Improved Diagnostics & Performance for Quantum Error Correction

by

Aditya Jain

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Applied Mathematics (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:	Prof. Dan Browne Department of Physics & Astronomy, University College London
Supervisor:	Prof. Joseph Emerson Department of Applied Mathematics, University of Waterloo
Internal Member:	Prof. Achim Kempf Department of Applied Mathematics, University of Waterloo
Internal-External Member:	Prof. Raymond Laflamme Department of Physics & Astronomy, University of Waterloo
Other Member:	Prof. Joel Wallman Department of Applied Mathematics, University of Waterloo

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 following table explicitly states my contributions to the contents presented in this thesis. This section is intended for the evaluation committee of this thesis.

Chapter	Author's contribution
Efficient diagnostics for quantum error correction [IJB+22]	This is one of the main chapters in the thesis; the author has contributed equally in part with Pavithran Iyer for the development of ideas, the implementation of the numerical work and writing of the article.
Improved quantum error correction with randomized compiling [JIB+23]	The author led this project and has contributed towards the development of ideas, the analytical calculations, numerical simulations, and in part, writing of the manuscript.
Testing distance of codes	This project was conceived as a result of a casual chat of the author with Sathyawageeswar. The author has contributed significantly towards the development of ideas with assistance from Sathyawageeswar. In particular, the author contributed towards the definition of the problem, the derivation of bounds via reductions, and in part, writing of the chapter. Specifically, the author received support from Sathyawageeswar in writing section 4.3 of the chapter.

Abstract

Building large scale quantum computers is one of the most exciting ventures being pursued by researchers in the 21^{st} century. However, the presence of noise in quantum systems poses a major hindrance towards this ambitious goal. Unlike the developmental history of classical computers where noise levels were brought under reasonable threshold levels early on, the field of quantum computing is struggling to do the same. Nonetheless, there have been many significant theoretical and experimental advancements in the past decade. Quantum error correction and fault tolerance in general is believed to be a reliable long term strategy to mitigate noise and perform arbitrarily long quantum computations. Optimizing and assessing the quality of components in fault-tolerance scheme is a crucial task. We address these tasks in this thesis.

In the first part of the thesis, we provide a method to efficiently estimate the performance of a large class of codes called concatenated stabilizer codes. We show how to employ noise tailoring techniques developed for computations at the physical level to circuits protected by quantum error correction to enable this estimation. We also develop a metric called the *logical estimator*, which is an approximation of the logical infidelity of the code. We show that this metric can be used to guide the selection of the optimal (concatenated stabilizer) code and the optimal (lookup style) decoder for a given device. Moreover, the metric also aids in estimating the resource requirements for a target logical error rate efficiently and reliably.

In the second part, we show how a combination of noise tailoring tools with quantum error correction can improve the performance of concatenated stabilizer codes by several orders of magnitude. These gains in turn bring down the resource overheads for quantum error correction. We explore the gains using concatenated Steane code under a wide variety of physically motivated error models including arbitrary rotations and combinations of coherent and stochastic noise. We also study the variation of gains with the number of levels of concatenation. For the simple case of rotations about a Pauli axis, we show that the gain scales doubly exponentially with the number of levels in the code. We analyze and show the presence of threshold rotation angles below which the gains can be arbitrarily magnified by increasing the number of levels in the code.

The last part of the thesis explores the testing of an important property of error correcting codes - the *minimum distance*, often referred to as the distance. We operate in the regime of large classical binary linear codes described in terms of their parity check matrices. We are given access to these codes in terms of an oracle which when supplied an index, returns a single column of the parity check matrix corresponding to that index.

We derive lower and upper bounds on the query complexity of finding the minimum distance of a given code. We also ask and (partially) answer the same question in the property testing framework. In particular, we provide a tester which queries a sublinear number of columns of the parity check matrix and certifies whether a code has high distance or is far away from all codes which have high distance. We also provide non-trivial lower bounds for this task. Although this study is done for classical linear codes, it has implications for designing quantum codes which are built using classical codes. This part of the thesis defines the beginning of a significant area of interest encompassing efficiently testing important properties of classical and quantum codes.

Acknowledgements

There are a lot of people who have made the journey behind this thesis possible. I would like to thank Joseph Emerson for giving me the opportunity to come all the way from India for an internship and offering a PhD position right after. Joseph has been really supportive in letting me explore a wide range of topics. He not only helped me with his valuable inputs throughout but also was very active in connecting me with other experts in the field for collaborations. Going to conferences and connecting with other researchers is an important part of a student's journey and Joseph was extremely supportive in these endeavours. I'd like to thank all the examiners for their valuable inputs and suggestions. I want to thank Stephen Bartlett for his vision and valuable insights that improved the quality of all the projects significantly. Thank you Steve for attending the 7 am meetings. I learnt a lot from working closely with Pavithran on all the projects that saw the light of the day and many that didn't go anywhere. Pavi helped simplify a lot of the concepts for me and encouraged me to push boundaries. I would like to thank John Watrous and Daniel Gottesman for offering two of the most influential courses I took during this period. I also benefited a lot from discussions with Joel Wallman on various topics. Sathyawageeswar Subramanian and Tom Gur played a key role in my entry into complexity theory. I thank Sathya for being patient in discussions and giving important feedback on the write-ups. I would like to thank all of Joseph's and Joel's group members for making research both interesting and fun - Stefanie, Kristine, Arnaud, Hammam, Athena, Jaron, Samuele, Anthony and others. I want to thank the administrative folks at IQC and AMATH for their smooth assistance.

Moving into a new country is tough but I was fortunate to have some of the best friends to help me settle here effortlessly. Thank you Rahul, Manjit, Júlia, Nachiket, Anand, Nikhil, Irene, Vadiraj, Jérémy, Sreesh, Daniel. Thank you for the countless memories and the time spent together. Thank you Karen and Soumik for the extended debates and discussions on the various topics under the sun. Thank you Michael, Alex and Harmen for all the adventures. A shout out to all the folks who helped me push myself at the climbing gym. The journey in quantum started in undergrad and I'd like to thank all the friendships initiated then including Agni, Bharat and the entire third floor crew. I want to thank all the friends back home who are just a call away for literally everything. Thank you Yash, Rahul, Juhi, Harsh, and Natasha.

This is one of the rare opportunities to thank all the family members who form the foundation of all this. Thank you Rahul, Prerna, Rohit, and the most special of all - my nephew Arham. Rahul has been not just an elder brother, but a mentor and a friend at the same time. Finally, thank you *mummy* and *papa* for all that you have done and continue to do for all of us selflessly.

Dedication

This is dedicated to *my family*.

Table of Contents

List of Figures

List of Tables

Nomenclature

Notation	Description
$[n]$	$\{1, 2, \ldots, n\}$. 7
$[m, n]$	$\{m, m+1, \ldots, n\}$. 7
$\mathrm{L}(\mathcal{X}, \mathcal{Y})$	Linear mapping between spaces $\mathcal{X}$ and $\mathcal{Y}$. 7
$A^\star$	The operator obtained by taking the complex conjugate of each entry of A. 8
A^T	The operator obtained by tranposing the matrix for the operator A. 8
$A^\dagger$	The operator obtained by taking the complex conjugate of each entry and then transposing the matrix for the operator A. 8
e_a	Basis vector with a^{th} entry equal to 1 and all the other entries equal to 0. 9
$\\|A\\|_p$	Schatten $p-$norm of any operator A. 9
$\mathcal{E}$	Quantum channel. 10
$\Gamma(\mathcal{E})$	Liouville representation of the quantum channel $\mathcal{E}$. 11
$J(\mathcal{E})$	Choi representation of the quantum channel $\mathcal{E}$. 12
$\chi(\mathcal{E})$	Chi representation of the quantum channel $\mathcal{E}$. 13
RC	Randomized Compiling. 24
CER	Cycle Error Reconstruction. 27
QEC	Quantum error correction. 33
MWD	Minimum weight decoding. 38
MLD	Maximum likelihood decoding. 38
FT	Fault tolerance. 48
$\overline{r}$	Average *logical* infidelity. 52
$\widetilde{p}_u$	The logical estimator for concatenated codes. 56
$R(f)$	The classical randomized query complexity of the function f. 84
$Q(f)$	The quantum query complexity of the function f. 85

Quote

"Education is the kindling of a flame, not the filling of a vessel."

– Socrates

Chapter 1

Introduction

The idea of quantum computation originated in a talk by Richard Feynman at a conference [Fey82; Pre23] in 1981 where he proposed using a new kind of computer to simulate quantum systems. The primary roadblock in the simulation of quantum physics using (present-day) digital computers is that the description of quantum systems using classical computers requires too many variables. Feynman was further interested in learning which sub-classes of quantum mechanical systems could be simulated efficiently by the existing classical computers. Since then the study of quantum computing as a field has intrigued not just physicists but also researchers in mathematics, computer science, chemistry and engineering. The community developed the mathematical models that describe the components (quantum bits and quantum gates) of a computation using the laws of quantum physics [Pre99; NC10]. These components have been realized in practice using a wide variety of architectures including superconducting platform, ion-traps and photonic sources [HRP+06; CZ95; BL05]. There have been many impressive experimental demonstrations of large quantum circuits which consist of quantum bits and quantum gates [AAB+19; ZDQ+21].

There was a spike in the interest in quantum computing with the discovery of exponentially fast quantum algorithms for contrived problems first [Deu85; BV93; Sim97] and soon after for factoring large numbers by Peter Shor [Sho99]. Although quantum computing promises to revolutionize multiple domains such as cryptography, drug design, finance, and logistics [BL17; OIS+22; OML19; HGT+21], there are two big hurdles towards using quantum computers for performing meaningful tasks that classical computers cannot. First, it has been established that we would require thousands of functional quantum bits that can execute millions of quantum gates to use the quantum algorithms at a scale that surpasses the capability of current day supercomputers

[CMN+18]. Second, the presence of noise in the current day quantum devices vastly limits their usefulness. Understanding the details of this noise and eliminating the same has been an area of intense focus in the past decade. Modern noise characterization, noise tailoring, error mitigation and quantum error correction tools have been developed to address this problem [CDDH+23; WE16; CBB+22; SER+23].

Error correction involves using redundant resources to encode information in order to detect and correct errors when they occur. It is used widely in classical communication and the core idea dates back to 1948 when Claude Shannon published an article titled "A Mathematical Theory of Communication" [Sha48]. The development of quantum error correction borrows a lot from classical coding theory. However, the challenges in dealing with errors and the solutions to these in the quantum computing world are significantly different. The general header under which all the operations in a quantum device are ensured to be error-free is called *fault-tolerance*. The resource overheads associated with fault-tolerant quantum computation are high. Reducing these overheads and optimizing the components of a fault-tolerance scheme in accordance with the physical architecture and varying noise profiles is an active area of ongoing research [Got14; WBP15; CR18b; YK22]. Before designing and adopting a fault tolerant scheme, it is absolutely crucial to have tools to estimate, and optimize, the performance of, and resources required for different schemes.

The fault tolerance accuracy threshold theorem [AGP07; CTV17] is commonly oversimplified as specifying a threshold on gate error rates that must be reached. There are several major shortcoming to this simplification. The most significant is that each fault tolerant threshold is derived under very strong, and often physically unrealistic, conditions on the error model, such as the absence of correlations and coherence in the errors. Another concern is that the metric usually invoked for assessing error rates is the diamond distance, which can not be measured in a scalable way and measuring it even for a single qubit is extremely resource intensive under arbitrary error models. A third is that the resource and overhead requirements for implementing the fault tolerant scheme depend critically on the precise relationship between the error model and the fault-tolerant scheme.

A fault tolerant scheme relies on a quantum error correction routine – which in essence helps to mitigate the effect of physical noise processes on logical quantum information. Hence, the task of estimating the performance of a quantum error correction routine is inherent to the task of estimating the overhead required to build a fault tolerant quantum computer. It is in general a resource intensive module of a fault tolerant quantum computer. There are several variables that specify the choice of a quantum error correction scheme, including an error correcting code and a decoding algorithm.

Often, a hardware architecture imposes restrictions on the local geometry of the code. For instance, if the qubit connectivity resembles a square lattice, the structure of the code – its stabilizer generators – must also respect the same geometry. However, there is still a freedom to choose between codes of irregular connectivity and varying sizes. Another quantity of interest for choosing a code is the trade-off between the number of physical qubits per logical qubit and its distance. The role of a decoding algorithm is also crucial to the logical error rate. The optimal choice for the various components of a quantum error correction scheme often depends on the properties of the underlying physical noise process. For example, the setting discussed in [TBF18], which deals with biased noise, is best suited for a code which can correct more of one type of errors. A bias is only one of the exponentially many parameters that describe the evolution of a system of n qubits. The general case presents a fundamental challenge towards understanding the key properties of the noise process that severely impact the performance of a quantum error correction scheme. So far, standard metrics of physical noise such as the diamond distance and infidelity have been ruled out as critical parameters. In this thesis, we will present an efficiently measurable quantity for concatenated codes that can be used to accurately predict the performance of the code. This serves as a feedback to make better choices for the components of the quantum error correction scheme.

Since the design of a fault-tolerant scheme is innately linked to noise modelling, the following optimization cycle was proposed in Ref. [Iye18] (and references therein):

(i) Experimental noise characterization of a device,

(ii) Noise modelling,

(iii) Fault tolerant protocol design tailored to the model, and

(iv) Numerical benchmark of protocol.

The authors argued that the above cycle is not viable in general. Specifically, it was shown in [IP18] that the performance of a quantum error correcting code depends strongly on the microscopic details of the underlying noise process and consequently that the logical error is not well predicted by typical figures of merit, such as average gate infidelity or diamond norm measured by randomized benchmarking [EAŻ05; DCE+09; MGE11] or gate set tomography [BKGN+17]. A general, Markovian noise process affecting a single qubit is completely specified already by 12 independent parameters [Woo09; DZP19] and this number grows exponentially for multi-qubit operations, due to correlated errors for instance, and hence methods for full error characterization are impractical beyond a few qubits.

In this thesis, we address some of the concerns raised in [Iye18] with respect to the feasibility of the above optimization cycle for the class of concatenated stabilizer codes. We devise methods to predict the performance of quantum error correction schemes. This involves using partial knowledge of the noise process obtained from experiments to efficiently approximate the logical performance. The knowledge obtained leads to a better modelling in step (ii) of the above cycle. The efficient and reliable prediction helps in designing better error correction schemes (thereby improving step (iii)). Finally, using the tools developed, one can benchmark the logical performance more accurately (step (iv)). In addition, we explore ways to tailor the physical noise to achieve better logical performance, and we explore fast algorithms to test the minimum distance of linear codes. These studies significantly improve step (iii) of the above cycle.

The thesis is structured as follows. To begin with, in chapter 2, we introduce the relevant background material required to understand the results presented in the thesis. The predictability of quantum error correction schemes is improved using information about the underlying physical noise processes that can be efficiently obtained from an experiment. To bring the problem into a more tractable form, we recall a noise tailoring technique known as Randomized Compiling (RC) [WE16]. Here, the gates in a circuit are compiled with random Pauli operations, thereby leading to an effective Pauli noise on the underlying qubits. This enables us to focus our attention to predicting the performance of quantum error correcting codes, under the effect of physical Pauli noise processes. The calibration data that we have access to for the Pauli noise processes are the probabilities of various Pauli errors, which can be efficiently extracted using noise reconstruction techniques [EWP+19; HFW20; CDDH+23]. Chapter 3 describes the above procedure along with other theoretical methods developed to improve the predictability of logical performance. In chapter 4, we show how randomized compiling can be used to improve the performance of quantum error correcting codes by several orders of magnitude. This in turn brings down the resource overheads required for achieving fault tolerance. Chapter 5 discusses super fast classical and quantum algorithms to test a crucial property of linear codes - *the minimum distance*. In this chapter, we derive classical and quantum query complexity bounds for finding the minimum distance of linear codes. We also derive similar bounds in the property testing framework where the input code is promised to either have high distance or be far from the set of codes having high distance. Finally, in chapter 6 we provide concluding remarks and list some interesting open problems for future research.

