}_d}\chi_k^*(j)[A^j + \epsilon j A^j \ln(\mathrm{A})] \\ &\approx \pi_k(A) + \frac{\epsilon \ln(\mathrm{A})}{d}\sum_{j\in\mathbb{Z}_d}\chi_k^*(j) j A^j, \end{aligned} \tag{3.52}$$

where $\pi_k(A) = M_k(1)$ are the projectors on the state space for the ideal measurement of A. Ideally, a measurement of A with outcome k projects fully onto the k^{th} eigenspace of A so that for a set of eigenvectors $\{|\psi_b\rangle : b \in \mathbb{Z}\}$ of A, $\langle\psi_b|\,\pi_k(A)\,|\psi_b\rangle = \delta_{b,k}\langle\psi_b|\,\psi_b\rangle$. If the second term is not proportional to $\pi_k(A)$, this measurement can leave some of the state outside of the k^{th} eigenspace of A. To see this, let

$$J_k = \sum_{j\in\mathbb{Z}_d}\chi_k^*(j) j A^j \ln(\mathrm{A}), \tag{3.53}$$

so that we can describe the second term more succinctly, and observe that the terms resulting from applying $M_k(t)$ to a state $\bar{\rho}$ will be proportional to:

1. $\pi_k(A)\bar{\rho}\pi_k(A)^\dagger$ (which is in the expected eigenspace)
2. cross terms $J_k\bar{\rho}\pi_k(A)^\dagger$ and $\pi_k(A)\bar{\rho}J_k^\dagger$, and
3. $J_k\bar{\rho}J_k^\dagger$.

If J_k is proportional to a projector (or a combination of projectors) other than $\pi_k(A)$, the cross terms allow superpositions between the cospaces to persist beyond the measurement, and $J_k\bar{\rho}J_k^\dagger$ allows terms outside of the k^{th} eigenspace to persist as well.

We now show that J_k does not implement a complete projection onto the k^{th} eigenspace, i.e. J_k is not proportional to $\pi_k(A)$. We can define an eigenbasis of A with eigenvalues $\{\chi_b(1) : b \in \mathbb{Z}_d\}$ by selecting a $+1$ eigenvector $|\psi_0\rangle$ of A for $b = 0$ and letting the remaining eigenvectors be $|\psi_b\rangle = T_b\,|\psi_0\rangle$, where $AT_b = \chi_b(1)T_bA$. To show that J_k does not project onto the k^{th} eigenspace, we need to demonstrate that $J_k\,|\psi_b\rangle$ is non-zero for some $b \neq k$. If $|\psi_b\rangle$ is an eigenvector of A with eigenvalue $\chi_b(1)$, then $|\psi_b\rangle$ is also an eigenvector of $\ln(\mathrm{A})$

with eigenvalue $\ln(\chi_b(1))$. So, recalling eq. (1.30), we have

$$J_k |\psi_b\rangle = \sum_{j\in\mathbb{Z}_d} \chi_k^*(j) j \ln[\chi_b(1)] \chi_b(1)^j |\psi_b\rangle \tag{3.54}$$

$$= \frac{2\pi i b}{d} \sum_{j\in\mathbb{Z}_d} \chi_k^*(j)\chi_b(j) j |\psi_b\rangle . \tag{3.55}$$

Using the series identity

$$\sum_{i=0}^{n} k x^k = x\frac{1-(n+1)x^n + n x^{n+1}}{(1-x)^2} \tag{3.56}$$

and eq. (1.30), it can be shown that

$$\frac{2\pi i}{d} \sum_{j\in\mathbb{Z}_d} \chi_1(j) j = \pi\left[\cot\left(\frac{\pi}{d}\right) - i\right]. \tag{3.57}$$

Therefore for $k = 0$ and $b = 1$, we have

$$J_0 |\psi_1\rangle = \pi\left[\cot\left(\frac{\pi}{d}\right) - i\right] |\psi_1\rangle , \tag{3.58}$$

which is always nonzero. As 0 and 1 are in $\mathbb{Z}_d$ for any $d > 2$, there will always be an outcome which can leave a portion of the state outside of the expected eigenstate after measurement for any meaningful qudit system. We can therefore conclude that an over- or under- rotation of the coupling gate for an indirect Weyl measurement can leave a portion of the state outside of the eigenspace associated with the measured outcome. Note that the asymmetry between, e.g., outcomes $k = 0$ and $k = 1$ arises as a result of the asymmetry in the error model; if the state of the system prior to the measurement is a $+1$ eigenvector of A, then $\hat{A}(t)$ leaves it unchanged for any value of t.

3.7 Conclusion

Measurements are a fundamental part of quantum computations, and are a major source of intractable noise in modern quantum systems. In this chapter, we showed that the common method for modeling noisy measurements is inadequate to capture the dynamics in a realistic system. We proposed a compilation method which addresses the intractability of measurement errors, rendering them easier to mitigate and model, and showed that applying our technique reduces measurement errors to a form which is commonly used to model measurement errors, that is, to probabilistic misreporting of measurement outcomes and a

stochastic channel on the unmeasured qudits. Our method applies to non-destructive and indirect measurements, making it fully generalizable to any case where measurements could be used in a quantum computation, and does not introduce significant overhead in either the number or gates or number of shots required.

Chapter 4

Logical Randomized Compiling

In the previous chapter we showed how to tailor measurement noise at the physical level to follow the typically assumed model. In this chapter, we extend this result to tailor noise on any gadget in an encoded implementation to follow the most commonly assumed models.

The results presented in this chapter were first presented in [4], and a patent has been filed for the proposed protocol.

4.1 Chapter overview

Studies of quantum error correction (QEC) typically focus on stochastic Pauli errors because the existence of a threshold error rate below which stochastic Pauli errors can be corrected implies that there exists a threshold below which generic errors can be corrected. However, rigorous estimates of the threshold for generic errors are typically orders of magnitude worse than the threshold for stochastic Pauli errors. Specifically, coherent errors have a particularly harmful impact on the encoded space because they can map encoded states to superpositions of logical and error states. Further, coherent errors can add up and interfere over multiple rounds of error correction or between syndrome measurements, which may result in significantly worse errors than expected under a stochastic Pauli error model. In this chapter, we present an algorithm which decoheres noise at the logical level, projecting the state of the system onto a logical state with a well-defined error. The algorithm does not significantly increase the depth of the logical circuit (and usually does not lead to any increase in depth), and applies generally to most fault-tolerant gadgets and error correction steps.

4.2 Introduction

Studies of quantum error correcting codes often assume a noise model which is relatively easy to model and correct effectively. In most cases, this is a probabilistic Pauli noise model on the encoded space, with ideal syndrome extraction [32, 44, 63], however, recently there have been many numerical studies of relatively simple coherent noise models, which result in a wide range of effective error rates [21, 30, 40]. In cases where noisy syndrome extraction is considered, it typically follows a model where syndromes are probabilistically misreported because this model is straightforward to simulate [21, 30, 32, 40, 44, 63, 64].

Quantum systems can be used for more efficient computations because they can be in coherent superpositions of states. When written as an expansion in terms of outer products of the superposed states, these coherences are represented by cross-terms between the states, for example $|0\rangle\langle 1|$ or $|1\rangle\langle 0|$ have coherence between $|0\rangle$ and $|1\rangle$. Unintended coherences can cause noise that is difficult to model and detrimental to the computation as they can interfere and build up quickly. Ideal measurements remove coherences (that is, the cross-terms) between eigenstates of the measured operator. For example, if we begin in a superposed state of $|0\rangle$ and $|1\rangle$, e.g.

$$|+\rangle\langle +| = \frac{1}{2}\left(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + |1\rangle\langle 1|\right) \tag{4.1}$$

and measure in the computational basis (i.e. measure the Z operator), we remove the cross terms $|0\rangle\langle 1|$ and $|1\rangle\langle 0|$. Similarly, averaging over powers of Z removes coherences between $|0\rangle$ and $|1\rangle$ (the eigenspaces of Z) [3].

Recall that in quantum error correction, a state is encoded in a subspace of a larger physical space, referred to as the codespace, and, for stabilizer codes, there are a set of subspaces isomorphic to the codespace which we refer to as cospaces. Ideal syndrome measurement allows us to project onto individual cospaces and distinguish which cospace the state is in after measurement, and we apply a correction based on this information. We refer to coherence between states in different spaces as coherences between those spaces. When non-trivial coherences arise in a stabilizer code, they take the form of coherences within cospaces and coherences between cospaces. There are three types of error which deviate from the standard assumptions about noise in QECCs that we are concerned about:

1. Measurement errors other than classical bit flips;

2. Coherence within a cospace;

3. Coherence between cospaces.

Much like with the example given above for coherences between $|0\rangle$ and $|1\rangle$, ideal syndrome measurement removes coherences between cospaces. However, syndrome measurement may not be ideal, and errors can build up and interfere during the syndrome measurement process. We therefore need to address all three of these deviations rather than letting syndrome measurement handle coherence between cospaces. In this chapter, we propose a technique called Logical Randomized Compiling (LRC), in which we address each of these problems as follows: we address measurement errors by applying the compilation method proposed in ref. [3] to physical measurements, and a generalization thereof to logical measurements; we address coherences within cospaces by applying a generalization of RC [18] to a logical (usually Pauli) twirling group; and we address coherences between cospaces by applying random stabilizers between operations (a more detailed breakdown of how each type of gadget is compiled is given in section 4.4). We show that averaging over the application of random powers of stabilizers removes coherences between cospaces (which are eigenspaces of the stabilizers), analogous to how averaging over the application of random powers of Z operations decoheres coherences between eigenspaces of Z.

Previous work has looked at the impact of RC [18] on the performance of QECs [46, 65]. In the absence of twirling, residual coherence can be more efficiently corrected using non-Pauli logical operations [46]. This possibility can still be used with LRC if the logical twirl is omitted. RC has been shown to improve the logical fidelity for a wide class of errors, however, it can sometimes decrease the logical fidelity by removing contributions from the physical noise that interfere in a helpful way [65]. Moreover, it is hard to predict whether the coherent contributions will interfere constructively or destructively. Another study, ref. [66], proposed inserting random stabilizers to mimic syndrome measurement (without getting a returned syndrome) and demonstrated that it can improve the logical error rate in the absence of error correction, but did not fully analyze the technique. Notably, as shown in theorem 13, the method proposed in ref. [66] removes coherences between cospaces but does not address coherences within cospaces. As in ref. [66], we propose inserting stabilizers throughout a computation to enforce a noise model which is more predictable and less harmful. However, we propose also using syndrome extraction and error correction to correct the remaining error. We include a mathematical analysis of the effect of applying random stabilizers, and combine this randomization with generalizations of other noise tailoring techniques [3, 18].

The chapter is structured as follows. We begin by illustrating the potential for coherent errors that cannot be removed by existing techniques in section 4.3. We then define LRC

in section 4.4 and give a qualitative explanation for how it removes undesired coherences. We next discuss some details around noise that will facilitate our proposal section 4.5. We then prove that LRC removes undesired coherences in section 4.6. We illustrate how our technique is applied to syndrome extraction circuits in section 4.7, and conclude with some implementation details and open questions in section 4.8.

4.3 Example: Coherent errors in the repetition code

In this section, we provide an example where LRC is helpful and RC cannot be applied. Specifically, we look at a noisy gadget implementation of a Toffoli gate for the 3-qubit repetition code, and show that a well-motivated noise model can produce an error which creates coherences between cospaces that can be suppressed by LRC. RC is not applicable in this setting because the Toffoli gate is a non-Clifford gate.

Recall the 3-qubit repetition code (reviewed in section 1.8.1), which has the codewords $|\bar{0}\rangle = |000\rangle$ and $|\bar{1}\rangle = |111\rangle$. This code can be described by the stabilizer generators ZZI and IZZ and the logical Pauli operations $\bar{I} = III$, $\bar{X} = XXX$, $\bar{Y} = -YYY$, and $\bar{Z} = ZZZ$. We denote a rotation about a Pauli P by an angle θ by

$$P_\theta = \exp(-i\theta P) = \cos\theta I - i \sin\theta P. \tag{4.2}$$

Conjugating a state by P_θ results in cross-terms between the initial state and the state that P maps to. In stabilizer codes, stabilizers act trivially, logical Pauli operators map within a cospace, and error operators map between cospaces. Therefore, coherent logical Pauli operations (that is, errors of the form $L_\theta = \exp(-i\theta L)$ for a logical Pauli operator $L \in \mathbb{L}$) create coherences within a cospace, and coherent errors create coherences between cospaces.

Recall that the Toffoli gate is the three-qubit gate

$$\hat{\hat{X}} = \sum_{i,j \in \mathbb{Z}_2} |ij\rangle\langle ij| \otimes X^{ij}. \tag{4.3}$$

The corresponding ideal logical gate is

$$\overline{\hat{\hat{X}}} = \sum_{i,j \in \mathbb{Z}_2} |\bar{i}\bar{j}\rangle\langle \bar{i}\bar{j}| \otimes \bar{X}^{ij}. \tag{4.4}$$

In the 3-qubit repetition code, the logical Toffoli gate can be implemented by a gate gadget composed of transversal $\hat{\hat{X}}$ gates, i.e.,

$$\overline{\hat{\hat{X}}} = \hat{\hat{X}}_{(1,1),(2,1),(3,1)} \hat{\hat{X}}_{(1,2),(2,2),(3,2)} \hat{\hat{X}}_{(1,3),(2,3),(3,3)}, \tag{4.5}$$

and the subscripted indices label which qubits the Toffoli gates act on, with a tuple (a, b) indicating the b^{th} qubit in the a^{th} code block.

We will look at an example where the conditional X in one of the physical Toffoli gates in the Toffoli gadget is overrotated. To model this overrotation, note that $X \cong X_{\pi/2}$ up to a phase. Then we can express an overrotation of an X gate by an angle δ as $X_{\pi/2+\delta} = \exp(-i(\pi/2+\delta)X) = X \cdot X_\delta$. A Toffoli gate overrotated by an angle δ is then given by

$$\hat{\hat{X}}(\delta) = \sum_{i,j \in \mathbb{Z}_2} |ij\rangle\langle ij| \otimes (X \cdot X_\delta)^{ij}. \tag{4.6}$$

We will let the first Toffoli gate in the gadget defined in eq. (4.5) be the one which undergoes an overrotation and observe the effect when this noisy gadget acts on the $|\bar{1}\bar{1}\bar{1}\rangle$ state, that is, the case where all three code blocks are in the logical 1 state:

$$\begin{aligned}
\hat{\hat{X}}(\delta)_{(1,1),(2,1),(3,1)} \hat{\hat{X}}_{(1,2),(2,2),(3,2)} \hat{\hat{X}}_{(1,3),(2,3),(3,3)} |\bar{1}\bar{1}\bar{1}\rangle &= |\bar{1}\bar{1}\rangle \otimes (X \cdot X_\delta) \otimes X \otimes X |\bar{1}\rangle && (4.7)\\
&= |\bar{1}\bar{1}\rangle \otimes (XII)_\delta \bar{X} |\bar{1}\rangle && (4.8)\\
&= |\bar{1}\bar{1}\rangle \otimes (XII)_\delta |\bar{0}\rangle && (4.9)
\end{aligned}$$

In the ideal implementation, i.e. $\delta = 0$, the final state is $|\bar{1}\bar{1}\bar{0}\rangle$. The error introduced by the overrotation is then $(XII)_\delta$ on the third code block. For the three qubit code, XII is an error and so $(XII)_\delta$ maps the system to a superposition of states in different cospaces. By applying random stabilizers, LRC removes the coherence between the cospaces, ensuring that the state is in the syndrome space. Note that due to the symmetry of the code and the gadget implementation, changing which Toffoli gate is overrotated results in different coherent errors.