Chapter 2

Background

Quantum computations are performed by executing circuits which comprise a series of quantum gates applied to a set of quantum systems. These gates are implemented by applying unitary operations via evolution under Hamiltonians. Each Hamiltonian has two components - a system Hamiltonian H_0, and the set of control Hamiltonians $\{H_1, H_2, \dots H_k\}$. Although the evolution is continuous, we can approximate it using discrete time steps. The Hamiltonian at a time $1 \leq j \leq N$ is given by

$$H(j) = H_0 + \sum_{i=1}^{k} u_i(j) H_i, \tag{2.1}$$

where $u_i(j) \in \mathbb{R}$ is the control amplitude for H_i at time step j. A common example of system Hamiltonian is $H_0 = \sigma_Z$ with control along X and Y axis i.e., $H_1 = \sigma_X, H_2 = \sigma_Y$. The piece-wise approximation for the resulting unitary is given by

$$U = \Pi_{i=1}^{N} e^{-iH(j)dt}, \tag{2.2}$$

where $T = Ndt$ is the total evolution time. So far we have described the evolution of a single closed quantum system. Ideally, if we were able to control our system of interest perfectly, we would be able to perform large-scale arbitrarily long quantum computations easily. However, the system of interest interacts with the environment in undesirable ways. These interactions between a quantum system and its environment manifest as noise processes affecting the system. The sources of noise can vary depending on the setup including imperfect pulses, unknown stray magnetic fields, photon losses, etc. In its most general form, the dynamics of a noisy quantum computer are described by a joint time evolution of the system and its environment. The study of these topics falls under the umbrella term of *open quantum systems* [RH12]. We will refer to everything except the system of interest as the environment. Let us denote the density matrix of the combined quantum state of the system and the environment by ρ_{SE}. We assume that the system is initially decoupled from the environment initially and their respective initial states are given by ρ_S and $|e\rangle\langle e|$ so that the initial state of the joint system is $\rho_{SE} = \rho_S \otimes |e\rangle\langle e|$. The initial state of the environment is assumed to be a pure state without loss of generality since a purification always exists. The combined state after evolution by a unitary $\mathcal{U}$ acting on the joint system is given by

$$\rho'_{SE} = \mathcal{U}\ (\rho \otimes |e\rangle\langle e|)\ \mathcal{U}^\dagger. \tag{2.3}$$

The state of the system only after the evolution is obtained by tracing out the environment from the combined state i.e.,

$$\rho'_S = \mathrm{tr}_E(\rho_{SE}), \tag{2.4}$$

where Tr_E denotes the partial trace over the subspace E (in this case the environment). The partial trace can be expressed in terms of an orthonormal basis of the system being traced over as

$$\rho'_S = \sum_k \langle e_k|\rho_{SE}|e_k\rangle, \tag{2.5}$$

where $\{|e_k\rangle\}$ is an orthonormal basis for the environment space. We denote the effective evolution of our system when the combined system-environment pair evolves under $\mathcal{U}$ by $\rho_S = \mathcal{E}_{\mathcal{U}}(|\psi\rangle\langle\psi|)$. Assuming orthonormal bases for the Hilbert space of the system and environment space to be $\{s_i\}$ and $\{e_k\}$ respectively, we can rewrite the entries of the evolved state of the system as

$$\begin{aligned}
\langle s_i|\mathcal{E}_{\mathcal{U}}(\rho_S)|s_j\rangle &= \mathrm{tr}_E(\rho_{SE}) \\
&= \mathrm{tr}_E(\mathcal{U}\ (\rho_S \otimes |e\rangle\langle e|)\ \mathcal{U}^\dagger) \\
&= \sum_{m,n}\sum_k \langle s_i, e_k|\mathcal{U}|s_m, e\rangle (\rho_S)_{m,n} \langle s_n, e|\mathcal{U}^\dagger|s_j, e_k\rangle \\
&= \sum_k \left[A_k \rho_S A_k^\dagger\right]_{i,j},
\end{aligned} \tag{2.6}$$

where $A_k = \langle e_k|\mathcal{U}|e_k\rangle$ is an operator acting on the system only. This simple derivation leads us to our first representation of noise processes called the *Kraus* representation. The operators $\{A_k\}$ are referred to as the Kraus operators and completely describe the noise process acting on the system of interest. A Markovian model of noise assumes that the environment is memory-less and leads to the convenient description of its effect on the system by completely positive trace preserving (CPTP) maps [Cho75; Hay17]. This is a standard assumption about the noise acting in quantum systems.

In this chapter, we will describe the background information necessary to understand the material presented in the rest of this thesis. We assume some familiarity with quantum information and quantum computation. If the reader is completely unfamiliar with notions of quantum information, we refer them to Refs. [NC10; Pre99]. This thesis deals with two broad topics - quantum error characterization and quantum error correction. We have attempted to cover all the necessary background required for these topics. For specific details, we refer the readers to the papers mentioned in the various sections. In section 2.1 we describe the basic mathematical preliminaries which consist of basic linear algebra tools to describe quantum objects such as states and channels. Next, we describe representations of noise processes and the distance metrics associated to them in sections 2.2 and 2.3 respectively. In section 2.4 we discuss some noise characterization and tailoring tools. Section 2.5 provides an introduction to classical and quantum error correction with focus on stabilizer codes and concatenated codes. Finally, in section 2.6, we provide an introduction to the area of property testing.

2.1 Mathematical preliminaries

There are several equivalent ways to represent noise processes, each of which may prove useful in different contexts. Before moving forward to the representations of noise processes, it is helpful to lay down some notation. This notation is a slightly adapted version of the notation set in Ref. [Wat18]. In this thesis, we will only deal with finite dimensional Euclidean spaces which will be referred to as $\mathcal{X}, \mathcal{Y}$ and so on. Typically $\mathcal{X} = \mathbb{C}^d$ for some integer $d \geq 2$. We will use $[n]$ and $[m, n]$ to denote the sets $\{1, 2, \ldots, n\}$ and $\{m, m+1, \ldots, n\}$ respectively.

2.1.1 Linear operators

The set of linear mappings between the spaces $\mathcal{X}$ and $\mathcal{Y}$ is denoted by $\mathrm{L}(\mathcal{X}, \mathcal{Y})$. Often we will associate $\mathcal{X}$ and $\mathcal{Y}$ with Hilbert spaces which store the quantum state vectors and hence the operator will take a state in $\mathcal{X}$ to a state in $\mathcal{Y}$. In these cases, they will be denoted more intuitively as $\mathcal{H}_1$ and $\mathcal{H}_2$ respectively. The set $\mathrm{L}(\mathcal{X}, \mathcal{Y})$ forms a complex vector space when addition and scalar multiplication are defined as follows:

1. Addition: For all $A, B \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$, the operator $A + B \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ is defined by:
$$(A + B)x = Ax + Bx \tag{2.7}$$
for all $x \in \mathcal{X}$.
2. Scalar multiplication: For all $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ and a scalar $\alpha \in \mathbb{C}$, the operator $\alpha A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ is defined by:
$$(\alpha A)x = \alpha Ax \tag{2.8}$$
for all $x \in \mathcal{X}$.

We will abbreviate $\mathrm{L}(\mathcal{X}, \mathcal{X})$ as $\mathrm{L}(\mathcal{X})$. All linear operators will have associated matrix representations and we may use the same name for both interchangeably. It will be clear from context if we are talking about the abstract object or the matrix representation in a given basis. A_{ab} will refer to the entry in the a^{th} row and b^{th} column of the matrix A. For every matrix of an operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ defined in a given basis, we define three more operators:

1. The operator $A^{\star} \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ will refer to the operator obtained by taking the complex conjugate of each entry of A i.e.,
$$A^{\star}_{ab} = \overline{A_{ab}}. \tag{2.9}$$
2. The operator $A^{T} \in \mathrm{L}(\mathcal{Y}, \mathcal{X})$ will refer to the operator obtained by transposing the matrix for operator A i.e.,
$$A^{T}_{ab} = A_{ba}. \tag{2.10}$$
3. The operator $A^{\dagger} \in \mathrm{L}(\mathcal{Y}, \mathcal{X})$ will refer to the operator obtained by taking complex conjugate of each entry and then transposing the matrix for operator A i.e.,
$$A^{\dagger}_{ab} = \overline{A_{ba}}. \tag{2.11}$$
It is easy to see that $A^{\dagger} = (A^{\star})^{T} = (A^{T})^{\star}$.

An operator $X \in \mathrm{L}(\mathcal{X})$ is said to be *positive semi-definite* if it holds that $X = Y^{\dagger}Y$ for some $Y \in \mathrm{L}(\mathcal{X})$. We will denote the set of positive semi-definite linear operators by $\mathrm{Pos}(\mathcal{X})$ defined by

$$\mathrm{Pos}(\mathcal{X}) = \{Y^{\dagger}Y : Y \in \mathrm{L}(\mathcal{X})\}. \tag{2.12}$$

Positive semi-definite operators with unit trace are called density operators. They belong to $\text{Pos}(\mathcal{X})$ and the set of all of them will be denoted by $\text{D}(\mathcal{X})$, where

$$\text{D}(\mathcal{X}) = \{\rho \in \text{Pos}(\mathcal{X}) : \text{Tr}(\rho) = 1\}. \tag{2.13}$$

Let $e_a \in \mathcal{X}$ and $e_b \in \mathcal{Y}$ be vectors with the a^{th} and b^{th} entry equal to 1 respectively while all the other entries equal to 0. We will denote the elementary matrices by $E^{a,b}$ such that $E^{a,b} = e_a e_b^\dagger$. They form a basis of $\text{L}(\mathcal{X}, \mathcal{Y})$ called the *standard basis*.

We will denote the Schatten $p-$norm $\|A\|_p$ of any operator $A \in \text{L}(\mathcal{X}, \mathcal{Y})$ by the following expression:

$$\|A\|_p = (\text{Tr}(A^\dagger A)^{p/2})^{1/p} \tag{2.14}$$

for all $p \geq 1$.

2.1.2 Operator vector correspondence

There is a one-to-one correspondence between the space $\text{L}(\mathcal{Y}, \mathcal{X})$ and the space $\mathcal{X} \otimes \mathcal{Y}$. The correspondence we will use in this thesis is given by the mapping

$$\text{vec} : \text{L}(\mathcal{Y}, \mathcal{X}) \to \mathcal{X} \otimes \mathcal{Y}, \tag{2.15}$$

and is defined as

$$\text{vec}(E^{a,b}) = e_a \otimes e_b.$$

In the literature, this is sometimes referred to as row vectorization because it involves flattening out a matrix one row at a time. It is easy to see via linearity that

$$\text{vec}(uv^\dagger) = u \otimes v^\star \tag{2.16}$$

for all $u \in X$ and $v \in \mathcal{Y}$. The special cases obtained by plugging scalars $u = 1$ and $v = 1$ respectively are

$$\text{vec}(v^\dagger) = v^\star \text{ and } \text{vec}(u) = u. \tag{2.17}$$

The following identities are useful with respect to vectorization of operators:

1. The vec mapping is an isometry. Every $u \in \mathcal{X} \otimes \mathcal{Y}$ defines a linear operator $A \in \text{L}(\mathcal{Y}, \mathcal{X})$. The inner product is preserved in this mapping i.e.,

$$\langle A, B \rangle = \langle \text{vec}(A), \text{vec}(B) \rangle \tag{2.18}$$

   for all $A, B \in \text{L}(\mathcal{Y}, \mathcal{X})$.

2. The vec of a product of three matrices $A_0 \in \mathrm{L}(\mathcal{X}_0, \mathcal{Y}_0), B \in \mathrm{L}(\mathcal{X}_1, \mathcal{X}_0)$ and $A_1 \in \mathrm{L}(\mathcal{Y}_1, \mathcal{X}_1)$ is given by

   $$\mathrm{vec}(A_0 B A_1) = (A_0 \otimes A_1^T)\,\mathrm{vec}(B). \tag{2.19}$$

   This identity is called Roth's lemma [HJ85].

3. The following are true with respect to partial traces:

   $$\begin{aligned} \mathrm{Tr}_{\mathcal{Y}}(\mathrm{vec}(A)\,\mathrm{vec}(B)^\dagger) &= AB^\dagger, \\ \mathrm{Tr}_{\mathcal{X}}(\mathrm{vec}(A)\,\mathrm{vec}(B)^\dagger) &= A^T B^\star, \end{aligned} \tag{2.20}$$

   for all operators $A, B \in \mathrm{L}(\mathcal{Y}, \mathcal{X})$.

Apart from the row stacking convention, we consider the following two more ways of vectorizing operators:

1. Column stacking convention: For an operator $A \in \mathrm{L}(\mathcal{Y}, \mathcal{X})$, the vectorization in the column stacking convention is defined by the following action on elementary matrices:

   $$[\mathrm{vec}(E^{a,b})]_c = e_b \otimes e_a. \tag{2.21}$$

2. Sometimes we may choose to vectorize with respect to an orthonormal basis $\{B_\alpha\}$, in which case the vectorization map is:

   $$[\mathrm{vec}(E^{a,b})]_B = \sum_\alpha \mathrm{Tr}(B_\alpha^\dagger E^{a,b}) e_\alpha. \tag{2.22}$$

The action on arbitrary operators can be obtained by taking a linear combination of the actions on elementary operators.

2.1.3 Superoperators or quantum channels

Finally the maps, denoted often by $\mathcal{E}$, that transform one linear operator to another i.e.,

$$\mathcal{E} : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y}) \tag{2.23}$$

are common in quantum information. We will denote the set of all such operators by $\mathrm{T}(\mathcal{X}, \mathcal{Y})$. We will again use the abbreviation $\mathrm{T}(\mathcal{X})$ in place of $\mathrm{T}(\mathcal{X}, \mathcal{X})$. A map $\mathcal{E} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})$ is said to be a completely positive trace preserving (CPTP) map or a *quantum channel* (referred to as simply channel sometimes) if:

1. $\mathcal{E}$ is *completely positive* i.e., $(\mathcal{E} \otimes \mathcal{I}_{\mathrm{L}(\mathcal{Z})})(P) \in \mathrm{Pos}(\mathcal{Y} \otimes \mathcal{Z})$ for all $P \in \mathrm{Pos}(\mathcal{X} \otimes \mathcal{Z})$ and any complex Euclidean space $\mathcal{Z}$, and

2. $\mathcal{E}$ is *trace preserving* i.e., $\mathrm{Tr}(\mathcal{E}(X)) = \mathrm{Tr}(X)$ for all $X \in \mathcal{X}$.

The set of all CPTP maps or quantum channels is denoted by $\mathrm{C}(\mathcal{X}, \mathcal{Y})$. Again $\mathrm{C}(\mathcal{X})$ stands for $\mathrm{C}(\mathcal{X}, \mathcal{X})$.

2.2 Representations of noise processes

In this section we will define the various representations of noise maps. Each representation has its own set of advantages and reveals interesting properties of the channels. It is also easy to convert between different representations as needed. As we will notice in later chapters, we will invoke different representations at various points to make the best use of their characteristic traits. Let $\mathcal{H} = \mathbb{C}^d$ describe the Hilbert space for a $d-$level quantum system where $d \geq 2$. For n-qubit systems $d = 2^n$. For simplicity, in this section, we will describe channels that map states between Hilbert spaces of similar dimension. However, most of these tools can be applied in scenarios where this condition is not true.

2.2.1 Liouville representation

Through this representation, we want to establish a map between the vectorizations of $\rho \in \mathrm{D}(\mathcal{H})$ and $\mathcal{E}(\rho)$ i.e., we want to find a map $\Gamma(\mathcal{E})$ such that

$$\mathrm{vec}(\rho) \xrightarrow{\Gamma(\mathcal{E})} \mathrm{vec}(\mathcal{E}(\rho)), \tag{2.24}$$

for all $\rho \in \mathrm{D}(\mathcal{H})$ and $\mathcal{E} \in \mathrm{C}(\mathcal{H})$. It is easy to see that this map is linear. This implies the existence of a linear operator $\Gamma(\mathcal{E}) \in \mathrm{L}(\mathcal{H} \otimes \mathcal{H}, \mathcal{H} \otimes \mathcal{H})$ such that the following is true:

$$\Gamma(\mathcal{E})\,\mathrm{vec}(\rho) = \mathrm{vec}(\mathcal{E}(\rho)) \tag{2.25}$$

for all $\rho \in \mathrm{D}(\mathcal{H})$. Eq.(2.25) is the defining equation for the Liouville representation $\Gamma(\mathcal{E})$ of the map $\mathcal{E}$. The unique operator for which this condition is true for all $\rho \in \mathrm{D}(\mathcal{H})$ will define the action of the map, since the equation describes a way to get the vectorized output of the channel on all possible inputs. Note that the map in Eq.(2.25) is linear:

$$\Gamma(\alpha\mathcal{E}_1 + \beta\mathcal{E}_2) = \alpha\Gamma(\mathcal{E}_1) + \beta\Gamma(\mathcal{E}_2), \tag{2.26}$$

for all $\alpha, \beta \in \mathbb{C}$ and $\mathcal{E}_1, \mathcal{E}_2 \in \mathrm{C}(\mathcal{H})$.