4.4 Introduction to Logical Randomized Compiling

In this section, we will give an intuitive overview of the motivation, implementation, and impact of LRC on a fault-tolerant quantum computation. The following sections will provide a more rigorous mathematical treatment and justification for the protocol.

Recall that any quantum computation can be written as a sequence of three types of primitive operations:

1. *state resets*;
2. *unitary operations*; and

3. *measurements.*

And that an encoded implementation is composed of gadget implementations of the logical versions of these primitives plus an additional gadget to implement syndrome measurement. Because gadgets are the building blocks of an error-corrected (possibly fault-tolerant) implementation of a circuit, we are concerned with compiling gadgets to tailor noise into a form that is easy to handle, i.e., with no unwanted coherences and only bit flip measurement errors.

Our protocol, LRC, reduces the effect of generic Markovian errors to the form that is typically considered in QEC, making them easier to analyze and allowing standard decoders designed for stochastic Pauli errors to be used effectively. Specifically, each type of operation is compiled to suppress deviations from the desired form of noise as follows:

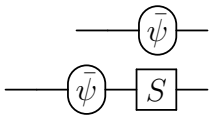
1. To account for measurement noise, measurements are randomly compiled [3] using logical operations.

2. To account for coherence within a cospace, logical twirling operations are applied before and after logical unitary operations.

3. To account for coherence between cospaces, random stabilizers are applied before and after every operation.

More concretely, the basic building blocks of fault tolerant quantum computing are modified as follows, where stabilizers and logical operations are always selected from the current code if code switching is being used [67].

1. *State resets*
   A random stabilizer S is applied after each encoded state reset. This ensures that the prepared state does not have coherences between cospaces.

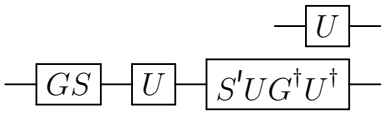


(Bare State Reset Gadget)

(LRC State Reset Gadget)

2. *Unitary operations*
   A random stabilizer S followed by a random logical operation G drawn uniformly from a group $\mathbb{G}_U$ is applied before each encoded unitary operation U. The compiled operation $UG^\dagger U^\dagger$ followed by a second random stabilizer S' are applied after the encoded unitary

operation. The use of random stabilizers ensures that the state before and after the logical operation does not have coherences between cospaces. The application of G and $UG^\dagger U^\dagger$ performs a twirl as in RC [18] which (with judicious selection of $\mathbb{G}_U$) removes coherent noise within the cospaces. We discuss the selection of $\mathbb{G}_U$ in section 4.6. For Clifford gates, we will typically select the logical Pauli group (or the logical Weyl group for qudits).

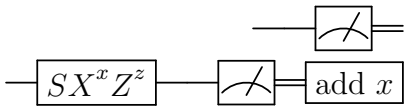


(Bare Gate Gadget)

(LRC Gate Gadget)

3. *Measurements*

   Before each encoded measurement, that is each measurement of an encoded logical Weyl operation, a random stabilizer S and random logical operation L are performed. The classical output of the measurement is adjusted according to the selected random logical operation. The random stabilizer ensures that the state does not contain coherences between cospaces prior to measurement. The inclusion of the other operations is analogous to randomized compiling for instruments [3].

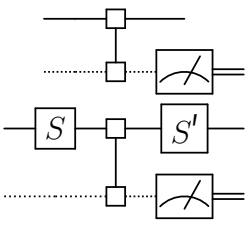


(Bare Measurement)

(LRC Measurement)

4. *Syndrome extraction*

   A syndrome extraction measures stabilizer generators of the code by applying operations on a combined space created by the union of some readout qubits and the qudits used in the encoding and then measuring the readout qudits. Before and after each syndrome measurement or part thereof (i.e. before and after measurement of a single dit of the syndromes), we apply a random stabilizer on the encoded space. We have omitted some details here for ease of readability. Specifically, the syndrome extraction circuit should be randomly compiled, and the physical measurement should undergo randomized compiling for measurements in addition to the randomization over powers of stabilizers. We will give a more complete specification in section 4.7.



(Bare Syndrome Extraction Gadget)

(LRC Syndrome Extraction Gadget)

Where possible, the additional operations introduced by LRC are compiled into adjacent operations. As the additional operations are typically transversal single-qubit operations, very few additional physical operations will be introduced by LRC. These modifications apply at the logical level, however protocols for mitigating or suppressing noise at the physical level can still be used for some parts of the implementation. In particular, dynamical decoupling or refocusing techniques and physical RC could be applied to the circuit implementations of encoded operations (see e.g. refs. [68, 69]).

4.5 Gauge degree of freedom

There exists a gauge degree of freedom in how noise is expressed on operations whereby an identity operation can be inserted between consecutive operations via any invertible matrix M as $M^{-1}M$ to alter the form of the noise. Under such a transformation, a simple circuit composed of a (possibly noisy) state preparation of a state ρ, a (possibly noisy) unitary operation U, and a (possibly noisy) measurement $\mathbb{M}$ is transformed as follows

$$\Theta(\rho)\Theta(U)\Theta(\mathbb{M}) \to \Theta(\rho)M_0^{-1}M_0\Theta(U)M_1^{-1}M_1\Theta(\mathbb{M}), \tag{4.10}$$

so that we get some new implementation map Θ' such that

$$\Theta'(\rho) = \Theta(\rho)M_0^{-1} \tag{4.11}$$

$$\Theta'(U) = M_0\Theta(U)M_1^{-1}, \text{ and} \tag{4.12}$$

$$\Theta'(\mathbb{M}) = M_1\Theta(\mathbb{M}). \tag{4.13}$$

As with any gauge transformation, the different implementation maps produce equivalent results, that is

$$\Theta(\rho)\Theta(U)\Theta(\mathbb{M}) = \Theta'(\rho)\Theta'(U)\Theta'(\mathbb{M}). \tag{4.14}$$

The matrices used to change the gauge of noises before and after each operation may be non-identical, e.g. we can have $M_0 \neq M_1$, and may, for example, depend upon the start time of an instruction. Noisy unitaries can be factored as

$$\Theta(U) = \Delta_L\mathcal{U}\Delta_R, \tag{4.15}$$

where Δ_L and Δ_R are noise maps [60]. Some time-dependent noise processes can be removed by an appropriate gauge transformation by folding the gauge which follows the unitary

operation into the noise map preceding the next operation, thereby allowing us to assume that a noisy implementation of a unitary operation and the corresponding gadget can be written as

$$\Theta(U) = \mathcal{U}\Delta_U, \text{ or} \tag{4.16}$$

$$\Gamma(U) = \Gamma_{\text{id}}(U)\bar{\Delta}_U \tag{4.17}$$

for some noise maps Δ_U or $\bar{\Delta}_U$.

4.6 Analysis of LRC

In this section, we show how averaging over the application of random stabilizers before and after an operation results in a probabilistic map between cospaces and that twirling over the logical Weyl group enforces a probabilistic noise map within cospaces. We then combine these results and measurement RC to show that the effective noise from the resultant protocol follows the probabilistic model we want to achieve when a logical Weyl twirl can be used, and a noise model which has fewer undesired coherences when such a twirl is not possible. We give some examples of other twirling groups to replace the twirl over logical Weyls where needful, and note that where there is not a known twirling group for the specific operation being tailored, the application of random stabilizers is still useful to remove coherences between cospaces and can be applied regardless of the form that the surrounding operations take.

We begin with a proof that, for stabilizer codes, applying uniformly random stabilizers projects the state of the system onto the syndrome space. This projection removes coherences between cospaces of the code. Note that if the state begins and ends on the syndrome space, the only permitted noise is within cospaces; the effective map if we apply this averaging to project onto the syndrome space before and after an operation is a probabilistic mapping between cospaces plus possible noise within each cospace. Because the proof is given in terms of channels and does not depend on the input state, the averaging is independent of the surrounding operations under the assumption that the stabilizer averaging occurs in the same location in the encoded circuit for each randomization.

Theorem 13. *For a code with stabilizer group* $\mathbb{S}$*, we have*

$$\mathbb{E}_{S\in\mathbb{S}}(\mathcal{S}) = \sum_{T\in\mathbb{T}} \Pi_T. \tag{4.18}$$

Proof. In the natural representation, reviewed in section 1.4.1, we have

$$\mathcal{M} = M \otimes M^*. \tag{4.19}$$

Substituting in eq. (1.70), we have

$$\begin{aligned}\sum_{T\in\mathbb{T}} \Pi_T &= \sum_{T\in\mathbb{T}} \pi_T \otimes \pi_T^* \\ &= \mathbb{E}_{S,S'\in\mathbb{S}} \left(S' \otimes S^* \sum_{T\in\mathbb{T}} \chi_T^*(S)\chi_T(S') \right)\end{aligned}$$

Recall that as $\mathbb{S}$ is an Abelian group and χ_T is a character of $\mathbb{S}$ and thus obeys the character orthogonality relations eq. (1.32)

$$\sum_{T\in\mathbb{T}} \chi_T^*(S)\chi_T(S') = \delta_{S,S'}|\mathbb{S}|. \tag{4.20}$$

Therefore, we have

$$\begin{aligned}\sum_{T\in\mathbb{T}} \Pi_T &= \mathbb{E}_{S\in\mathbb{S}} \left(S \otimes S^* \right) \\ &= \mathbb{E}_{S\in\mathbb{S}}(\mathcal{S}),\end{aligned}$$

as claimed, where the factor of $|\mathbb{S}|$ cancels with its inverse in the average over S'. □

By theorem 13, averaging over applications of stabilizers is equivalent to projecting onto the syndrome space, which removes coherences between cospaces. Now that we have a method that handles coherence between cospaces, we next look at a method to remove or reduce coherences within cospaces. Specifically, we look at the effect of twirling an operation over a group up to a correction that preserves the operation's ideal action. We show that when this method is applied to a noisy implementation of an operation, it can be used to address coherences within a cospace. Recall that a standard twirl (without correction to preserve the desired action) has the following definition:

Definition 9. *Given a group $\mathbb{G}$, a channel Λ, we define the twirl of Λ over $\mathbb{G}$ as*

$$\Lambda^{\circlearrowleft\mathbb{G}} \equiv \mathbb{E}_{G\in\mathbb{G}} \left(\mathcal{G}^\dagger \Lambda \mathcal{G} \right). \tag{4.21}$$

We now show that applying a corrected twirl to a noisy implementation of a unitary operation results in an effective map comprised of the ideal unitary operation followed by a

twirled version of the noise. This is useful for the prevention of coherences within cospaces because twirled operations do not induce coherences between the states that would be returned by applications of different elements of the twirling group. We can then remove coherences within cospaces by selecting a twirling group whose elements act within a cospace. Note that twirling only over the logical Weyl group does not address coherences between cospaces because the representation of the logical Weyl group on the physical space is not multiplicity free. That is, states outside of the syndrome space can persist despite a twirl over the logical Weyl group. The following is a straightforward generalization of randomized compiling [18] without assuming any structure for the twirling group.

Theorem 14. *Let $\mathbb{G}_U$ be a group, U be a unitary operator. Averaging over specific ideal operations applied before and after a noisy encoded unitary gadget $\Gamma(U)$ produces an effective channel comprised of the ideal action of U followed by the noise introduced by $\Gamma(U)$ twirled over $\mathbb{G}_U$. These operations take the form*

$$G \qquad \text{(Before)}$$
$$UG^\dagger U^\dagger \qquad \text{(After)}$$

for $G \in \mathbb{G}$.

Proof.

$$\mathbb{E}_{G\in\mathbb{G}}\left(\mathcal{U}\mathcal{G}^\dagger\mathcal{U}^\dagger\Gamma(U)\mathcal{G}\right) = \mathcal{U}\mathbb{E}_{G\in\mathbb{G}}\left(\mathcal{G}^\dagger\Delta_U\mathcal{G}\right) \tag{4.22}$$
$$= \mathcal{U}\Delta_U^{\circlearrowleft\mathbb{G}}, \tag{4.23}$$

where $\Delta_U \equiv \mathcal{U}^\dagger\Gamma(U)$ can be interpreted as the noise introduced by $\Gamma(U)$, for which a noisy unitary gadget for U would have an implementation map $\Gamma(U) = \mathcal{U}\Delta_U$, consistent with eq. (4.17). □

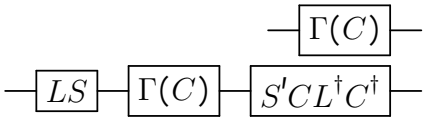
In practice, the additional operations G and $UG^\dagger U^\dagger$ are compiled into surrounding operations. As such, we can assume that the additional operations are ideal under suitable assumptions of gate-independent noise [18]. Under gate-dependent noise, we can perturb from the average and apply theorem 14 provided the perturbations are small.

We now explore different options for the logical twirling group to use when applying theorem 14 to logical unitary operations. Note that for the purposes of the following discussion we are omitting the random stabilizers from consideration, however, we recommend applying both techniques in combination. A compiled version of a unitary operation U of

the form $(UG^\dagger U^\dagger)UG$ will always be equivalent to the bare unitary U, assuming that G is unitary. We can therefore in theory select any twirling group composed of unitary operations to implement the twirling component of LRC, based on what noise tailoring we wish to achieve. In practice, however, ease of implementation matters; we generally want to select twirling operations for which the implementation of G and $UG^\dagger U^\dagger$ is composed of single-qudit operations where possible so that twirling operations can be more easily compiled into neighbouring operations and introduce minimal noise where such compilation is not possible. The suggested twirling groups, $\mathbb{G}_U$, given in this section assume that Weyl operations are "easy", as is U.

4.6.1 Clifford operations

Recall that, by definition, Clifford operations preserve the projective Weyl group, that is for any Clifford operation C, $\mathcal{C}[L] \in \mathbb{PW}_{d,n}\ \forall L \in \mathbb{PW}_{d,n}$. For any unitary operation C in the Clifford group, we select $\mathbb{G}_C = \mathbb{L}$. Then the LRC implementation of C takes the form



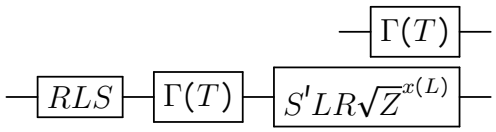
(Bare Clifford Gadget)

(LRC Clifford Gadget)

for gates $L \in \mathbb{L}$. Twirling over $\mathbb{L}$ results in a stochastic Weyl channel, so the noise introduced during the compiled Clifford is stochastic Weyl.