Now, lets turn to the advantages of this representation. The Liouville representation allows us to seamlessly compose two different channels acting on a system i.e.,

$$\Gamma(\mathcal{E}_1 \circ \mathcal{E}_2) = \Gamma(\mathcal{E}_2).\Gamma(\mathcal{E}_1), \tag{2.27}$$

for all $\mathcal{E}_1, \mathcal{E}_2 \in \mathrm{C}(\mathcal{H})$, where $\circ$ denotes composition of channels and "." denotes matrix multiplication. This is easy to see from Eq.(2.25). Consider $\mathcal{E} = \mathcal{E}_1 \circ \mathcal{E}_2$ which implies $\mathcal{E}(\rho) = \mathcal{E}_2(\mathcal{E}_1(\rho))$ for all $\rho \in \mathrm{D}(\mathcal{H})$. Therefore,

$$\begin{aligned}\Gamma(\mathcal{E})\,\mathrm{vec}(\rho) &= \mathrm{vec}\left(\mathcal{E}(\rho)\right)\\ &= \mathrm{vec}\left(\mathcal{E}_2\left(\mathcal{E}_1(\rho)\right)\right)\\ &= \Gamma(\mathcal{E}_2)\,\mathrm{vec}\left(\mathcal{E}_1(\rho)\right)\\ &= \left(\Gamma(\mathcal{E}_2).\Gamma(\mathcal{E}_1)\right)\mathrm{vec}(\rho),\end{aligned} \tag{2.28}$$

where we invoke Eq.(2.25) multiple times in either direction. Since the above equation holds for all $\rho \in \mathrm{D}(\mathcal{H})$, it implies $\Gamma(\mathcal{E}) = \Gamma(\mathcal{E}_1 \circ \mathcal{E}_2) = \Gamma(\mathcal{E}_2).\Gamma(\mathcal{E}_1)$. So, to obtain the Liouville matrix of the composed map, we need to just take a matrix product of the Liouville matrices of the component maps. Instead of using the standard basis for vectorization if one alternatively chooses to use the Pauli matrices as the basis, it leads to the representation called the *Pauli-Liouville representation*. It enjoys the same compositional advantage and is more convenient to deal with in certain applications related to quantum error correction. The recipe to derive the entries of the Pauli-Liouville representation (sometimes referred to as the Pauli Transfer Matrix (PTM) [WBC15]) of a map $\mathcal{E} \in \mathrm{C}(\mathcal{H})$ is given by the following equation:

$$\left[\Gamma^{PTM}(\mathcal{E})\right]_{i,j} = \mathrm{Tr}\left(P_i \mathcal{E}(P_j)\right), \tag{2.29}$$

where $P_i, P_j \in \mathrm{L}(\mathcal{H})$ are Pauli matrices. Sometimes we will drop the superscript PTM in this thesis but the vectorization basis can be inferred from the context.

Although this representation composes naturally, it sheds no light into the complete positivity of a given map. This will be fulfilled by the next representation.

2.2.2 Choi representation

For every quantum channel $\mathcal{E} \in \mathrm{C}(\mathcal{H}_1, \mathcal{H}_2)$, we define a corresponding *Choi* matrix [Cho75] $J(\mathcal{E}) \in \mathrm{L}(\mathcal{H}_2 \otimes \mathcal{H}_1)$ as

$$J(\mathcal{E}) = (\mathcal{E} \otimes \mathcal{I}_{\mathrm{L}(\mathcal{H}_1)})(\mathrm{vec}(\mathbb{I}_{\mathcal{H}_1})\,\mathrm{vec}(\mathbb{I}_{\mathcal{H}_1})^\dagger), \tag{2.30}$$

where $\mathcal{I}_{\mathrm{L}(\mathcal{H}_1)}$ is the identity map and $\mathbb{I}_{\mathcal{H}_1}$ is the identity matrix of appropriate dimension. The Choi matrix can also be expressed as

$$J(\mathcal{E}) = \sum_{a,b \in \{1,2,\ldots,d\}} \mathcal{E}(E^{a,b}) \otimes E^{a,b}, \tag{2.31}$$

where we assume that $\mathcal{H}_1 = \mathcal{H}_2 = \mathbb{C}^d$. Similar to the Liouville representation, the Choi representation of a map is a linear bijection. To recover the action of the noise map $\mathcal{E} \in \mathrm{C}(\mathcal{H}_1, \mathcal{H}_2)$ from its Choi matrix $J(\mathcal{E}) \in \mathrm{L}(\mathcal{H}_2 \otimes \mathcal{H}_1)$, one can use the following equation:

$$\mathcal{E}(\rho) = \mathrm{Tr}_{\mathcal{H}_1}\left(J(\mathcal{E})(\mathbb{I}_{\mathcal{H}_2} \otimes \rho^T)\right). \tag{2.32}$$

The advantage of the Choi representation is that it provides a convenient method to check if a map $\mathcal{E} \in \mathrm{T}(\mathcal{H}_1, \mathcal{H}_2)$ is a quantum channel or not. In other words, it allows us to check if a map is completely positive and trace preserving (CPTP) or not. A given map $\mathcal{E} \in \mathrm{T}(\mathcal{H}_1, \mathcal{H}_2)$ is CPTP if and only if its corresponding $J(\mathcal{E})$ is positive semidefinite i.e., $J(\mathcal{E}) \in \mathrm{Pos}(\mathcal{H}_2 \otimes \mathcal{H}_1)$, and $\mathrm{Tr}_{\mathcal{H}_2}\left(J(\mathcal{E})\right) = \mathbb{I}_{\mathcal{H}_1}$.

2.2.3 Chi-matrix representation

The Chi-matrix $\chi(\mathcal{E})$ of a channel $\mathcal{E} \in \mathrm{T}(\mathcal{H})$ is defined by the following relation:

$$\mathcal{E}(\rho) = \sum_{ij} \chi(\mathcal{E})_{ij} P_i \rho P_j, \tag{2.33}$$

where $\{P_i\}$ are Pauli matrices. It is related to the Choi matrix by a change of basis from the computational basis to the Pauli basis [WBC15]. The conversion is given by the following equation:

$$\chi(\mathcal{E}) = T_{c\to\sigma} J(\mathcal{E}) T^{\dagger}_{c\to\sigma}, \tag{2.34}$$

where $T_{c\to\sigma}$ is the vectorization change of basis operator from standard computational basis denoted by c to Pauli basis denoted by σ i.e.,

$$T_{c\to\sigma} : [\mathrm{vec}(A)]_c \to [\mathrm{vec}(A)]_\sigma, \tag{2.35}$$

for all matrices $A \in L(\mathcal{H} \otimes \mathcal{H})$. The description of the operator is given by

$$T_{c\to\sigma} = \sum_{\alpha} e_\alpha [\mathrm{vec}(\sigma_\alpha)]^{\dagger}_c, \tag{2.36}$$

where $\{\sigma_\alpha\}$ is the Pauli basis.

2.2.4 Kraus representation

In this representation, the noise map $\mathcal{E} \in \mathrm{C}(\mathcal{H})$ is associated to a set of operators $\{A_k\}$ where $A_k \in \mathrm{L}(\mathcal{H})$. The action of the map on the state $\rho \in \mathrm{D}(\mathcal{H})$ is given by

$$\mathcal{E}(\rho) = \sum_k A_k \rho A_k^\dagger. \tag{2.37}$$

As we saw in Eq.(2.6) at the beginning of the background section, this representation pops out naturally when we consider the action of a unitary map on the larger Hilbert space comprised of system and the environment. Unlike Liouville and Choi representations, the set of Kraus operators [KBD+83; NC10] corresponding to a channel are not unique. The trace preserving property of the map implies

$$\begin{aligned}
\mathrm{Tr}(\mathcal{E}(\rho)) &= \mathrm{Tr}(\sum_k A_k \rho A_k^\dagger) \\
&= \sum_k \mathrm{Tr}(A_k \rho A_k^\dagger) \\
&= \sum_k \mathrm{Tr}(A_k^\dagger A_k \rho) && \text{(using cyclic property of trace)} \\
&= \mathrm{Tr}(\sum_k (A_k^\dagger A_k) \rho).
\end{aligned} \tag{2.38}$$

Since the last equation holds for all $\rho \in \mathrm{D}(\mathcal{H})$, it implies that $\sum_k (A_k^\dagger A_k) = \mathbb{I}$.

2.2.5 Converting between representations

It is important to be able to understand the connection between the different representations i.e., Liouville, Choi and the Kraus representation. We omit Chi representation from the discussion in this section as it can be obtained by a change of basis from the Choi matrix. The following proposition [Wat18] ties all these three representations together.

Proposition 1. *Consider a quantum channel $\mathcal{E} \in \mathrm{C}(\mathcal{H})$, where $\mathcal{H} = \mathbb{C}^d$ and a set of operators $\{A_k\}$ where $A_k \in \mathrm{L}(\mathcal{H})$ for all k. The following statements which correspond to the three different representations are equivalent.*

1. *(Liouville representation) The Liouville representation of $\mathcal{E}$ is expressed as:*

$$\Gamma(\mathcal{E}) = \sum_k A_k \otimes \overline{A_k}. \tag{2.39}$$

2. *(Choi representation) The Choi matrix is given by*

$$J(\mathcal{E}) = \sum_k \mathrm{vec}(A_k)\,\mathrm{vec}(A_k)^\dagger. \tag{2.40}$$

3. *(Kraus representation) The following statement is true:*

$$\mathcal{E}(\rho) = \sum_k A_k \rho A_k^\dagger \tag{2.41}$$

for all $\rho \in \mathrm{D}(\mathcal{H})$.

Proof. The equivalence between statements 1 and 3 is established by taking vec and applying Roth's lemma i.e.,

$$\begin{aligned} \mathrm{vec}\left(\mathcal{E}(\rho)\right) &= \Gamma(\mathcal{E})\rho \\ &= \sum_k \mathrm{vec}(A_k \rho A_k^\dagger) \\ &= \sum_k (A_k \otimes \overline{A_k})\,\mathrm{vec}(\rho). \end{aligned} \tag{2.42}$$

Since this set of equations holds for all $\rho \in \mathrm{D}(\mathcal{H})$, it implies statement 1 $\iff$ statement 3. The equivalence between statements 2 and 3 is established using

$$\mathrm{vec}(A_k) = (A_k \otimes \mathbb{I}_{\mathcal{H}})\,\mathrm{vec}(\mathbb{I}_{\mathcal{H}}). \tag{2.43}$$

□

2.2.6 Some important quantum channels

In this section, we will describe some commonly occurring noise phenomena in quantum devices and their corresponding representations.

Unitary channel

Quantum circuits are composed of quantum gates which are unitary operators. In addition, unitary noise is commonly observed in quantum systems and arises as a consequence of imperfect control sequences. Recall from Eq.(2.1) that the Hamiltonian according to which a system is evolved is a function of control parameters. When these parameters are applied imperfectly in practice, the resulting unitary is different from the ideal unitary. This leads to unitary noise commonly called as overrotation errors. The action of a unitary channel $\mathcal{E} \in \mathrm{C}(\mathcal{H})$ is given by

$$\mathcal{E}(\rho) = U\rho U^{\dagger}, \tag{2.44}$$

where $U \in \mathrm{L}(\mathcal{H})$ is a unitary operator. The simplest example of unitary noise is (over or under) rotation around the Pauli $Z-$axis by a small angle expressed as:

$$\mathcal{E}(\rho) = R_Z(\theta)\rho R_Z(\theta)^{\dagger}, \tag{2.45}$$

where $R_Z(\theta) = e^{-i\frac{\theta}{2}Z}$.

Notice that unitary channels have only one Kraus operator namely $A_1 = U$. It is also easy to see that $\sum_k A_k^{\dagger}A_k = U^{\dagger}U = \mathbb{I}$. The other representations can be easily derived using Proposition 1. We mention some of them here for easy readability. The Liouville representation of a unitary operation $\mathcal{U}$ is given by

$$\Gamma(\mathcal{E}) = U \otimes \overline{U}. \tag{2.46}$$

The corresponding Choi matrix is given by

$$J(\mathcal{E}) = \mathrm{vec}(U)\,\mathrm{vec}(U)^{\dagger}. \tag{2.47}$$

Pauli channel

The action of a generalized Pauli channel $\mathcal{E} \in \mathrm{C}(\mathcal{H})$ on a state $\rho \in \mathrm{D}(\mathcal{H})$ is given by

$$\mathcal{E}(\rho) = \sum_{\alpha=1}^{d^2} p_\alpha P_\alpha \rho P_\alpha, \tag{2.48}$$

where $p_\alpha \in \mathbb{R}$ is the probability of the state being acted on by the Pauli operator P_α. Being probabilities, the coefficients obey $p_\alpha \geq 0 \;\forall\; \alpha$ and $\sum_\alpha p_\alpha = 1$. Note that the Pauli operators for qubit systems are self-adjoint i.e., $P_\alpha^{\dagger} = P_\alpha$ and they square to identity i.e., $P_\alpha^{\dagger}P = P_\alpha^2 = \mathbb{I}$. For a multi-qubit Pauli operator P, we will denote the number of non-identity single qubit operators in it as the *Hamming weight* of P and denote it by $|P|$.

The Kraus operators for the Pauli channel are given by $A_\alpha = \sqrt{p_\alpha}P_\alpha$. The trace preserving property can be verified as follows:

$$\begin{aligned} \sum_\alpha A_\alpha^{\dagger}A_\alpha &= \sum_\alpha p_\alpha P_\alpha^{\dagger}P_\alpha \\ &= \sum_\alpha p_\alpha \mathbb{I} && (P_\alpha \text{ is a unitary}) \\ &= \mathbb{I}. && (2.49) \end{aligned}$$

By comparing equations (2.33) and (2.48), we conclude that the Chi-matrix elements for $\mathcal{E}$ are given by

$$\chi_{\alpha\beta} = \delta_{\alpha\beta} p_\alpha, \tag{2.50}$$

where $\delta_{\alpha\beta} = 1$ if $\alpha = \beta$ and $\delta_{\alpha\beta} = 0$ otherwise.

A common example of the Pauli channel is the *depolarizing channel*. The probabilities of all Pauli errors are set to be identical in the depolarizing channel except the Identity (assumed to correspond to $\alpha = 1$). Suppose we set $p_\alpha = p_d/d^2$ for all $\alpha \neq 1$. The normalization condition $\sum_\alpha p_\alpha = 1$ gives $p_1 = 1 - p_d + p_d/d^2$. The action of the depolarizing channel on a state is then given by:

$$\mathcal{E}(\rho) = (1 - p_d)\rho + \frac{p_d}{d}\mathbb{I}. \tag{2.51}$$

2.3 Noise metrics

In this section, we will discuss the popular error metrics used to compare the CPTP maps and understand the strength of the associated noise. Before describing distance between noise maps, it is helpful to understand ways to measure the distance between quantum states.

2.3.1 Distance between quantum states

Given two quantum states $\rho, \sigma \in \mathrm{D}(\mathcal{H})$, the *fidelity* between them is defined as

$$F(\rho, \sigma) = \| \sqrt{\rho}\sqrt{\sigma} \|_1, \tag{2.52}$$

where $\sqrt{\rho}$ and $\sqrt{\sigma}$ are the unique positive semi-definite operators that satisfy $\sqrt{\rho}\sqrt{\rho} = \rho$ and $\sqrt{\sigma}\sqrt{\sigma} = \sigma$ respectively. Fidelity can be thought of as the amount of overlap between the two states. Note that fidelity is not a metric in the strict mathematical sense. However, we will use this terminology in the first three chapters of this thesis. Expanding the above expression leads to the following alternative expression:

$$F(\rho, \sigma) = \mathrm{Tr}(\sqrt{\sqrt{\sigma}\rho\sqrt{\sigma}}). \tag{2.53}$$

We will state some key properties about the fidelity function without proving them here. For proofs, please refer to Ref. [Wat18]. For density operators $\rho, \sigma \in \mathrm{D}(\mathcal{H})$, the fidelity obeys the following properties:

1. It is symmetric in the arguments i.e., $F(\rho, \sigma) = F(\sigma, \rho)$.
2. $F(\rho, \sigma) \geq 0$ with equality if and only if $\rho\sigma = \mathbf{0}$.
3. $F(\rho, \sigma) \leq 1$ with equality if and only if $\rho = \sigma$.