4.6.2 T gates

There are some known instances where non-Clifford gadgets can be compiled to further tailor the noise arising from the imperfect gadget implementation. The T gate is one such gate with a known twirl. For $d = 2$, we can apply operations before and after the T gadget, defined as $T = \exp(-i\pi Z/8)$, to reduce the effective noise from this gadget to a stochastic Pauli channel, with the further restriction that $p_X = p_Y$. This compilation is possible using the dihedral group [70], i.e. setting $\mathbb{G}_U = \langle \mathbb{PW}_{2,n}, T^2 \rangle$ to perform the twirl. Then, for any $G \in \mathbb{G}_U$, we can factor $G = RL$ for a rotation $R \in \langle T^2 \rangle$, $L \in \mathbb{PW}_{2,n}$ and simplify the compilation gates to get



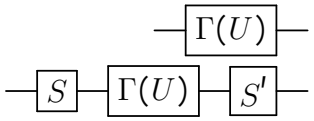
(Bare T Gadget)

(LRC T Gadget)

where $a(L)$ for $a \in \{x, z\}$ is the power of A in L such that $L = X^{x(L)}Z^{z(L)}$. For any encoding for which we can implement the Clifford group and the T gate fault-tolerantly (which together form a universal gateset), we can therefore apply LRC to get a universal gateset for which the effective noise is a stochastic Weyl channel and which remains in the union of the cospaces of the code.

4.6.3 Gates with no known twirl

For operations for which there is no known useful twirling group, we can still apply the stabilizer randomization to project back onto the cospaces, thereby reducing some of the effects of coherent errors. This amounts to setting $\mathbb{G}_U = \{I\}$ so that the compilation looks like



(Bare Gate Gadget)

(LRC Gate Gadget)

4.7 Syndrome extraction circuits

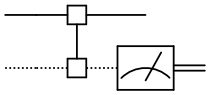
Recall that a QEC gadget is composed of a syndrome extraction step followed by a recovery operation conditioned on the extracted syndrome and that syndrome measurement is performed by coupling the encoded register to another register of qudits called the readout register to perform an indirect measurement of each of the stabilizer generators. In this section, we show how to apply LRC, focusing on syndrome extraction as it is the piece of a fault-tolerant implementation with the highest number of distinct types of operations working in tandem and so the most complicated to compile (a syndrome extraction contains unencoded measurement and state preparation as well as operations which act on an encoded and unencoded register). Ideal syndrome extraction is a common assumption in studies of QEC because the resulting circuits are simpler and can often be efficiently simulated. Specifically, for channels containing multiple rounds of error correction with imperfect

syndrome extraction (specifically when the imperfection is not constrained to probabilistically misreported syndromes), the build up and interference of terms within and outside of the syndrome space quickly becomes complicated and expensive to track.

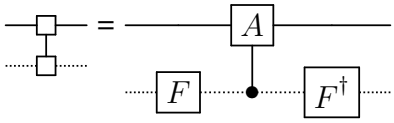
Under LRC, we apply random stabilizers before and after the syndrome extraction circuit to project the state of the encoded qudits onto the syndrome space. The projection onto the syndrome space removes any coherences between cospaces so that these coherences cannot interact and build up during the syndrome extraction circuit (theorem 13). We apply a corrected twirl, as in theorem 14, to ensure that the effective noise for coupling operations is Pauli both on the encoded register and the readout register. Finally, we apply measurement RC [3] to the measurements of the readout qudits.

We note that in practice, a refocusing sequence and stabilizer randomization would be introduced during any idling time on the encoded register while the syndrome register is being measured. We discuss this more in section 4.8.2. The refocusing sequence and random stabilizers applied while the readout register is being measured account for any noise that might accumulate on the encoded register during the measurement and ensure that the final state is in the syndrome space.

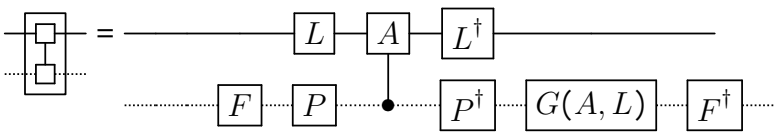
Though we can apply LRC to the full syndrome extraction, it is beneficial to apply it instead to the syndrome extraction performed for each individual syndrome dit. In this way, we prevent coherences that could be removed by applying random stabilizers from persisting and interfering during subsequent operations before the full syndrome extraction is completed. We will now step through the compilation process for a single-dit syndrome extraction, beginning with a bare syndrome extraction gadget,



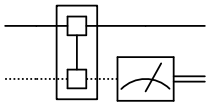
where the syndrome is extracted by writing the eigenvalue of a stabilizer generator A to the readout qudit using a controlled Weyl operation, so that



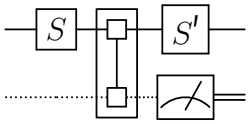
The randomly compiled controlled Weyl operation (including the basis change implemented by F gates) is then



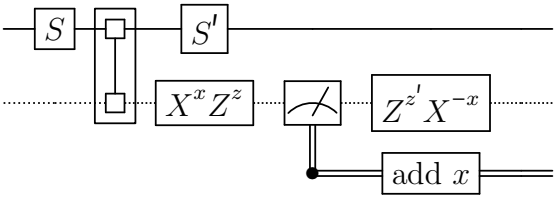
where the L operations are drawn from the logical Weyl group, $G(A, L)$ is a Weyl operator that corrects for the operation that propagates from the L acting on the encoded register to the readout register via the controlled-A operation, and P is an element of the Weyl group on the physical register. Applying this random compilation to the syndrome extraction circuit, we get



Next, we add random stabilizers before and after the syndrome extraction circuit to get

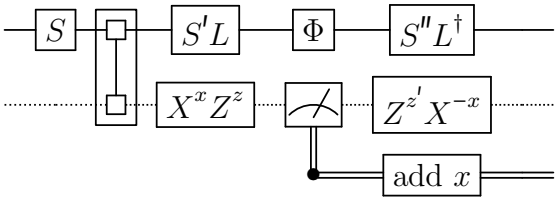


This ensures that the state before and after the noisy measurement is in the syndrome space (theorem 13). Finally, we add measurement RC to get



This ensures that the measurement decouples the readout space from the encoded space and that measurement noise takes the form of probabilistically misreported syndromes.

Optionally, we can add operations to account for noise which accumulates on the encoded register during measurement (as measurement takes longer than unitary operations and the encoded register can undergo noise while idling). We let these operations be denoted Φ and include a discussion about what form they might take in section 4.8.2. The simplest option would be to repeatedly apply random stabilizers. Including this adjustment, the fully compiled syndrome extraction is



(LRC Syndrome Extraction Gadget)

We add a twirl of Φ over the logical Weyl operators to ensure that the noise introduced on the encoded register during idling is a stochastic Weyl channel. Because Φ is an implementation

of the identity channel, the twirled version will simply be a stochastic Weyl map. Further, since the entire syndrome extraction is composed of Clifford operations, we can commute all of the stochastic Weyl noise terms that remain after compiling each element to one end of the circuit (e.g. the end) and combine them into a single stochastic Weyl error channel with bit flip errors on the on the readout register and reported syndrome.

Doing a complete syndrome extraction requires that we repeat this process $n - k$ times, once for each generator. Because each repetition is a circuit that has the same form, i.e. is composed of Clifford gates and a measurement, we can similarly move all of the remaining noise processes to the end of the sequence of syndrome dit extractions. The effective noise on the encoded register after completing the syndrome extraction is then a stochastic Weyl map on the cospaces of the code, and the confusion matrix for the complete syndrome can be constructed by taking products of the elements of the confusion matrices for individual syndrome dit measurements, as in [2]. The syndrome dits may additionally be left in the incorrect state by the syndrome measurements, but this is accounted for by the reset performed on the readout registers between syndrome measurements.

4.8 Discussion

In this section, we discuss some of the implementation details that are relevant to applying LRC effectively. Specifically, we explore the considerations for compiling the operations introduced by LRC into a circuit in such a way that added noise and overhead are minimal to none in section 4.8.1, discuss some options for compiling idling registers in section 4.8.2, briefly address some considerations regarding code switching in section 4.8.3, and finally comment on the applicability of RC and LRC to single-shot algorithms in section 4.8.4.

4.8.1 Timing and Compilation

We expect that the gates introduced by LRC will generally be transversal single-qudit gates (SQGs), which introduce less noise than other operations. We also assume that each gadget begins and ends with a cycle of single-qudit gates (with appropriate modifications for state reset and measurement gadgets). Under these assumptions, we can compile the gates introduced by LRC into the start and end of each gadget, and also compile the SQGs at the end of one gadget with the SQGs at the beginning of the next gadget. LRC will therefore introduce minimal overhead in terms of gate count, and in the assumption that SQGs have gate-independent noise, will not introduce any additional noise to the system.

One concern that arises for implementing the stabilizer averaging is that each shot of a QEC circuit will be slightly different as different syndromes are measured and, accordingly, different recovery operations are applied. Because theorem 13 is proven using channels, the input state to the randomization does not impact the proof. As such, the surrounding operations do not impact the effect of randomizing over stabilizers as long as the randomization occurs at the same location in the circuit for each shot. The timing does not need to be exact in terms of wall clock time because delays are identity operations and inserting identity operations does not impact the implemented map.

An additional note is that while RC removes the coherent part of the noise much like LRC does, there are some instances where the coherent part of physical noise can improve the logical fidelity. In such cases, removing the coherence at the physical level worsens performance. In contrast, LRC can be used without a physical twirl, which will not remove the contribution of the physical coherent noise to the logical fidelity.

Moreover, LRC can be used in cases where RC would not be viable. For example, stabilizer averaging can be used even in cases where the gate being implemented does not have an easily implementable twirling group, so for a gadget implementation of a non-Clifford gate like the Toffoli gate example described for the 3-qubit code in section 4.3, some coherences can be removed by LRC, while RC could not be implemented for this use-case.

4.8.2 Idling

It is common for some subset of qudits in a computation to be idle while others undergo operations. Perhaps most notably, measurements take longer to implement than gates so if the computation cannot proceed beyond the measurement until it has been completed, some registers may be left idle while the measurement is underway. In [3], operations on unmeasured qudits are randomly compiled to make the effective noise on the unmeasured register follow a stochastic model. We propose a similar implementation, where when qudits are idle, we apply first and last a random stabilizer, and in between we apply a refocusing sequence that persists for the remaining duration of the idle time. There are many choices of refocusing techniques; one should be chosen to optimize performance based on the noise model that is present in the system and possibly on the encoding. This process should be applied anywhere that qudits would otherwise idle in a computation and not just when measurements are in progress. Because the ideal implementation of an idling sequence is the identity, twirling gates can and should be added to constrain the effective noise map to a stochastic Weyl channel.

4.8.3 Code switching

While we do not look at code switching methods explicitly in this chapter, it should be noted that computations which involve code switching can be randomly compiled by applying randomizations according to whatever the encoding is at any given step in a circuit. For example, in switching from a code C to a code C', the minimal example would be to apply random stabilizers from code C prior to switching and random stabilizers from code C' after switching. This ensures that any portion of the state outside of the codespace prior to the switch does not get mapped into the new code in a way that is disadvantageous and ensures that the initial state after switching to C' is in the syndrome space of C'. For any steps where the register is unencoded, randomized compiling should be used, and for any where the system is in a concatenated encoding, stabilizers (and logical operations) of the combined code should be used.

4.8.4 Sampling Algorithms

Some quantum computing algorithms require only sampling from a distribution rather than estimating the full distribution. In this section, we will argue that for such algorithms, LRC (and RC) can still be used.

For each compilation c of a circuit, we have an output probability distribution $\vec{p}_c$. The average over random compilations results in an average over output distributions, where the average output distribution $\vec{p}_{avg}$ approximates the ideal distribution $\vec{p}_{ideal}$, that is

$$\vec{p}_{ideal} \approx \vec{p}_{avg} = \frac{1}{|c|} \sum_c \vec{p}_c. \tag{4.24}$$

The probability of getting a ditstring b when sampling a single shot from $\vec{p}_{avg}$ is given by

$$\vec{p}_{avg}(b) = \frac{1}{|\{c\}|} \sum_c \vec{p}_c(b) \tag{4.25}$$

because the distributions $\vec{p}_c$ are mutually independent. If we take a single shot with a single compilation sampled uniformly at random, the probability of returning ditstring b is

$$\sum_c P(\vec{p}_c)\vec{p}_c(b) = \frac{1}{|\{c\}|} \sum_c \vec{p}_c(b), \tag{4.26}$$

where the equality comes from the different compilations being equiprobable (i.e. $P(\vec{p}_c) = 1/|\{c\}|$). This is the same probability as in eq. (4.25), so we can conclude that taking a single shot with a randomly sampled compilation is equivalent to taking a single shot from the average distribution which approximates the ideal distribution.

4.9 Conclusion

Noise in quantum error corrected systems often deviates from the common assumption of a stochastic Weyl model with ideal or probabilistically misreported syndrome measurements. In this chapter, we proposed a method that generalizes and combines existing compilation strategies in the context of fault-tolerant quantum computations to show how noise can be tailored to follow the desired form that is generally assumed. Under our protocol, noisy quantum error corrected implementations are easier to model and the remaining noise is easier to correct using the standard Weyl recovery methods.

Chapter 5

Conclusion

5.1 Summary of results

In this thesis, we developed results that significantly reduce the computational complexity of simulating quantum error correcting codes (QECCs), and proposed protocols for noise tailoring for measurements and in quantum error correction (QEC). These results render the average behaviour of noise in an error corrected setting and for measurements consistent with standard assumptions in both contexts. A summary of our results is included below.

In chapter 2, we developed results on the use of symmetries for efficient simulation of quantum error correcting codes. We derived conditions under which effective encoded memory channels are equivalent up to logical operations, thereby extending the work begun in refs. [1, 21] to general QECCs rather than restricting to stabilizer codes as was done in all previous work. We showed that these symmetries are still useful when syndrome measurements undergo noise which causes the physical measurement to probabilistically misreport outcomes, extended to the case where multiple QEC steps are performed, included a treatment of non-IID noise and concatenated codes, and presented examples of such symmetries for the toric code. We also showed that no symmetry operation can hold for every noise model for a given code. These results enable a significant reduction in computational complexity in simulations of QECCs and could enable more efficient error correction. For current implementations of surface codes, the time taken to implement the decoding step is a major roadblock for QEC; this is because the decoder relies on a large set of possible errors so a lookup table is intractable and other methods must be used. The symmetries we found for the toric code significantly reduce the size of the lookup table. The number of single-qubit errors that need to be considered for the toric code is reduced from $\mathcal{O}(n)$ to $\mathcal{O}(1)$, and the

number of 2-qubit errors from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$. Notably, the number of single qubit errors that needs to be considered is independent of n, which will lead to massive gains for large codes. For example, for a 10×10 code with 50 qubits (i.e. a distance 5 toric code), this amounts to a reduction in single-qubit errors from 150 to 6 and 2-qubit errors from 11,025 to 297. Such reductions could make decoders tractable for even large codes. Further, knowledge of symmetries also enables improved selection of recovery operations by suggesting recovery operations which lead to degeneracy with maps known to have a smaller error.

In chapter 3, we developed a method called *measurement randomized compiling*. Isolating the effects of noise coming from measurements is a challenging task and as a result, measurement noise is particularly difficult to model accurately. As a result, the standard assumptions about the form that measurement noise takes in theoretical treatments and when creating protocols for assessing or mitigating measurement noise are not well motivated by physical systems. In particular, measurements are often assumed to be terminal and destructive, which is insufficient for many usecases. Perhaps the most notable use case for non-destructive measurements is quantum error correction, where readout qudits are repeatedly reused for subsequent syndrome measurements. Studies of QEC which consider measurement noise typically assume a model wherein ideal projectors are applied and outcomes are probabilistically misreported; most of the protocols that have been developed that consider measurement noise rely on this assumed form. In fact, the results presented in chapter 2 on symmetries in QECCs relied on this model. Our protocol, measurement randomized compiling, enforces a model on the average measurement consistent with the typical assumptions and applies to non-destructive measurements as well as terminal measurements. That is, under measurement randomized compiling, the noise on the measurement takes the form of bit flip errors on the measured register, with a stochastic Weyl noise on the unmeasured qudits. Applying measurement RC therefore allows any protocols founded under the standard assumptions around measurement noise to be applied effectively. Importantly, the average channel under measurement RC removes entanglement between the measured register and any unmeasured register, leaving only classical correlations. Because our technique applies to non-destructive and indirect measurements, it is applicable to the QEC setting and other settings where non-destructive measurements might be used. Measurement RC does not introduce significant overhead in terms of number of shots or number of gates.