The *trace distance* between two quantum states $\rho, \sigma \in \mathrm{D}(\mathcal{H})$ is defined as

$$D(\rho, \sigma) = \frac{1}{2}\|\rho - \sigma\|_1. \tag{2.54}$$

For pure states, $\rho = |\psi\rangle\langle\psi|$ and $\sigma = |\phi\rangle\langle\phi|$, the trace distance evaluates to

$$D(|\psi\rangle\langle\psi|, |\phi\rangle\langle\phi|) = \sqrt{1 - |\langle\phi, \psi\rangle|^2}. \tag{2.55}$$

These two measures satisfy the following Fuchs-van de Graaf inequalities [FG99]

$$1 - F(\rho, \sigma) \leq D(\rho, \sigma) \leq \sqrt{1 - F(\rho, \sigma)^2}. \tag{2.56}$$

2.3.2 Distance between quantum channels

There are several inequivalent ways to quantify the strength of noise. Of these, two are used widely to study fault tolerance - fidelity and diamond distance. A natural way to define fidelity between two given quantum channels $\mathcal{E}_1, \mathcal{E}_2 \in \mathrm{C}(\mathcal{H})$ is to take the fidelity of the output states when they are supplied the same input state ρ i.e.,

$$F(\mathcal{E}_1(\rho), \mathcal{E}_2(\rho)). \tag{2.57}$$

It is helpful to take a step back and think about situations when we are likely to care about distance or overlap between channels in experimental settings. When experimentalists implement quantum gates (or unitaries) in labs, they are not perfect. We can therefore use these metrics to quantify how close or far these imperfect implementations are to the ideal unitaries. Consider $\mathcal{E}_1(\rho) = \mathcal{U}(\rho) = U\rho U^\dagger$ to be the ideal unitary, and $\mathcal{E}_2(\rho) = \mathcal{E}_U(\rho)$ to be the noisy implementation of U. Given a pure state $\rho = |\psi\rangle\langle\psi|$, the fidelity between the ideal and noisy implementation of $\mathcal{U}$ is given by

$$F(\mathcal{E}_U(\rho), \mathcal{U}(\rho)) = \langle\psi|U^\dagger \mathcal{E}_U(|\psi\rangle\langle\psi|)U|\psi\rangle. \tag{2.58}$$

The above measure is sometimes referred to as the *gate fidelity* in the context where $\mathcal{U}$ is an implementation of a gate in a quantum circuit. The *average gate fidelity* is correspondingly defined by the average taken over all pure states i.e.,

$$\langle F(\mathcal{E}_U,\mathcal{U})\rangle = \int d\mu(\psi) F(\mathcal{E}_U(|\psi\rangle\langle\psi|),\mathcal{U}(|\psi\rangle\langle\psi|)) = \int d\mu_{FS}(\psi)\langle\psi|U^{\dagger}\mathcal{E}_U(|\psi\rangle\langle\psi|)U|\psi\rangle, \tag{2.59}$$

where $d\mu_{FS}(\psi)$ is the Fubini-Study measure which is a unitarily invariant measure on the complex projective space. We refer the readers to Ref. [Wat18] for more details on unitarily invariant measures. Sometimes we extract just the noisy part of the imperfect implementation by assuming the imperfect implementation of gates to be a composition of the perfect unitary followed by the noise channel. We want to determine the strength of the noise by comparing the noisy part of the imperfect implementation to the identity process i.e., we set the first noise process to be the identity map $\mathcal{E}_1(\rho) = \mathcal{I}(\rho) = \rho$, and we determine its overlap with the second map which represents only the noisy part of the imperfect implementation i.e., $\mathcal{E}_2(\rho) = \mathcal{E}(\rho)$. The average fidelity of the noise is then referred to as $\langle F(\mathcal{E})\rangle$ and is expressed as

$$\langle F(\mathcal{E})\rangle = \langle F\left(\mathcal{E}(\rho),\mathcal{I}(\rho)\right)\rangle = \int d\mu_{FS}(\psi)\langle\psi|\mathcal{E}(|\psi\rangle\langle\psi|)|\psi\rangle. \tag{2.60}$$

Finally, the *average gate infidelity*: $r(\mathcal{E})$ [Nie96; Sch96; Rag01] is defined by

$$r(\mathcal{E}) = 1 - \langle F(\mathcal{E})\rangle = 1 - \int d\mu_{FS}(\psi)\langle\psi|\mathcal{E}(|\psi\rangle\langle\psi|)|\psi\rangle\ . \tag{2.61}$$

The *diamond distance* [Kit97a; Kit97b; Wat09; KSV02; Gut12] between two quantum channels $\mathcal{E}_1$ and $\mathcal{E}_2$ is defined as

$$\|\mathcal{E}_1 - \mathcal{E}_2\|_{\diamondsuit} = \|\mathcal{E}_1 \otimes \mathcal{I}_{\mathrm{L}(\mathcal{H})} - \mathcal{E}_2 \otimes \mathcal{I}_{\mathrm{L}(\mathcal{H})}\|_1, \tag{2.62}$$

where

$$\|\mathcal{E}_1 - \mathcal{E}_2\|_1 := \max_{\rho\in\mathrm{D}(\mathcal{H})} ||\mathcal{E}_1(\rho) - \mathcal{E}_2(\rho)||_1. \tag{2.63}$$

In most cases we will be interested in the diamond distance of a noise process $\mathcal{E}$ to the identity channel expressed as follows:

$$||\mathcal{E} - \mathcal{I}||_{\diamondsuit} = \max_{\rho\in\mathrm{D}(\mathcal{H})} ||(\mathcal{E}\otimes\mathcal{I})\rho - \rho||_1\ . \tag{2.64}$$

The average gate infidelity in Eq.(2.61) can be efficiently estimated using experimental techniques such as randomized benchmarking [EAŻ05; KLR+08; MGE11; MGE12]. The diamond distance satisfies mathematical properties that are needed to demonstrate fault tolerance proofs [ABO08; SDT07; AP09]. The following bounds relate these two quantities [WF14]

$$\frac{d+1}{d} r(\mathcal{E}) \leq \frac{1}{2} ||\mathcal{E} - \mathcal{I}||_{\diamondsuit} \leq \sqrt{d(d+1)}\sqrt{r(\mathcal{E})} \tag{2.65}$$

where $\mathcal{E} \in \mathrm{C}(\mathcal{H})$ is assumed to act on a d-dimensional system. The lower bound is saturated by Pauli channels whereas the upper bound is saturated for unitary errors. The diamond distance cannot be estimated using experimental techniques; it can only be loosely bounded by the above expression using values of r obtained from randomized benchmarking. For example, consider a practically relevant error rate of $r = 10^{-4}$, and one can see that $||\mathcal{E} - \mathcal{I}||_{\diamondsuit}$ can span several orders of magnitude from 10^{-4} to 10^{-2}. Therefore, these bounds are not very useful for the purpose of estimating the relevant numbers required to assess whether a system meets the requirements for fault tolerance. We highlight and address this challenge of having a metric that satisfies both desirable properties - being experimentally measurable and being relevant for quantum error correction in chapter 3.

2.4 Noise characterization and tailoring

In this section, we will discuss some key noise characterization protocols which are relevant to the methods proposed later in this thesis.

2.4.1 t-designs and twirling

Similar to the requirement for sampling random quantum states to calculate average gate infidelity, there often is a requirement to sample unitaries uniformly from the unitary group on finite-dimensional Hilbert spaces. Let $\mathcal{U}_{\mathcal{H}}$ denote the group of all unitaries acting on quantum states in $\mathrm{D}(\mathcal{H})$. A unitary t-design is defined to be a finite set of unitary operators $\{U_i\}, 1 \leq i \leq K$ where $U_i \in \mathrm{L}(\mathcal{H})$ such that for every homogeneous polynomial $P_{t,t}$ of degree at most t in the matrix elements of a unitary operator $U \in \mathrm{L}(\mathcal{H})$ and at most t in their complex conjugates, the following condition holds true:

$$\frac{1}{K}\sum_{i=1}^{K} P(U_i) = \int_{\mathcal{U}_{\mathcal{H}}} P_{t,t}(U) d\mu_{Haar}(U), \tag{2.66}$$

where $d\mu_{Haar}(U)$ is the unitarily invariant Haar measure. We refer the reader to Ref. [Wat18] for more details on the Haar measure. The polynomial $P(U)$ is an explicit polynomial constructed out of entries of the matrix U and their complex conjugates. Essentially, t-designs provide an efficient way to evaluate integrals over polynomials involving unitaries. In practice, to evaluate any such polynomial one can use the finite elements from the t-design and evaluate the integral exactly using the above equation.

In particular, we are interested in the case where $t = 2$. Consider a quantum channel $\mathcal{E} \in \mathrm{C}(\mathcal{H})$ acting on a quantum state $\rho \in \mathrm{D}(\mathcal{H})$. Suppose that $\mathcal{E}$ is conjugated by a random unitary operation which is chosen according to a measure μ on $\mathcal{U}_{\mathcal{H}}$. The composite operator is of the form $\mathcal{U}_1 \circ \mathcal{E} \circ \mathcal{U}_2$ where

$$\mathcal{U}_1(\rho) = U\rho U^\dagger, \text{ and} \tag{2.67}$$

$$\mathcal{U}_2(\rho) = U^\dagger \rho U. \tag{2.68}$$

The resulting operator on average is denoted by $\mathcal{T}_\mu(\mathcal{E}) = \mathbb{E}_\mu(\mathcal{E})$, where

$$\mathcal{T}_\mu(\mathcal{E})(\rho) = \int_{U \in \mathcal{U}_{\mathcal{H}}} U^\dagger \mathcal{E}(U\rho U^\dagger) U \, d\mu(U). \tag{2.69}$$

The transformation of $\mathcal{E}$ to $\mathcal{T}_\mu(\mathcal{E})$ is called *twirling*. Of particular interest is the twirl corresponding to the Haar measure μ_{Haar}. In this case, sampling from the Haar measure is equivalent to sampling uniformly at random from the elements of 2-*design*. In other words, the condition in Eq.(2.66) takes the following equivalent form:

$$\frac{1}{K} \sum_{i=1}^{K} U_i^\dagger \mathcal{E}(U_i \rho U_i^\dagger) U_i = \int_{U \in \mathcal{U}_{\mathcal{H}}} U^\dagger \mathcal{E}(U\rho U^\dagger) U \, d\mu_{Haar}(U), \tag{2.70}$$

where $\{U_1, U_2, \ldots, U_K\}$ forms a 2-design.

The *Clifford* group on *n-qubits* is defined to be a group of unitaries that normalize the Pauli group:

$$\mathrm{Clif}_n = \{V \in \mathcal{U}_{\mathcal{H}} | V\mathcal{P}_n V^\dagger = \mathcal{P}_n\}, \tag{2.71}$$

where $\mathcal{P}_n$ is the n-qubit Pauli group. In simple terms, the elements of the Clifford group map Paulis to Paulis. The Clifford group plays a central role in quantum computation with applications to noise estimation, quantum error correction, simulation of quantum systems, etc. In particular, a uniform distribution over Clif_n forms a 2-design. Therefore, we can use the elements of the Clifford group in Eq.(2.70) i.e.,

$$\frac{1}{|\,\mathrm{Clif}_n\,|} \sum_{i=1}^{|\,\mathrm{Clif}_n\,|} C_i^\dagger \mathcal{E}(C_i \rho C_i^\dagger) C_i = \int_{U \in \mathcal{U}_{\mathcal{H}}} U^\dagger \mathcal{E}(U\rho U^\dagger) U \, d\mu_{Haar}(U), \tag{2.72}$$

where $\text{Clif}_n = \{C_1, C_2, \ldots, C_{|\text{Clif}_n|}\}$ is the n-qubit Clifford group. It can be shown [EAŻ05; Nie02] that the channel $\mathcal{E}^{\mathcal{T}}_{\mu_{Haar}}$, which can be obtained via twirling using the Clifford group, produces a depolarizing channel $\mathcal{E}_{dep} \in \mathrm{C}(\mathcal{H})$ with the same average fidelity as $\mathcal{E}$. Therefore, the average fidelity of the channel $\mathcal{E}$ in Eq.(2.60) is given by

$$\langle F(\mathcal{E}) \rangle = p + \frac{1-p}{d}, \tag{2.73}$$

where the parameter p comes from the depolarizing channel $\mathcal{E}_{dep}$ defined by:

$$\mathcal{E}_{dep}(\rho) = p\rho + (1-p)\frac{\mathbb{I}}{d}. \tag{2.74}$$

The depolarizing channel was introduced in section 2.2.6. In summary, twirling any quantum channel with the Clifford group produces a depolarizing channel with the same fidelity as the channel being twirled. In some applications of twirling such as randomized benchmarking, we will be interested in a sequence of twirls of length m say. In this case, the twirled channel can be expressed as the m-fold composition of $\mathcal{E}_{dep}$ with itself i.e.,

$$\mathcal{E}^m_{dep}(\rho) = p^m\rho + (1-p^m)\frac{\mathbb{I}}{d}. \tag{2.75}$$

The resulting average fidelity of the depolarizing channel and equivalently the m-fold twirled channel is then given by:

$$\langle F(\mathcal{E}^m_{dep}) \rangle = p^m + \frac{1-p^m}{d}. \tag{2.76}$$

Finally, we want to mention that twirling using the Pauli group $\mathcal{P}_n$ produces a Pauli channel [Mag08; DCE+09]. The effective twirled Pauli channel is given by:

$$\mathcal{T}_{\mathcal{P}_n}(\mathcal{E})(\rho) = \sum_{P \in \mathcal{P}_n} P\mathcal{E}(P\rho P)P\,. \tag{2.77}$$

We will use the notation $\mathcal{E}^T$ in later chapters to denote the Pauli Twirl of the CPTP map $\mathcal{E}$.

2.4.2 Randomized benchmarking

The traditional way of benchmarking quantum noise processes includes standard process tomography [CN97; PCZ97; WHE+04], ancilla/entanglement assisted process tomography [ABJ+03; DLP01] and Monte Carlo methods. While quantum process tomography is able to fully characterize an unknown noise process occurring in the device, it is not scalable in the number of qubits n i.e., the number of experiments to be executed and the corresponding post-processing time both grow exponentially with n. Often we are interested in analyzing errors coming from the implementations of gates separate from the ones from state preparation and measurements. Quantum process tomography and Monte-Carlo methods assume negligible state preparation and measurement errors. This is a strong assumption and hence these methods are not robust to state preparation and measurement (SPAM) errors. Randomized benchmarking (RB) and its variants are scalable as well as robust to SPAM errors. They help calculate quantities such as the average gate infidelity of a gateset [MGE11], infidelity of a particular gate of interest [MGJ+12], coherence in the noise [WGH+15], leakage in the system [WG18], etc.

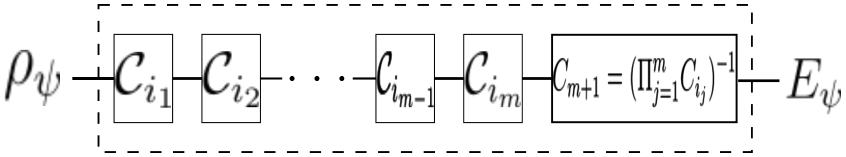


Figure 2.1: The figure shows the sequence of Clifford gates applied in a standard RB experiment. The last gate is chosen to be the inverse of the composition of the first m gates so that the entire sequence is an identity operation in the absence of noise.

In this section, we will outline the standard RB protocol [MGE11] which estimates the average gate infidelity of a given gate set. The protocol involves the following steps:

1. Generate a sequence of $(m+1)$ quantum operations where the first m operations are chosen uniformly at random from a group $\mathcal{G} \subset \mathcal{U}_{\mathcal{H}}$. The $(m+1)^{th}$ operation is chosen to be the inverse of the composition of first m operations as shown in Fig. 2.1. In other words, the ideal (error free) composition of these operations is the identity operation. We will choose the group $\mathcal{G}$ to be the Clifford group on n-qubits i.e., Clif_n defined in Eq.(2.71). It is known that polynomial size Clifford circuits can be simulated efficiently on a classical computer [Got97]. Therefore, it is easy to pre-compute the $(m+1)^{th}$ operation for any set of m Clifford operations. When implemented on a device, these Clifford operations will have errors associated to them. We imagine a noise channel $\mathcal{E}_{i_j} \in \mathrm{C}(\mathcal{H})$ associated to each Clifford operation $\mathcal{C}_{i_j}$, where $\mathcal{C}_{i_j}(\rho) = C_{i_j} \rho C_{i_j}^\dagger$. The sequence of operations is represented by:
$$S_{\vec{i_m}} = \bigcirc_{j=1}^{m+1} (\mathcal{E}_{i_j} \circ \mathcal{C}_{i_j}), \tag{2.78}$$
where $\vec{i_m}$ denotes the tuple $(i_1, i_2, \ldots, i_m)$ and signifies the sequence of random Clifford operations. For simplicity, we will assume gate-independent noise in this

section i.e., $\mathcal{E}_{i_j}$ to be the same for all i, j. However, a similar protocol and analysis holds for the gate dependent case [Wal18].