Combining the results in chapter 3 with the derived form of syndrome measurement noise in chapter 2, we saw that when the syndrome extraction circuit is ideal and the physical measurement undergoes measurement RC, the noise on syndrome measurements follows the standard assumption of ideal projections onto cospaces with probabilistically misreported

syndromes. We thus extended our treatment to encoded implementations in chapter 4 and proposed another protocol, *logical randomized compiling*, which enforces standard noise assumptions on any gadget. Specifically, LRC ensures that errors do not map out of the syndrome space, that any noise within the cospaces follows a stochastic Pauli (or Weyl) model, and that syndromes are probabilistically misreported. These constraints on the noise are standard assumptions in studies of quantum error correction, as stochastic noise is easier to model, mitigate, and correct using the standard Pauli (or Weyl) recovery methods. While some studies deviate from these assumptions by including coherent noise, these studies are limited in number and in scope. LRC can be used in any QEC implementation that uses stabilizer codes, and ensures that coherent errors are not allowed to persist and interfere. We included for completeness compilation methods for a universal gateset, a state preparation gadget, a measurement gadget, and a syndrome extraction gadget. We showed that RC and LRC can be used even in the case where a single shot is used to sample from a distribution, and we note that LRC can be used even in cases where RC would not be usable, e.g. when the physical operations used to implement a gadget are non-Clifford. LRC does not introduce significant overhead in terms of number of shots or number of gates.

5.2 Future work

Recently, a protocol to implement randomized benchmarking at the logical level (logical randomized benchmarking) was proposed [71]. However, more research showed that effects of non-Markovianity arising from coupling between readout qudits and encoded qudits can impede a useful implementation of logical randomized benchmarking [72]. One area of exploration for LRC and/or measurement RC would be to see if the effective removal of entanglement between measured and unmeasured qudits under these techniques is sufficient to allow logical randomized benchmarking to be applied effectively.

Because LRC enforces a stochastic Weyl model which is consistent with typical recovery protocols, knowing the probabilities of different errors would enable a much improved performance of a QECC. Cycle error reconstruction is an error characterization technique that assumes a stochastic Weyl model and relies on RC to enforce that assumption [12]. LRC could therefore allow us to extend cycle error reconstruction to a logical implementation and it could be used to inform recovery selection, resulting in drastically improved performance as errors are accurately assessed. Another area of exploration would be to implement measurement RC on a device and see if it enables improved performance when combined with measurement noise mitigation techniques founded on the typically assumed measurement

noise model. Future work could also include finding more symmetry operations for other codes and noise models. More efficient decoding methods could be designed for topological codes by leveraging symmetries and data compression techniques, thereby increasing decoder speed, which is currently a major limiting factor in the successful implementation of QECCs.

References

[1] Stefanie J. Beale. The Effects of Quantum Error Correction on Noisy Systems. uwspace.uwaterloo.ca, 2018.

[2] Stefanie J. Beale and Joel J. Wallman. Efficiently computing logical noise in quantum error-correcting codes. Physical Review A, 103:6, 2021.

[3] Stefanie J. Beale and Joel J. Wallman. Randomized compiling for subsystem measurements. arXiv, 2304.06599, 2023.

[4] Stefanie J. Beale and Joel J. Wallman. Randomized compiling in fault-tolerant quantum computation. arXiv, 2306.13752, 2023.

[5] Robert E. Fontana Jr. and Gary M. Decad. Moore's law realities for recording systems and memory storage components: Hdd, tape, nand, and optical. AIP Advances, 8:056506, 2018.

[6] Richard P. Feynman. Simulating physics with computers. International Journal of Theoretical Physics, 21:467-488, 1982.

[7] W. Gale, E. Guth, and G. T. Trammell. Determination of the quantum state by measurements. Phys. Rev., 165:1434-1436, 1968.

[8] J. F. Poyatos, J. I. Cirac, and P. Zoller. Complete characterization of a quantum process: The two-bit quantum gate. Phys. Rev. Lett., 78:390-393, 1997.

[9] Joseph Emerson, Robert Alicki, and Karol Życzkowski. Scalable noise estimation with random unitary operators. J. Opt. B, 7(10):S347, 2005.

[10] Alexander Erhard, Joel James Wallman, Lukas Postler, Michael Meth, Roman Stricker, Esteban A. Martinez, Philipp Schindler, Thomas Monz, Joseph Emerson, and Rainer

Blatt. Characterizing large-scale quantum computers via cycle benchmarking. Nat. Commun., 10(1): 5347, 2019.

[11] Joel Wallman, Chris Granade, Robin Harper, and Steven T. Flammia. Estimating the coherence of noise. New Journal of Physics, 17:113020, 2015.

[12] Arnaud Carignan-Dugas, Dar Dahlen, Ian Hincks, Egor Ospadov, Stefanie J. Beale, Samuele Ferracin, Joshua Skanes-Norman, Joseph Emerson, and Joel J. Wallman. The error reconstruction and compiled calibration of quantum computing cycles. arXiv, 2303.17714, 2023.

[13] Boone, Kristine. Concepts and methods for benchmarking quantum computers. uwspace.uwaterloo.ca, 2021.

[14] Ying Li and Simon C. Benjamin. Efficient variational quantum simulator incorporating active error minimization. Phys. Rev. X, 7(2):021050, Jun 2017.

[15] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. Error mitigation for short-depth quantum circuits. Phys. Rev. Lett., 119(18):180509, Nov 2017.

[16] Samuele Ferracin, Akel Hashim, Jean-Loup Ville, Ravi Naik, Arnaud Carignan-Dugas, Hammam Qassim, Alexis Morvan, David I. Santiago, Irfan Siddiqi, and Joel J. Wallman. Efficiently improving the performance of noisy quantum computers. arXiv, 2201.10672, 2022.

[17] Zhenyu Cai, Ryan Babbush, Simon C. Benjamin, Suguru Endo, William J. Huggins, Ying Li, Jarrod R. McClean, and Thomas E. O'Brien. Quantum error mitigation. arXiv, 2210.00921, 2023.

[18] Joel J. Wallman and Joseph Emerson. Noise tailoring for scalable quantum computation via randomized compiling. Phys. Rev. A, 94:052325, 2016.

[19] Daniel Gottesman. An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation. Proceedings of Symposia in Applied Mathematics, 68:13, 2010.

[20] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. Physical Review A, 55:900, 1997.

[21] Eric Huang, Andrew C. Doherty, and Steven Flammia. Performance of quantum error correction with coherent errors. Physical Review A, 99(2):022313, 2019.

[22] John Watrous. The Theory of Quantum Information. Cambridge University Press, 2018.

[23] Matthew A. Graydon, Joshua Skanes-Norman, and Joel J. Wallman. Designing stochastic channels. 2201.07156, 2022.

[24] W. Fulton and J. Harris. Representation Theory: A First Course. Graduate Texts in Mathematics. Springer New York, 2011.

[25] Christian Dickel. Programming for the quantum computer. Qutech: Bits of Quantum, 2016.

[26] Easwar Magesan. Gaining Information About a Quantum Channel Via Twirling. uwspace.uwaterloo.ca, 2008.