2. For each sequence $S_{\vec{i_m}}$ we measure the survival probability of the initial state ρ_ψ when it is evolved through the sequence i.e., we measure $\mathrm{Tr}(E_\psi S_{\vec{i_m}}(\rho_\psi))$. Here, ρ_ψ captures state preparation errors and E_ψ is the POVM element that takes into account measurement errors. In the absence of noise, $\rho_\psi = E_\psi = |\psi\rangle\langle\psi|$.

3. Calculate the average sequence fidelity defined by:

$$F_{seq}(m, \psi) = \mathrm{Tr}(E_\psi S_m(\rho_\psi)), \tag{2.79}$$

where S_m is the average sequence operation taken over several random sequences:

$$S_m = \frac{1}{|\vec{i_m}|} \sum_{\vec{i_m}} S_{\vec{i_m}}. \tag{2.80}$$

4. Fit the results obtained for the averaged $F_{seq}(m, \psi)$ into the following equation:

$$F_{seq}(m, \psi) = Ap^m + B. \tag{2.81}$$

The parameters A, B capture the state preparation and measurement errors. The value of parameter p obtained can be plugged into Eq.(2.73) to get the average gate fidelity $\langle F(\mathcal{E}) \rangle$. The average gate infidelity $r(\mathcal{E})$ is then given by:

$$r(\mathcal{E}) = 1 - \langle F(\mathcal{E}) \rangle = 1 - p - \frac{(1-p)}{d}. \tag{2.82}$$

For the derivation of Eq.(2.81), we refer the reader to Ref. [MGE11].

2.4.3 Randomized compiling

Randomized Compiling (RC) [WE16] is a noise-tailoring technique that transforms coherent errors into stochastic errors with little to no overhead. Randomized compiling imagines a given quantum circuit as a sequence of layers called cycles. Furthermore, it classifies them into *easy* and *hard* cycles based on the expected noise level for a given cycle, with cycles that are expected to have low error rates being called "easy" and the remaining cycles called "hard". For instance, the easy cycles can comprise of all single qubit gates and the hard cycles are composed of only entangling gates. Every quantum circuit can be broken into a collection of alternating easy and hard cycles.

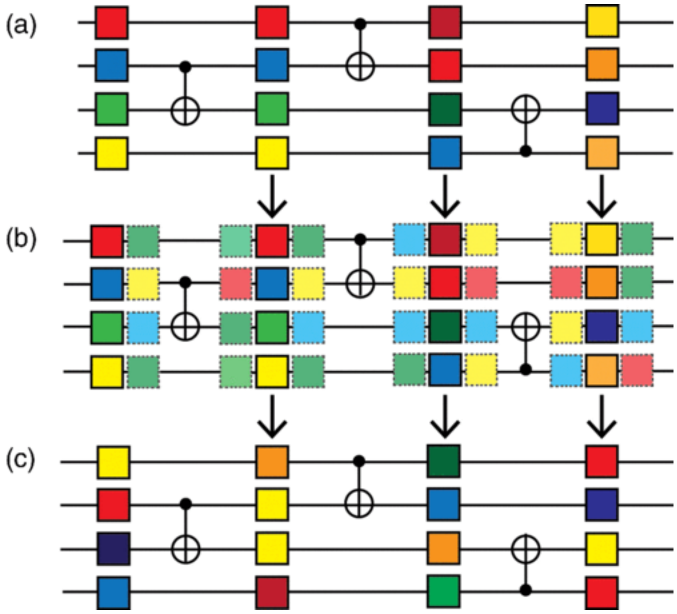


Figure 2.2: Randomized Compiling high level - this is figure 1 in Ref. [WE16]. The top figure shows a bare circuit with alternating easy and hard cycles. The middle figure shows insertion of random Pauli gates in between easy and hard cycles. The bottom figure shows that the extra randomization gates are compiled into the existing gates resulting in a random compilation of the bare circuit.

At a high level, the basic idea is to insert Pauli randomizing gates (twirling gates) around the hard cycles of a target circuit. To ensure that the circuit depth remains the same, these gates are compiled into the existing ones. Moreover, it is ensured that the circuit remains logically[1] equivalent i.e., the computation being implemented is unchanged by the insertion of random gates. Fig. 2.2 illustrates the key steps of randomized compiling. This procedure is repeated for many different compilations and the results are averaged at the end.

Now, we will describe the details of how the above technique tailors the coherent parts of the noise to stochastic noise. Let $C_{j,k}$[2] denote the easy gate acting on qubit j in the k^{th} clock cycle, G_k denote the hard gate in the same clock cycle, $\mathcal{E}_e$ denote the gate-independent noise on the easy gates, $\mathcal{E}(G_k)$ denote the gate-dependent noise on

[1] Note that although we use the term logical, there is no error correction involved here. We are just referring to the net unitary operation implemented by the given quantum circuit.

[2] Note that this notation is unrelated to the Clifford group defined previously. The scope of this is only within the randomized compiling subsection.

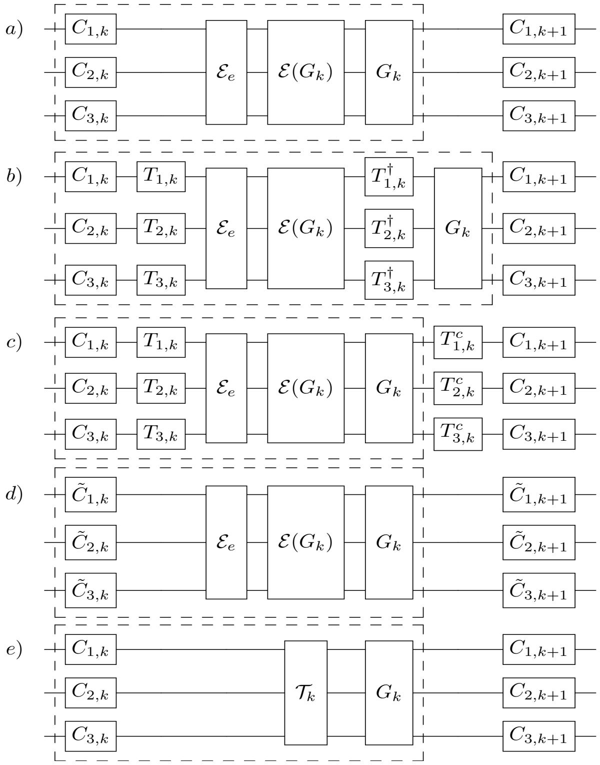


Figure 2.3: Randomized Compiling at the clock cycle level - this is figure 2 in Ref. [WE16]. (a) A snapshot of the k^{th} clock cycle. (b) Twirling gates are inserted in this step. (c) Twirling gates are commuted through the hard cycle. (d) Twirling gates and correction gates are compiled into adjacent easy gates. (e) The tailored circuit which contains the average noise $\mathcal{T}_k$ seen by the circuits over many randomization as defined in Eq.(2.84).

the hard gate G_k and $\mathbf{T}$ denote the twirling set. Consider a snapshot of the k^{th} cycle of a given quantum circuit as depicted in Fig. 2.3(a). We replace each round of easy gates $\vec{C}_k$ (comprising of all easy gates in the k^{th} clock cycle) with a round of dressed gates i.e.,

$$\vec{C}_k \rightarrow \tilde{C}_k = \vec{T}_k \vec{C}_k \vec{T}^c_{k-1}, \tag{2.83}$$

where the twirling gate acting on the j^{th} qubit in the k^{th} clock cycle $T_{j,k}$ is chosen uniformly at random from the twirling set $\mathbf{T}$. The gates $\vec{T}^c_{k-1} = G_k \vec{T}_k G^\dagger_k$ in Fig. 2.3(c) undo the effect of randomization from the previous cycle. Therefore, in the absence of noise, the circuits remain logically equivalent. Finally, to ensure that the dressed gates can be compiled into the circuit, we require that $\vec{T}^c_k$ is an easy gate for all choices of twirling gates $T_{k,j}$ and hard gates G_k. One choice of a division that ensures this is $\mathbf{T} = \mathcal{P}_1$ (twirling gates are all single qubit Pauli gates); easy gates come from the group generated by the phase gate $R = |0\rangle\langle 0| + i|1\rangle\langle 1|$ and $\mathcal{P}_1$; and the hard gates are either the Hadamard gate, $\sqrt{R}$ or the two-qubit controlled-Z gate. Note that this division ensures that the combination of hard and easy gates form a universal gate set. Finally, notice in Fig. 2.3 that uniformly averaging over the twirling group tailors the noise in the k^{th} cycle to

$$\mathcal{T}_k = \mathbb{E}_{\vec{T}} \vec{T}^\dagger \mathcal{E}(G_k) \mathcal{E} \vec{T}, \tag{2.84}$$

which is a Pauli channel for any choice of $\mathbf{T}$ that is a unitary 1-design. Note that $\mathbf{T} = \mathcal{P}_1$ has this property. Since coherent errors accumulate faster than stochastic errors, this technique helps mitigate the effect of noise on the output of the circuit. Randomized compiling has been applied in several mitigation [HNM+21; FHV+22] and validation [FKD19] works. Recently, randomized compiling was also used to enforce simpler noise models for subsystem measurements [BW23]. Later, in chapters 3 and 4, we will show how we can use randomized compiling in the context of quantum error correction for better diagnostics and performance.

2.4.4 Cycle error reconstruction

Cycle Error Reconstruction (CER) [CDDH+23] is a noise characterization tool that goes beyond standard randomized benchmarking while borrowing a lot of its nice features like robustness to SPAM for instance. It aims to characterize the noise of cycles which are also referred to as n-qubit gates in the tomography literature. Cycles in this context should be thought of as the repeating instruction blocks in any quantum algorithm that are more complicated than individual gates and less complicated than the entire circuit. For example, a cycle for a 5-qubit system can be $\{(0,1) : CZ, (2) : H, (3,4) : CZ\}$ where the tuples specify which qubit(s) the corresponding gate acts on. In this example, Controlled-Z is applied on two pairs of qubits and a Hadamard gate is applied on one qubit. It is important to characterize complete cycles rather than gates in isolation to capture all the cross talk effects that can occur while implementing gates in parallel. Typically, cycles are classified into easy and hard cycles in a spirit similar to randomized compiling. Hard cycles will contain a (fixed) entangling gate (say Controlled-X, Controlled-Z, etc.) with the possibility of them being applied on disjoint pairs of qubits simultaneously. The easy cycles contain (relatively) higher fidelity single qubit gates.

As we noticed before, when circuits are implemented with randomized compiling, the effective noise is stochastic, i.e., a Pauli channel. CER provides a way to estimate

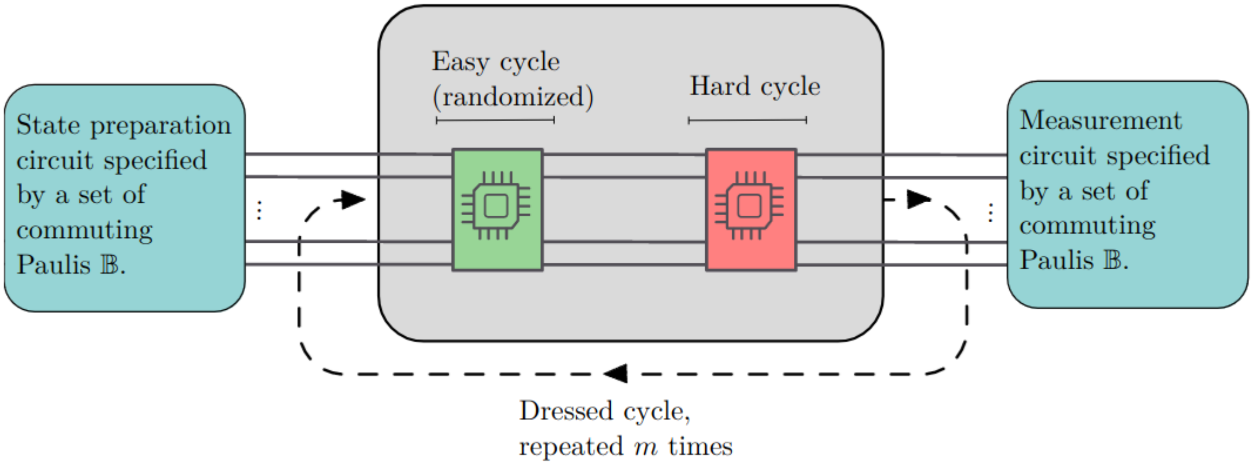

State preparation circuit specified by a set of commuting Paulis $\mathbb{B}$.
Easy cycle (randomized)
Hard cycle
Measurement circuit specified by a set of commuting Paulis $\mathbb{B}$.
Dressed cycle, repeated m times

Figure 2.4: Cycle error reconstruction circuit - this is figure 1 in Ref. [CDDH+23]. The above figure describes the structure of circuits used in both Cycle Benchmarking and Cycle Error Reconstruction. They include repetitions of pairs of easy and hard cycle (collectively called a dressed cycle) sandwiched between state preparation and measurement steps. These cycles when averaged over different randomization are called *effective dressed cycles* whose error profile will be described by a stochastic map (Pauli channel).

the Pauli error probabilities associated to this channel. It builds on a method called *cycle benchmarking* (CB) [EWP+19] which provides a way to obtain the total error probability in a cycle. In other words, while CB only estimates the probability of the *identity* error (or no error), CER provides methods to estimate other dominating Pauli error probabilities in the cycle. The usefulness of these methods really shine when the depolarizing assumption doesn't hold and characterizing one qubit at a time is insufficient due to presence of cross talk. Accounting for cross-talk with very few assumptions makes these techniques readily useful. Other methods similar to CER which perform cycle centric characterization include Average Circuit Eigenvalue Sampling (ACES) [Fla21] and Gate-Set Tomography (GST) [MGS+13; NGR+21]. Since CER type protocols rely on implementing randomized compiling under the hood, it is more useful to characterize the noise on average. In other words, they aim to characterize the noise on *effective cycles*. GST on the other hand characterizes deterministic cycles. A major differentiation between ACES and CER is the precision and the sample overhead. ACES has additive-precision and requires $O(^1/_{\epsilon^2})$ runs to estimate an error rate ϵ whereas CB and CER offer multiplicative precision and require only $O(^1/_{polylog(\epsilon)})$ runs. This makes a huge difference given the current error rates are close to 10^{-4} for which ACES will require about 10^8 runs whereas CER will only need a handful runs. It will make even larger difference when in future the devices attain extremely low error rates in the order of 10^{-8}.

In what follows, we will give a high level overview of CER and we refer the reader to Ref. [CDDH+23] for details. The general structure of circuits that are implemented in both CB and CER are identical and is depicted in Figure 2.4. They contain repetitions of a combination of easy and hard cycle, referred to as the *dressed cycle* sandwiched between appropriate state preparation and measurement steps. The repetition idea has the same impact as standard RB i.e., it amplifies the noise coming from the dressed cycle. The key difference is that in each repetition, similar to randomized compiling, the easy cycle component of the dressed cycles is randomized. The average of each cycle over several randomization is called the *effective dressed cycle*. These effective cycles, as a consequence of the twirling impact of randomized compiling, have a stochastic error profile described by a Pauli channel. Let C_i and G_i be the i^{th} easy and hard cycles respectively. The corresponding effective dressed cycle averaged over all different compilations when twirled using the Pauli group takes the form [CDDH+23]

$$\nu_{\text{drs}}^{\text{eff}}(G_i, C_i) = \phi(G_i C_i)\mathcal{E}_i, \tag{2.85}$$

where $\phi(G_i C_i)$ and $\nu_{\text{drs}}^{\text{eff}}(G_i, C_i)$ are used to denote the ideal and noisy implementations of the i^{th} dressed cycle respectively, and $\mathcal{E}_i$ is a Pauli stochastic channel of the form

$$\mathcal{E}_i(\rho) = \sum_{P \in \mathcal{P}_n} p_i(P) P \rho P^{\dagger} \tag{2.86}$$

for some probability distribution $\{p_i\}$. The above statement is true subject to the condition that at least one of the following statements hold:

1. The cycles G_i and C_i are Cliffords.
2. The easy cycles have a fixed error i.e., $\nu(C_i) = \mathcal{E} \circ \phi(C_i)\ \forall i$, where $\phi(C_i)$ and $\nu(C_i)$ denote the ideal and noisy implementations of C_i respectively.