[27] Leonardo Bautista-Gomez, Ferad Zyulkyarov, Osman Unsal, and Simon McIntosh-Smith. Unprotected computing: A large-scale study of dram raw error rate on a supercomputer. SC '16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2167-4337, 2016.

[28] William K. Wootters and Wojciech H Zurek. A single quantum cannot be cloned. Nature, 299:802-803, 1982.

[29] Daniel Gottesman. Surviving as a quantum computer in a classical world. Textbook manuscript, 2016.

[30] Christopher Chamberland, Joel Wallman, Stefanie Beale, and Raymond Laflamme. Hard decoding algorithm for optimizing thresholds under general markovian noise. Physical Review A, 95(4):042332, 2017.

[31] David Poulin. Tensor networks and quantum error correction. Physical Review Letters, 113:030501, 2014.

[32] David Poulin. Optimal and efficient decoding of concatenated quantum block codes. Physical Review A, 74:052333, 2006.

[33] Tomas Jochym-O'Connor and Raymond Laflamme. Using concatenated quantum codes for universal fault-tolerant quantum gates. Physical Review Letters, 112:010505, 2014.

[34] Stefan M. Moser and Po-Ning Chen. A student's guide to coding and information theory. Cambridge, 2012.

[35] Michael A. Nielsen. A simple formula for the average gate fidelity of a quantum dynamical operation. Physics Letters A, 303:249–252, 2002.

[36] Joel J. Wallman and Steven T. Flammia. Randomized benchmarking with confidence. New Journal of Physics, 16:103032, 2014.

[37] Salman Beigi and Robert König. Simplified instantaneous non-local quantum computation with applications to position-based cryptography. New Journal of Physics, 13:125202, 2011.

[38] Yuval R Sanders, Joel J. Wallman, and Barry C. Sanders. Bounding quantum gate error rate based on reported average fidelity. New Journal of Physics, 18:012002, 2015.

[39] Richard Keung, David M. Long, Andrew Doherty, and Steven T. Flammia. Comparing experiments to the fault-tolerance threshold. Physical Review Letters, 117:170502, 2016.

[40] Pavithran S. Iyer and David Poulin. A Small Computer is Needed to Optimize Fault-Tolerant Protocols. Quantum Sci. Technol., 3:030504, 2018.

[41] Pavithran Iyer, Aditya Jain, Stephen D. Bartlett, and Joseph Emerson. Efficient diagnostics for quantum error correction. Physical Review Research, 4(4):043218, 2022.

[42] Mauricio Gutiérrez, Conor Smith, Livia Lulushi, Smitha Janardan, and Kenneth R. Brown. Errors and pseudothresholds for incoherent and coherent noise. Physical Review A, 94:042338, 2016.

[43] Daniel Greenbaum and Zachary Dutton. Modeling coherent errors in quantum error correction. Quantum Science and Technology, 3:015007, 2017.

[44] Andrew S. Darmawan and David Poulin. Tensor-network simulations of the surface code under realistic noise. Physical Review Letters, 119(4):040501, 2017.

[45] Sergey Bravyi, Matthias Englbrecht, Robert Koenig, and Nolan Peard. Correcting coherent errors with surface codes. 1710.02270, 2017.

[46] Stefanie J. Beale, Joel J. Wallman, Mauricio Gutiérrez, Kenneth R. Brown, and Raymond Laflamme. Quantum error correction decoheres noise. Physical Review Letters, 121(19):190501, 2018.

[47] Zhenyu Cai, Xiaosi Xu, and Simon C. Benjamin. Mitigating Coherent Noise Using Pauli Conjugation. npj Quantum Inf, 6(17), 2019.

[48] Joseph Iverson and John Preskill. Coherence in logical quantum channels. New Journal of Physics, 22:073066, 2020.

[49] Giuliano G. La Guardia. Quantum Codes Derived from Cyclic Codes. International Journal of Theoretical Physics, 56:2479-2484, 2017.

[50] Markus Grassl and Martin Roetteler. Leveraging automorphisms of quantum codes for fault-tolerant quantum computation. IEEE Transactions on Automatic Control, 534-538, 2013.

[51] Rui Chao and Ben W. Reichardt. Fault-tolerant quantum computation with few qubits. npj Quantum Information, 4:42, 2018.

[52] Benjamin Rahn, Andrew C. Doherty, and Hideo Mabuchi. Exact performance of concatenated quantum codes. Physical Review A, 66:032304, 2002.

[53] E. Knill. Group Representations, Error Bases and Quantum Codes. OSTI.GOV, 1996.

[54] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. Physical Review A, 86(3):032324, 2012.

[55] Héctor Bombín and M. A. Martin-Delgado. Topological Quantum Distillation. New Journal of Physics, 97(18):180501, 2006.

[56] Héctor Bombín. Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes. New Journal of Physics, 17:083002, 2015.

[57] Sergey Bravyi, Sarah Sheldon, Abhinav Kandala, David C. Mckay, and Jay M. Gambetta. Mitigating measurement errors in multiqubit experiments. Phys. Rev. A, 103(4):042605, 2021.

[58] Robert Raussendorf and Hans J. Briegel. A one-way quantum computer. Phys. Rev. Lett., 86(22):5188–5191, May 2001.

[59] Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. Physical Review A, 80(1):012304, 2009.

[60] Joel J. Wallman. Randomized benchmarking with gate-dependent noise. Quantum, 2:47, 2018.

[61] Christopher Granade, Christopher Ferrie, and David G. Cory. Accelerated Randomized Benchmarking. New J. Phys., 17:013042, 2014.

[62] M. Fukuda and A. S. Holevo. On weyl-covariant channels. arXiv, 0510148, 2006.

[63] Jesse Fern. Correctable noise of quantum-error-correcting codes under adaptive concatenation. Phys. Rev. A, 77(1):010301, 2008.

[64] Luka Skoric, Dan E. Browne, Kenton M. Barnes, Neil I. Gillespie, and Earl T. Campbell. Parallel window decoding enables scalable fault tolerant quantum computation. arXiv, 2209.08552, 2023.

[65] Aditya Jain, Pavithran Iyer, Stephen D. Bartlett, and Joseph Emerson. Improved quantum error correction with randomized compiling. Phys. Rev. Research, 5(3):033049, 2023.

[66] A. Greene, M. Kjaergaard, M. E. Schwartz, G. O. Samach, A. Bengtsson, M. O'Keeffe, D. K. Kim, M. Marvian, A. Melville, B. M. Niedzielski, A. Vepsalainen, R. Winik, J. Yoder, D. Rosenberg, S. Lloyd, T. P. Orlando, I. Marvian, S. Gustavsson, and W. D. Oliver. Error mitigation via stabilizer measurement emulation. arXiv, 2102.05767, 2021.

[67] Jonas T. Anderson, Guillaume Duclos-Cianci, and David Poulin. Fault-tolerant conversion between the steane and reed-muller quantum codes. Physical Review Letters, 113(8):080501, 2014.

[68] Wan-Jung Kuo and Daniel A. Lidar. Quadratic dynamical decoupling: Universality proof and error analysis. Physical Review A, 84(4):042329, 2011.

[69] Imdad S. B. Sardharwalla, Toby S. Cubitt, Aram W. Harrow, and Noah Linden. Universal refocusing of systematic quantum noise. arXiv, 1602.07963, 2016.

[70] Arnaud Carignan-Dugas, Joel J. Wallman, and Joseph Emerson. Characterizing universal gate sets via dihedral benchmarking. Phys. Rev. A, 92(6):060302, 2015.

[71] Joshua Combes, Christopher Granade, Christopher Ferrie, and Steven T. Flammia. Logical randomized benchmarking. arXiv, 1702.03688, 2017.

[72] Athena Ceasura, Pavithran Iyer, Joel J. Wallman, and Hakop Pashayan. Non-exponential behaviour in logical randomized benchmarking. arXiv, 2212.05488, 2022.