The second condition is more commonly known as the gate-independent noise assumption in the literature and is generally (approximately) true for cycles composed of high fidelity single qubit gates. It is possible to relax these assumption with more work and the procedure is described in Ref. [CDDH+23]. CER provides a way to estimate the probabilities in the distribution $\{p_i\}$. Since there are exponentially many of them, it makes sense to obtain a subset of them efficiently. Most of the times the choice of this subset is guided by locality constraints in the device. For instance, one can ask about the probability of all two-qubit nearest neighbour Pauli errors. Alternatively, one can ask about the probability of errors with Hamming weight (number of non-identity Paulis) at most w, where w is a constant (say 3). CER provides a way to estimate the marginal probability distribution of any subset of Paulis.

The subroutine used to characterize effectively dressed cycles is detailed in [FW20]. It will be treated here as an oracle and will be referred to as Pauli Infidelity Estimation (PIE). The oracle's action on a subset of Paulis $S \subseteq \mathcal{P}_n$ and a hard cycle G is given by

$$PIE(S, G) = \left\{ f(P^{\text{orb}(G)}) : P \in S) \right\}, \tag{2.87}$$

where

$$\begin{aligned} P^{\text{orb}(G)} &:= \left\{ G^j P G^{-j} | j \in \mathbb{N} \right\}, \text{and} \\ f(P^{\text{orb}(G)}) &:= \frac{1}{2^n |P^{\text{orb}(G)}|} \sum_{Q \in P^{\text{orb}(G)}} \operatorname{Tr}(Q\, \mathcal{E}(Q)). \end{aligned}$$

$P^{\text{orb}(G)}$ is called the G-orbit of P. Similarly, one can define the G-orbit on the subset of qubits as the collection of G-orbits of all the Paulis within the support of the subset. Suppose the set S has support $\mathbb{A} \subseteq [n]$. We use the notation $\mathcal{P}_{\mathbb{A}}$ to denote the Pauli group restricted to this subset of qubits. For example, if S only has non-trivial support on qubits $\mathbb{A} = \{0, 1\}$, then $\mathcal{P}_{\mathbb{A}}$ is the two-qubit Pauli group. Also, we write $P_{\mathbb{A}}$ to denote the Pauli restricted to the support $\mathbb{A}$. For instance, $P_{\mathbb{A}}$ for $XXIII$ when $\mathbb{A} = 0, 1$ would refer to the Pauli XX. We define the G-orbit of a set of Paulis $\mathbb{P}_{\mathbb{A}}$ as the union of the individual G-orbits of the Paulis in the set i.e.,

$$\mathbb{P}_{\mathbb{A}}^{\text{orb}(G)} := \{ P_{\mathbb{A}}^{\text{orb}(G)} | P_{\mathbb{A}} \in \mathbb{P}_{\mathbb{A}} \}. \tag{2.88}$$

The marginal probability distribution over $\mathbb{P}_{\mathbb{A}}^{\text{orb}(G)}$ is defined to be the sum of marginal probabilities of all the Paulis in the orbit as

$$\mu(\mathbb{P}_{\mathbb{A}}^{\text{orb}(G)}) := \sum_{Q_{\mathbb{A}} \in \mathbb{P}_{\mathbb{A}}^{\text{orb}(G)}} \mu(Q_{\mathbb{A}}), \tag{2.89}$$

where $\mu(Q_{\mathbb{A}}) := \sum_{\{P \in \mathcal{P}_n | P_{\mathbb{A}} = Q_{\mathbb{A}}\}} p(P)$ is the marginal probability of $Q_{\mathbb{A}}$. The following lemma, which we state here without proof, relates the marginal probability $\mu(\mathbb{P}_{\mathbb{A}}^{\text{orb}(G)})$ to the quantity obtained using PIE namely $f(\mathbb{P}_{\mathbb{A}}^{\text{orb}(G)})$.

Lemma 2. *Suppose $\mathbb{P}_{\mathbb{A}}$ is invariant under the action of G i.e., $G\mathbb{P}_{\mathbb{A}}G^{-1} = \mathbb{P}_{\mathbb{A}}$. The marginal probability distribution $\mu(\mathbb{P}_{\mathbb{A}}^{orb(G)})$ can be obtained from $f(\mathbb{P}_{\mathbb{A}}^{orb(G)})$ using the following equation* [CDDH+23]

$$\mu(P_{\mathbb{A}}^{orb(G)}) = \frac{|P_{\mathbb{A}}^{orb(G)}|}{4^{\mathbb{A}}} \sum_{Q_{\mathbb{A}} \in \mathbb{P}_{\mathbb{A}}} \zeta(Q_{\mathbb{A}}, P_{\mathbb{A}}) f(P_{\mathbb{A}}^{orb(G)}), \tag{2.90}$$

where

$$\zeta(P, Q) = \begin{cases} 1 & \textit{if P and Q commute} \\ -1 & \textit{otherwise.} \end{cases} \tag{2.91}$$

We will use the term *parallel gate supports* (denoted by $\mathbb{A}_i$) to denote sets of pairwise disjoint qubit indices on which a hard cycle acts. For example, consider a hard cycle on a 5-qubit system consisting of two parallel CZ gates on the pairs $\mathbb{A}_0 = \{0, 2\}$, $\mathbb{A}_1 = \{1, 3\}$ and a Hadamard gate on $\mathbb{A}_2 = \{4\}$. Now, we are in a good position to describe the CER protocol.

Protocol 1 Cycle error reconstruction [CDDH+23]

Input: A hard cycle G; the number of parallel gate supports m.

Output: Marginal probability distributions for errors on the union of all m distinct parallel gate supports.

The protocol: Let the parallel gate supports of G be $\mathbb{A}_0, \mathbb{A}_1, \ldots, \mathbb{A}_{s-1}$ and let $\mathbb{S}_m$ be the set of all $\binom{s}{m}$ unions of m distinct gate supports.

For each support $\mathbb{A} \in \mathbb{S}_m$:

1. Invoke $PIE(\mathbb{P}_{\mathbb{A}}, G)$ and collect the output fidelities namely $\{f(Q_{\mathbb{A}}^{\mathrm{orb}(G)}) : Q_{\mathbb{A}} \in \mathbb{P}_{\mathbb{A}}\}$.
2. Use lemma 2 to obtain the marginal probabilities $\mu(Q_{\mathbb{A}}^{\mathrm{orb}(G)})$ from $f(Q_{\mathbb{A}}^{\mathrm{orb}(G)})$.

In chapter 3, we will use the CER protocol as one of the key steps for designing an efficient diagnostic technique to estimate the fidelity of quantum error correction schemes.

2.5 Error correction

In this section, we will briefly discuss classical error correction followed by the basics of quantum error correction.

2.5.1 Classical error correction

Linear codes constitute a huge and important class of classical error correcting codes. For any $[n, k, d]$ classical linear code which encodes k logical bits in n physical bits and has distance d, the following properties are relevant:

- rate = k/n, and
- relative distance = d/n.

It is desirable to have high rate as well as high relative distance. A code $\mathcal{C}$[3] is said to be linear if any linear combination of codewords in $\mathcal{C}$ is also a valid codeword. In this thesis, we will restrict ourselves to binary linear codes. Linear codes can be described by specifying a basis for all the codewords in the form of a generator matrix $G_{\mathcal{C}} \in \mathbb{F}_2^{k\times n}$, where each row of the matrix represents a basis codeword. The row span of the generator matrix contains all the codewords in $\mathcal{C}$. For example, the generator matrix for the $[7, 4, 3]$ Hamming code is given by

$$G_{\mathcal{C}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \tag{2.92}$$

Therefore, any $x \in \mathbb{F}_2^n$ is a valid codeword if there exists $m \in \mathbb{F}_2^k$ such that $mG = x$. Another equivalent representation of a linear code is in the form of parity check matrix $H_{\mathcal{C}} \in \mathbb{F}_2^{n-k\times n}$. A binary string $x \in \mathbb{F}_2^n$ is a valid codeword if $H_{\mathcal{C}} x^T = \mathbf{0}$. In other words, the null space of the matrix $H_{\mathcal{C}}$ is the code space of the code $\mathcal{C}$. The parity check matrix of the Hamming code is

$$H_{\mathcal{C}} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{2.93}$$

[3] This notation for codes should not be confused with the one used for Cliffords earlier. We will not be referring to Cliffords in this section.

The number of distinct possible codewords is referred to as the size of the code. It is denoted by $|\mathcal{C}|$ and is equal to 2^k for a binary linear code. The code space is a k-dimensional linear subspace of $\mathbb{F}_2^n$. For every binary linear code $\mathcal{C}$, we define a dual code $\mathcal{C}^\perp$ such that the codewords for $\mathcal{C}^\perp$ are generated by the rows of the parity check matrix for $\mathcal{C}$. In other words, the generator matrix for $\mathcal{C}^\perp$ is the parity check matrix for $\mathcal{C}$ and vice versa. Also, note that by definition, the codewords of the two codes are orthogonal to each other.

The *minimum distance* or simply the *distance* of a linear code is defined as the minimum hamming distance between all pairs of codewords in the code i.e.,

$$\mathrm{d}_{\mathcal{C}} = \min_{u,v \in \mathcal{C}; u \neq v} d(u,v), \tag{2.94}$$

where $d(u,v)$ is the Hamming distance between vectors i.e., number of positions in which they differ for binary codes. To calculate the distance for linear codes, it is sufficient to find the codeword with the least Hamming weight i.e., least number of $1'$s. Note that, the brute force method to do this entails evaluating the Hamming weights of all the $|\mathcal{C}| = 2^k$ codewords of length n. Another equivalent way to find the distance is to use the following definition which uses the parity check matrix $H_{\mathcal{C}}$:

$$\mathrm{d}_{\mathcal{C}} = \min_{\{x : H_{\mathcal{C}} x^T = \mathbf{0}\}} |x|. \tag{2.95}$$

Note that any x that satisfies $H_{\mathcal{C}} x^T = \mathbf{0}$ is a valid codeword and implies the existence of a set of dependent columns in $H_{\mathcal{C}}$. Specifically, the set of columns $\{H_{\mathcal{C}}[i] : i \in S\}$ are linearly dependent, where $S = \{i \in [n] : x_i = 1\}$ and $H_{\mathcal{C}}[i]$ denotes the i^{th} column of $H_{\mathcal{C}}$. In other words, distance is the size of the smallest set of columns that are linearly dependent in $H_{\mathcal{C}}$. The brute-force method to use this definition would require iterating over all subsets of columns in increasing order of size until a dependence is detected. In both the methods, we are essentially finding the codeword with the least Hamming weight.

2.5.2 Quantum error correction

Quantum error correction (QEC) is one of the fundamental parts of any fault-tolerant quantum scheme. It employs redundancy and symmetry to detect and correct errors that may have corrupted an incoming quantum state. The process of (quantum) error correction aims to protect (noisy) physical qubits and store them as (noise-protected) logical qubits. A code that uses n physical qubits to produce k logical qubits is referred

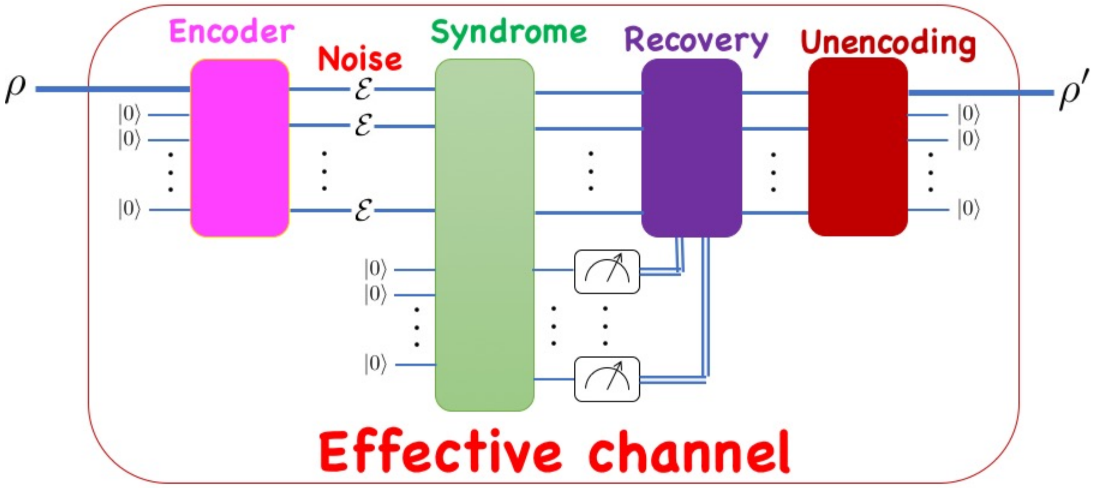


Figure 2.5: The above figure shows the key steps of a quantum error correcting scheme. The effective channel is a virtual construct that encapsulates all of these steps into a single process that directly acts on the logical qubit.

to as an $[[n, k]]$ code. Fig 2.5 highlights the key steps involved in a quantum error correcting circuit.

1. Encoding : In this step, the input state is padded with some blank qubits and then encoded using a unitary operator,

$$\rho \mapsto \overline{\rho} = U \left(\rho \otimes |0\rangle\langle 0|^{\otimes (n-k)} \right) U^{\dagger}. \tag{2.96}$$

   The unitary operator U is the encoding circuit for the underlying error correcting code.

2. Noise : Although the noise is present all throughout the computation process, we model its effect on the encoded state $\bar{\rho}$ by an explicit map $\mathcal{E}^{\otimes n}$ as:

$$\bar{\rho} \mapsto \mathcal{E}^{\otimes n} \left(\overline{\rho} \right) \tag{2.97}$$

   Please note that although the figure and the above equation depicts the application of identical noise on each qubit, in some cases we will apply correlated noise as well.

3. Syndrome detection : This is a crucial step where we check whether the symmetry of the encoded state was affected by the noise or not. If the noise breaks the symmetry, it leaves a signature for the same, which we call a *syndrome*. The process of syndrome detection involves coupling of the n physical qubits to $(n-k)$ ancillary qubits and then measuring the ancilla to detect errors. Given the outcome of the measurement was a syndrome $s \in \{1, 2, \ldots 2^{n-k}\}$, the transformation of the quantum state is described by,

$$\mathcal{E}^{\otimes n}\left(\overline{\rho}\right) \mapsto \overline{\rho}_s = \frac{\Pi_s \mathcal{E}^{\otimes n}\left(\overline{\rho}\right) \Pi_s}{\mathrm{tr}(\Pi_s \mathcal{E}^{\otimes n}\left(\overline{\rho}\right))}, \tag{2.98}$$

where Π_s is the projector onto the eigenspace associated to the syndrome s.

4. Decoding and recovery: This is a classical inference step wherein we attempt to guess the error that might have led to the syndrome s in the previous step. In general, there can be multiple errors that lead to the same syndrome. We will discuss the popular strategies for decoding later. Given the recovery chosen is R_s, the following transformation ensues

$$\overline{\rho}_s \mapsto R_s \overline{\rho}_s R_s. \tag{2.99}$$

5. Un-encoding : This final step maps the logical state back to physical state :

$$R_s \overline{\rho}_s R_s \mapsto \rho' \otimes |0\rangle\langle 0|^{\otimes(n-k)}. \tag{2.100}$$

Combining all of the above steps, we map an n qubit physical noise $\mathcal{E}^{\otimes n}$ to an effective logical channel acting on k qubits, which we call $\mathcal{E}_1^s$. This notation implies that the effective channel is conditioned on observing a syndrome s in the syndrome extraction step followed my appropriate recovery application. The subscript denotes that we have applied error correction once to the bare physical channel.

Stabilizer codes

Traditionally, quantum error correcting codes were specified by explicitly mentioning the expansion of logical basis states in terms of 2^n physical states. For instance, a popular code with $n = 7$ and $k = 1$ known as *Steane code* has the logical states spanned by

$$
\begin{aligned}
|\bar{0}\rangle &= \frac{1}{\sqrt{8}}(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle \\
&\qquad + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle), \text{ and} \\
|\bar{1}\rangle &= \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle \\
&\qquad + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle).
\end{aligned}
\tag{2.101}
$$

To avoid the challenges associated with using an exponential number of basis vectors to specify a code, Gottesman in Ref. [Got97] came up with a formalism to describe quantum codes succinctly called the Stabilizer formalism and the corresponding codes are called *Stabilizer codes*. The complexity of the description of these codes is linear in the number of physical qubits n.

The code space for *Stabilizer codes* is defined to be the simultaneous +1 eigenspace of the Stabilizer generators $\{S_i\}$ i.e.

$$Q = \{|\psi\rangle : \; S_i|\psi\rangle = |\psi\rangle \; \forall \; i \; \in \{1, 2, \ldots, n-k\}\} \tag{2.102}$$

where $\{S_i\}$'s are Pauli matrices. The corresponding Abelian subgroup $\mathcal{S} = \langle S_1, \ldots, S_{n-k}\rangle$ is called the Stabilizer group, and its elements, the stabilizers. Logical operations on the encoded states $|\overline{\psi}\rangle$ are elements of the normalizer: $\mathcal{N}(\mathcal{S})$, where $\mathcal{N}(\mathcal{S})$ is the set of all Pauli operators which commute with every Stabilizer generator. We will sometimes refer to logical Pauli operators $\overline{P} \in \mathcal{N}(\mathcal{S})/\mathcal{S}$ simply as logicals. The Hamming weight of the logical Pauli operator with least number of non-identity Paulis is called the *distance* of the stabilizer code. A code with distance d can correct any Pauli error of weight up to $t = \lfloor \frac{d-1}{2} \rfloor$. The revised notation for such a code with n physical qubits, k logical qubits and distance d is $[[n, k, d]]$. Note that, the same notation with single brackets i.e., $[n, k, d]$ was used to refer to a classical error correcting code with n physical bits, k logical bits and distance d. A classical code with distance d can correct errors of weight up to $t = \lfloor \frac{d-1}{2} \rfloor$ bits.

For Stabilizer codes, in the syndrome detection step described above (step 2), an error $P \in \mathcal{N}(\mathcal{S})/\mathcal{S}$ is detected by measuring the stabilizer generators. The *error-syndrome* of E, denoted by $s(E) \in \mathbb{F}_2^{n-k}$ is an $(n-k)$ bit sequence, where $(-1)^{s_i(E)}$ denotes the outcome of measuring the stabilizer generator S_i on $E|\overline{\psi}\rangle$. In other words, $s_i(E)$ is 0 whenever $[E, S_i] = 0$ and 1 otherwise. The effect of measuring s is captured by the projector Π_s given by

$$\Pi_s = \prod_{i=1}^{n-k} \frac{\mathbb{I} + (-1)^{s_i} S_i}{2} \; . \tag{2.103}$$

A Pauli error T can be decomposed with reference to a stabilizer code:

$$T = \overline{T} \; S_T \; E_T \; , \tag{2.104}$$

where S_T is an element of the stabilizer group $\mathcal{S}$, $\overline{T}$ is a logical Pauli operator in $\mathcal{L} = \mathcal{N}(\mathcal{S})/\mathcal{S}$, and E_T is an element of $\mathcal{N}(\mathcal{L})/\mathcal{S}$, usually called a pure error [LB13; Pou06]. Unlike pure errors, stabilizers and logical operators commute with quantum error correction (QEC) routines. A Pauli error T can be compiled into QEC, by a simple change to the decoder, resulting in a new quantum error correction routine $\mathsf{QEC}(T)$. In particular, if the syndrome measurement outcome in QEC is s, then the decoder in $\mathsf{QEC}(T)$ is given as input the error syndrome $s \oplus s(T)$, where $s(T)$ is the error syndrome of T, and the recovery it prescribes is conjugated by $\overline{T}$. This fact will be useful in chapter 3 when we discuss combining noise tailoring methods with quantum error correction.

While the quantum error correction formalism has been traditionally developed to address Pauli errors, realistic noise processes are often inaccurately described by the Pauli error model. There have been only a handful studies [RDM02; CWB+17; GSL+16; HDF19; BEK+18; DP17; BWG+18; IP18] of quantum error correction schemes for generic Markovian noise processes. Unlike a Pauli error model, a general Markovian process cannot be understood as a probability distribution over several unitary errors. Instead one is left with studying the full quantum evolution of an n qubit system. The composite process encapsulating (i) the physical noise $\mathcal{E}_0$, (ii) measuring a syndrome outcome s, and (iii) applying a recovery R_s prescribed by a decoder, is the *effective logical channel* $\mathcal{E}_1^s$ [RDM02]. Its action on an encoded state $\overline{\rho}$ is summarized by:

$$\mathcal{E}_1^s(\overline{\rho}) = \frac{R_s \; \Pi_s \; \mathcal{E}_0(\overline{\rho}) \; \Pi_s \; R_s^\dagger}{\mathsf{Pr}(s)} \; , \tag{2.105}$$

where $\mathsf{Pr}(s)$ is the probability of measuring the syndrome outcome s. Taking the average over syndromes, we find the average logical channel $\overline{\mathcal{E}}_1$ given by

$$\overline{\mathcal{E}}_1(\overline{\rho}) = \sum_s \mathsf{Pr}(s) \mathcal{E}_1^s(\overline{\rho}) \; . \tag{2.106}$$

Note that, in practice, to compute the average channel one can avoid the computation of $\mathsf{Pr}(s)$ in equations (2.105) and (2.106). While physical error rates are measured by noise-metrics on $\mathcal{E}_0$, logical error rates are measured by noise metrics on $\overline{\mathcal{E}}_1$.

The two most popular decoding strategies for Stabilizer codes are described below.

- Minimum weight decoding (MWD) : A correction R_s in Eq.(2.105) takes the form $T_s\, L_\star\, S$, where T_s is the pure error corresponding to the syndrome s, $L_\star$ is a logical and S is a stabilizer. A minimum weight decoding strategy [HLG11] prescribes a correction R_s which has the least Hamming weight.
- Maximum likelihood decoding (MLD) : This decoding strategy [HLG11] picks an $L_\star$ that maximizes the following probability

  $$L_\star = \arg\max_{L \in \mathcal{L}} \; Pr(L|s) \tag{2.107}$$

  where $Pr(L|s) = \sum_{S \in \mathcal{S}} Pr(T_s \cdot L \cdot S)$ is the sum of the probabilities of all the errors in a coset of the Stabilizer group with respect to logicals.

Note that the decoder essentially only has to prescribe an appropriate $L_\star$ since the stabilizers leave the code space invariant and T_s is fixed by the syndrome observed in the syndrome detection step. MLD is known to be optimal [IP15] whereas MWD is sub-optimal.

A stabilizer code and a decoder pair are designed to correct a target set of errors, called *correctable errors* [Ste06; Rau12] $\mathcal{E}_C$. For an $[[n, k]]$ code, $\mathcal{E}_C$ can be partitioned into 2^{n-k} disjoint subsets $\mathcal{E}_{C,1}, \ldots, \mathcal{E}_{C,2^{n-k}}$, each of which can be identified with a unique syndrome measurement outcome. The construction of the set $\mathcal{E}_{C,s}$ closely depends on the choice of a decoder. Recall that the output of a decoder on input syndrome s is a Pauli recovery operator R_s, i.e., $R_s \in \mathcal{E}_{C,s}$. A key observation to construct elements in $\mathcal{E}_{C,s}$ besides R_s is that any error of the form $R_s S$ where S is an element of the stabilizer group is also correctable, so, $\mathcal{E}_{C,s} = \{R_s S : S \in \mathcal{S}\}$.

Concatenated codes

In general, logical error rates can be improved by increasing the number of physical qubits. Concatenated quantum codes are a popular family of codes of increasing sizes [KL96], and are often used to guarantee error suppression in fault tolerance proofs [AGP07; JOL14]. Concatenation provides a simple recipe to construct a large code from two small codes. In this scheme, the physical qubits of a $[[n_2, k_2]]$ code $\mathcal{C}_2$ are encoded using a $[[n_1, k_1]]$ code $\mathcal{C}_1$, yielding a $[[n_1 n_2, k_1 k_2]]$ code. This procedure can be repeated recursively with L code-blocks $\mathcal{C}_1, \ldots, \mathcal{C}_L$ where $\mathcal{C}_i$ is a $[[n_i, k_i]]$ code, yielding a *level*$-L$ concatenated code. In this thesis, we consider concatenated codes where the constituent code-blocks are described by the same code. Furthermore, the number of logical qubits encoded in each code block is one.

Figure 2.6 presents a schematic of the concatenated code structure. The recursive encoding structure is depicted as a tree where a horizontal layer corresponds to a level. The i-th node at level ℓ denotes a quantum error correcting code block $\mathcal{C}_{\ell,i}$. The sub-tree of the node is itself a concatenated code, denoted by $\mathcal{C}^{\star}_{\ell,i}$, consisting of $(n^{\ell}-1)/(n-1)$ code blocks. There are $(n-1)$ independent stabilizer measurements corresponding to each of the code-blocks of $\mathcal{C}^{\star}_{\ell,i}$. The resulting error syndrome $s(\mathcal{C}^{\star}_{\ell,i})$ has $(n^{\ell}-1)$ bits, which can be grouped into subsets of $(n-1)$ bits that are identified by the code-blocks. We will often identify the subset of syndrome bits obtained by measurements on a code-block $\mathcal{C}_{\ell,j}$ by $s(\mathcal{C}_{\ell,j})$.

For quantum error correction simulations in this thesis, we apply a simple, but sub-optimal decoding strategy for concatenated codes, which functions independently on each block [Got97]. We consider the following iterative routine for QEC in concatenated codes. For each level $\ell = 1, \ldots, L$: (i) syndromes are extracted for each code block $\mathcal{C}_{\ell,1}, \ldots, \mathcal{C}_{\ell,n}$, and (ii) a minimum-weight correction [HLG11] is applied in each case. Although we assume the popular choice of minimum-weight decoder in (ii), the methods prescribed in this thesis can be adapted to any lookup table decoder [TS14]. The correction applied at any level depends on the syndrome history of the code blocks in the lower levels.

The effective channel for a level ℓ concatenated code can also be computed in a recursive fashion, i.e., using Eq.(2.105) where $\mathcal{E}_0$ is replaced by effective channel on the level $(\ell-1)$ code blocks, i.e., $\mathcal{E}^{s}_{\ell-1,1} \otimes \ldots \otimes \mathcal{E}^{s}_{\ell-1,n}$ [RDM02; Pou06]. The average of logical channels $\mathcal{E}^{s}_{\ell}$ over all syndrome outcomes, denoted by $\overline{\mathcal{E}}_{\ell}$ is expressed as:

$$\overline{\mathcal{E}}_{\ell} = \sum_{s} \mathcal{E}^{s}_{\ell} \mathsf{Pr}(s) \, , \tag{2.108}$$

where $\mathsf{Pr}(s)$ is the probability of observing the outcome s [IP18; BEK+18; CWB+17]. The average logical channel $\overline{\mathcal{E}}_{\ell}$ indicates how quantum error correction suppresses the effect of physical errors, on average. We will use logical infidelity $r(\overline{\mathcal{E}}_{\ell})$ [IP18; GSL+16] as a measure of the logical error rate defined by:

$$r(\overline{\mathcal{E}}_{\ell}) = \sum_{s} \mathsf{Pr}(s) r(\mathcal{E}^{s}_{\ell}) \, . \tag{2.109}$$

In general, we need to overcome two challenges to estimate $r(\overline{\mathcal{E}}_{\ell})$. First, given a syndrome outcome s, we need an efficient method to compute its probability $\mathsf{Pr}(s)$, and the associated effective channel $\mathcal{E}^{s}_{\ell}$. Second, we need an efficient method to compute the sum over the exponentially large set of syndrome outcomes. For instance, the number

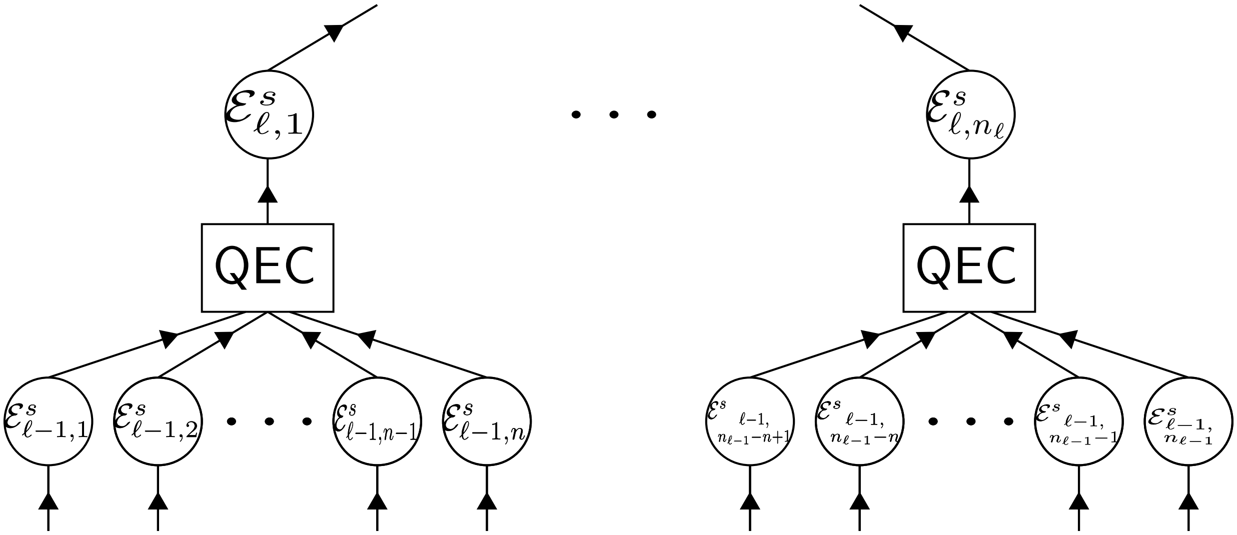

$\mathcal{E}^s_{\ell,1}$
$\mathcal{E}^s_{\ell,n_\ell}$
QEC
QEC
$\mathcal{E}^s_{\ell-1,1}$
$\mathcal{E}^s_{\ell-1,2}$
$\mathcal{E}^s_{\ell-1,n-1}$
$\mathcal{E}^s_{\ell-1,n}$
$\mathcal{E}^s_{\ell-1,n_{\ell-1}-n+1}$
$\mathcal{E}^s_{\ell-1,n_{\ell-1}-n}$
$\mathcal{E}^s_{\ell-1,n_{\ell-1}-1}$
$\mathcal{E}^s_{\ell-1,n_{\ell-1}}$

Figure 2.6: The above figure shows the level ℓ of concatenated code, resembling a tree. Each of the horizontal layers refer to a concatenation level. While the physical qubits are placed at level $\ell = 0$, the encoded qubit is at the topmost level $\ell = L$. The blocks in the intermediate levels denote quantum error correcting codes.

of syndrome outcomes for the level $\ell = 2$ concatenated Steane code is approximately 10^{14}.

To circumvent the first problem, we have focused our attention to concatenated codes, for which there exists an efficient method to compute $\mathsf{Pr}(s)$ and $\mathcal{E}^s_\ell$. There is no known method to overcome the second challenge for concatenated codes.

However, our problem of approximating the average logical error rate in Eq.(2.109), falls into the general framework of problems addressed by Monte Carlo sampling. A sampling estimate for $r(\overline{\mathcal{E}}_\ell)$ is given by

$$r(\hat{\mathcal{E}}_\ell) = \frac{1}{N}\sum_{\hat{s}} r(\mathcal{E}^{\hat{s}}_\ell) \; , \tag{2.110}$$

where $\hat{s}$ is an outcome sampled from the syndrome distribution $\mathsf{Pr}(s)$. As the number of syndromes grow exponentially with the number of levels, Monte Carlo sampling techniques described in section A.6 of the appendix can be used to estimate this average.

Now, we provide the expression to calculate the average logical infidelity for a code under a noise process $\mathcal{E}$: [IJB+22]

$$r(\overline{\mathcal{E}}_1) = 1 - \sum_{\substack{E,E' \in \mathcal{E}_C \\ s(E)=s(E')\,,\overline{E}=\overline{E}'}} \phi(E)\ \phi^{\star}(E')\ \chi_{E,E'}\ , \tag{2.111}$$

where $\chi_{i,j}$ represents the $(i,j)^{th}$ entry of the $\chi-$matrix of $\mathcal{E}$, $\mathcal{E}_C$ is the set of correctable errors, $\overline{E}$ is the logical component in the decomposition of E with respect to the Stabilizer group and $\phi(E)$ is specified by $R_{s(E)}E = \phi(E)\ S$ for any Pauli error E and some stabilizer S. We use this expression at various points to calculate the logical infidelity. To calculate the entries of the $\chi-$matrix of the effective logical channel we use the following general expression: [IJB+22]

$$\chi(\overline{\mathcal{E}}_1)_{l,m} = \sum_{\substack{E,E' \in \mathcal{E}_C \\ s(E)=s(E')\,,\overline{E}=\overline{E}'}} \phi(E,l)\ \phi^{\star}(E',m)\ \chi_{E\overline{P}_l,\overline{P}_m E'}\ . \tag{2.112}$$

where $R_{s(E)}\ \ |E\ \ \overline{P}_l| = \phi(E,l)\ \ S\ \ |\overline{P}_l|$, for $l \in \{0,1,2,3\}$, any Pauli error E and some stabilizer S. Here $|P|$ stands for the bare Pauli without any associated global phase. The last two equations have been derived in appendix section A.1 for completeness. In chapter 4, we will calculate the $\chi-$matrix for logical channels at higher levels i.e., for $\ell > 1$ by recursing the expression in Eq.(2.112) and using the entries of $\chi(\overline{\mathcal{E}}_{\ell-1})$ in the right hand side to evaluate $\chi(\overline{\mathcal{E}}_\ell)$.

2.6 Property testing

The field of property testing deals with investigating *global* properties of large objects [Gol17]. The goal is to determine if a given object satisfies a certain property or is *far* from all the objects satisfying the given property. The notion of the distance between objects depends on their nature. The nature of objects spans various categories including strings, matrices, graphs and functions to name a few. For instance, the distance could be the Hamming distance if one is concerned with binary strings. For functions, that are usually supplied as oracles, the distance between any two of them is the number of inputs on which their outputs differ. For graphs, one can define the distance to be the Hamming distance between their adjacency matrices.

Typically, the algorithms that are interesting for this task are super fast and look at only a tiny fraction of the input. In most cases, there is a linear time algorithm to exactly determine whether the object has the given property. However, we are operating in the regime where it is impossible to look at the entire input. Therefore, we are interested in approximating this decision by looking at a small part of the given input. Adding the promise that the given input either has the property or is far from the set of objects having the property i.e., it is not on the *boundary*, makes the problem simpler, opening up the possibility of efficient testing algorithms. However, one needs to be careful to not trivialize the problem entirely. We will elaborate later on this by describing situations where this can occur.

Before looking at problems that can have efficient testers, let us define the notion of distance between strings on some alphabet Σ. Let $x, y \in \Sigma^n$. The Hamming distance (or discrete metric) between them is defined to be $\delta(x,y) = |\{i \in [n] : x_i \neq y_i\}|$. In other words, it is the count of locations where the two inputs differ. We will often use the notation $|x - y|$ to denote this quantity. Along similar lines, we will say that an input $x \in \Sigma^n$ is ϵ-far from a set S if $\delta_S(x) := min_{z \in S}\delta(x,z) = \epsilon n$. Therefore, the testing algorithm distinguishes the set of input that belong to the set S from the set of inputs that are ϵ-far from the set S. In other words, all inputs $x : 0 < \delta_S(x) \leq \epsilon n$ are ignored. An algorithm to do this is referred to as ϵ-testing algorithm or simply an ϵ-tester.

Formally, a property testing algorithm with proximity parameter ϵ is a randomized algorithm which on input x (being tested for property P) satisfies the following conditions:

- If x is in P, the tester accepts with probability at least $2/3$.
- If x is ϵ-far from P, the tester rejects with probability at least $2/3$.

Such a tester is said to have *two-sided error*. This is also referred to as the *bounded error* setting sometimes. On the other hand, a tester with *one-sided error* satisfies the stronger condition that it accepts all inputs that have the property (x in P) with certainty (probability 1). As is standard in the theory of randomized algorithms, the core idea behind having probability $> 1/2$ is that the probability of success can be amplified arbitrarily by repeating the algorithm multiple times.

2.6.1 Testing binary strings

In this section, we discuss testing properties of binary strings [Gol17]. The first property concerns determining if a given binary string has majority $1'$s. Let $\textsc{Maj} = \{x : \sum_{i=1}^{|x|} x_i > |x|/2\}$. This property can be tested in poly($1/\epsilon$)-time whereas it can be shown that no sub-linear-time (randomized) algorithm can solve the exact decision version of this problem. A general approach for most testers is to query a (small) sample from the given input object and test if the property holds on the sample. If it does, ACCEPT the input with high probability ($> 1/2$). If it does not, declare that the input is far from the property with high probability ($> 1/2$). We will fix the probability of success in both the cases to be $\geq 2/3$ but any fraction more than $1/2$ works since it can be amplified with multiple repetitions and taking a majority vote of the individual runs.

Proposition 3. *There exists a randomized algorithm which runs in $O(1/\epsilon^2)$-time and decides where a given string $x \in \textsc{Maj}$ or x is ϵ-far from $\textsc{Maj}$ [Gol17].*

Proof. The algorithm is similar to the general theme described before. It queries the input at $m = O(1/\epsilon^2)$ uniformly and independently distributed locations. Let the indices queried be given by $i_1, i_2, \ldots, i_m$. The algorithm accepts the input if the average of the entries queried i.e., $\sum_{j\in[m]} x_{i_j}/m$ is more than $(1-\epsilon)/2$. It can be shown using the Chernoff bound that with probability at least $2/3$, the average of the sample is close to the true average. Precisely, the following is true:

$$Pr_{i_1,i_2,\ldots,i_m\in[|x|]}\left[\left|\frac{\sum_{j\in[m]} x_{i_j}}{m} - \frac{\sum_{k\in[|x|]} x_k}{|x|}\right| \leq \frac{\epsilon}{2}\right] \geq \frac{2}{3}. \tag{2.113}$$

Now, lets analyze the two cases and determine the algorithm's output on them. Suppose $x \in \textsc{Maj}$. This implies that $\frac{\sum_{k\in[|x|]} x_k}{|x|} > 1/2$. Therefore, according to the previous equation, with probability at least $2/3$, $\frac{\sum_{j\in[m]} x_{i_j}}{m} > (1-\epsilon)/2$. Thus, the algorithm will accept x in this case with high probability. On the other hand, if x is ϵ-far from $\textsc{Maj}$, it implies that $\frac{\sum_{i\in[|x|]} x_i}{|x|} \leq (0.5-\epsilon)$. Therefore, according to the previous equation, with probability at least $2/3$, $\frac{\sum_{j\in[m]} x_{i_j}}{m} \leq (1-\epsilon)/2$. Thus, the algorithm will reject x in this case with high probability. □

In general, all *symmetric properties* of binary strings can be efficiently tested. A property S is symmetric if it holds that $x \in S$ if and only if all permutations of x are also in S. An example of a non-symmetric property that can be efficiently tested is whether a given string is sorted or not. Let $\textsc{Sorted} = \{x : x_i \leq x_{i+1}\}$. It can be tested if a string $x \in \textsc{Sorted}$ or ϵ-far from it in $O(1/\epsilon)$ time.

2.6.2 Testing functions

In this section, we briefly explore testing properties of functions. In particular, we will describe a tester for checking whether a given function is linear [BLR93]. Suppose, we are given two groups G and H with the same group operation denoted by $+$. We call a function $f : G \to H$ a (group) homomorphism if the following condition holds for all $x, y \in G$:

$$f(x+y) = f(x) + f(y) \text{ and } f(1_G) = 1_H.$$

Testing Homomorphism

Input: Function $f : G \to H$, where G and H are groups with same group operation denoted by $+$.

Promise: f is a (group) homomorphism or is δ-far from all homomorphisms.

Output: ACCEPT if f is a homomorphism, REJECT otherwise.

Recall that the distance between the two functions implied here is defined by the fraction of inputs on which their outputs disagree. The testing procedure for the above problem is described in Algorithm 1.

Algorithm 1: Testing Homomorphism

```
Select $x, y \in G$ uniformly at random;
Query $f$ at $x, y, x+y$;
if $f(x+y) = f(x) + f(y)$ then
    ACCEPT;
else
    REJECT;
end
```

It is easy to see that the tester always accepts inputs which are homomorphisms with probability one. In what follows, we will show a partial proof that the tester will reject homomorphisms which satisfy the testing promise with probability at least $3\delta - 6\delta^2$. This lower bound is only meaningful for $\delta \in [0, 1/4]$. For a stronger lower bound on the probability of rejection which is valid for all δ, we refer the refer the readers to Refs. [BLR93; Gol17].

Proposition 4. *Given $f : G \to H$ is at a distance δ from the set of homomorphisms from G to H, algorithm 1 rejects it with probability at least $3\delta - 6\delta^2$.*

Proof. Let h be the homomorphism closest to f. The rejection probability $Pr_{x,y \in G}[f(x+$

$y) \neq f(x) + f(y)]$ is lower bounded by

$$Pr_{x,y\in G}[f(x) \neq h(x) \wedge f(y) = h(y) \wedge f(x+y) = h(x+y)] \tag{2.114}$$
$$+Pr_{x,y\in G}[f(x) = h(x) \wedge f(y) \neq h(y) \wedge f(x+y) = h(x+y)] \tag{2.115}$$
$$+Pr_{x,y\in G}[f(x) = h(x) \wedge f(y) = h(y) \wedge f(x+y) \neq h(x+y)], \tag{2.116}$$

since these are disjoint events and the condition $f(x+y) \neq f(x) + f(y)$ implies that the two functions f and h must disagree on at least one of $x, y, x+y$ (given $h(x+y) = h(x) + h(y)$). We used the notations $\wedge$ and $\vee$ to denote logical AND and logical OR respectively. Since they can potentially disagree on more than one point, this lower bound is not tight. Now, we lower bound the first of the three terms whereas the other two can be bounded analogously.

$$\begin{aligned}
&Pr_{x,y}[f(x) \neq h(x) \wedge f(y) = h(y) \wedge f(x+y) = h(x+y)] \\
&= Pr_{x,y}[f(x) \neq h(x)] - Pr_{x,y}[f(x) \neq h(x) \wedge (f(y) \neq h(y) \vee f(x+y) \neq h(x+y))] \\
&\geq Pr_{x,y}[f(x) \neq h(x)] - (Pr_{x,y}[f(x) \neq h(x) \wedge f(y) \neq h(y)] \\
&\qquad\qquad\qquad\qquad + Pr_{x,y}[f(x) \neq h(x) \wedge f(x+y) \neq h(x+y)]) \\
&= \delta - (\delta^2 + \delta^2) \\
&= \delta - 2\delta^2,
\end{aligned}$$

where in the second last equality we use the fact that x, y are independently and uniformly distributed in G and so are $x, x+y$. Adding up the contributions from the other two terms completes the proof. □

We briefly mention some other interesting properties of functions which can be tested efficiently. Let $f : \{0,1\}^\ell \to \{0,1\}$ be a Boolean function. The following properties of f have efficient testers [FKR+04; PRS02]:

1. Dictatorship: The goal is to detect if f depends *only* on a single Boolean variable i.e., $f(x) = x_i + b$ for some $i \in [l]$ and $b \in \{0,1\}$.

2. Junta (of size k): The goal is to detect if f depends on at most k Boolean variables i.e., $f(x) = f'(x_S)$ for some $S \subset [\ell], |S| = k$ and $f' : \{0,1\}^k \to \{0,1\}$.

3. Monomial (of size k): The goal is to detect if f is a conjunction of exactly k Boolean literals i.e., $f(x) = \wedge_{i\in S}(x_i \oplus b_i)$ for some $S \subset [\ell], |S| = k$ and $b_1, b_2, \ldots, b_\ell \in \{0,1\}$.

2.6.3 Graph testing

In this section, we discuss testing properties of graphs. We call any subset of graphs a *symmetric graph property* if it is closed under graph isomorphism (relabelling of vertices) i.e., if a graph $G = (V, E)$ has the property Π, then any permutation π of vertices V leading to $\pi(G)$ also has the property Π. The graph is essentially supplied as an oracle function to the tester. Graphs can be modelled in multiple ways when it comes to testing their properties. The two most common models are:

1. *Adjacency predicate model*: In this model, the tester is given access to a function $g : V \times V \to \{0,1\}$ such that $g(u,v) = 1$ if and only if there is an edge between vertices u and v in the graph G, and $g(u,v) = 0$ otherwise. In other words, this model helps query each entry of the adjacency matrix of the graph. Given two graphs $G = (V, E)$ and $G' = (V, E')$ with corresponding oracle functions g and g', the relative distance between them is the number of pairs of input $(u,v) \in V \times V$ such that $g(u,v) \neq g'(u,v)$. This is analogous to the notion of distance between functions discussed in the previous section. This distance can also be interpreted as the Hamming distance between the adjacency matrices.

2. *Incidence function model*: In this model, the oracle function corresponding to a graph $G = (V, E)$ has the structure $g : V \times [D] \to V \cup \{\perp\}$, where we assume that the maximum number of vertices connected to any vertex (degree of the graph) is D. Given an input $(u,i) \in V \times [D]$, $g(u,i)$ returns the i^{th} neighbour of the vertex u. If some vertex u has $j < D$ neighbours, then $g(u, j+1) = g(u, j+2) = \ldots = g(u, D) = \perp$. The notion of distance is similar to the previous model: the distance is the number of inputs for which the oracle outputs are different.

Some graph properties that can be tested efficiently include:

- *k-Colorability* - The number of queries required to test k-colorability is $poly(k/\epsilon)$. The tester has one sided error i.e., it always *accepts* when the graph is k-colorable and with high probability *rejects* the input if it is not. The special case of $k = 2$ leads to testing if a graph is bipartite. For this case, the running time is $O(1/\epsilon^2)$, whereas for $k > 2$, the running time scales as $e^{poly(1/\epsilon)}$.

- *ρ-Clique* - This property tests if a given graph $G = (V, E)$ has a clique of size $\rho|V|$ for any fixed $\rho > 0$.

- *ρ-Cut* - This property tests if a given graph $G = (V, E)$ has an edge cut of size $\rho|V|^2$ for any fixed $\rho > 0$.

Except *k-Colorability*, the rest of the testers have two-sided error. All of the above are instances of the General Graph Partition Testing problem. This problem asks whether there exists a partition such that the number of vertices in each part and the number of edges between each pair of parts fall between given lower and upper bounds. The tester for this can be found in Ref. [GGR96]. A characterization of all the natural graph properties which are testable with one-sided error is presented in Ref. [AS08].

Chapter 3

Efficient diagnostics for quantum error correction

> This chapter consists of a literal transcription of [IJB+22] for which my contribution was major. Some notation and stylistic changes have been done to be consistent with the rest of thesis.

Noise is pervasive in quantum processing and must be overcome to achieve the disruptive capabilities of quantum computing. Fault tolerance (FT) guarantees reliable logical quantum computation in the presence of noise under prescribed conditions often oversimplified as achieving a threshold on gate error rates. However, achieving low logical error rates in practice is challenging, in part because of the large overheads required in terms of the number of additional qubits and gates. Optimizing quantum error correction (QEC) strategies for a particular platform requires accurate prediction of its expected logical performance. For instance, in the presence of biased noise [AP08; RGB+17; TBF18; GM19; TBF+20; BATB+21], tailored codes have been shown to outperform traditional codes that are designed to correct unstructured noise. However, bias is only one of the exponentially many parameters that describe the noise on n physical qubits. This chapter addresses the lack of tools for predicting the logical performance of a fault tolerant architecture based on a description of noise at the physical level.

Related work. The existing framework for choosing a FT scheme is centered around the threshold theorem [AGP07; CTV17] which provides a threshold on the physical noise strength below which reliable quantum computation can be guaranteed. However, directly applying the theorem to realistic noise has several challenges. The FT threshold is derived under oversimplified conditions that implicitly model a physical noise process as an incoherent error model with the same diamond distance. This leads to loose estimates of the logical performance when the noise has coherence or strong correlations. Another is that diamond distance, which is usually invoked for assessing error rates in FT proofs, cannot be measured in a scalable way [MC13]. It has been shown that the resource overheads for a fault tolerant architecture depend critically on the precise relationship between the architecture and the underlying error model. While there are several well-studied error metrics, none of them can accurately predict the logical error rate of a QEC [IP18]. In this work we address this crucial deficiency prevalent in all these metrics.

Our contributions. We present a new figure of merit specifically tailored to predict the performance of a class of error correcting codes, namely concatenated codes, which can be measured efficiently using experimental protocols. As opposed to average gate fidelity and diamond distance, our approach captures the interplay between the physical noise model and the choice of a fault tolerant architecture. Our method leverages Randomized Compiling (RC) [WE16] to create an effective Pauli noise on the physical qubits, and then uses cycle error reconstruction (CER) techniques [EWP+19; FW20; CDDH+23] to estimate Pauli error probabilities. An overview of these methods can be found in section 2.4. Using these experimental data, we design a *logical estimator* that predicts the total probability of Pauli errors that a code cannot correct. While exactly computing this quantity is inefficient for a generic code, we introduce an efficient approximation for concatenated codes. We provide a bound on the efficiency and demonstrate the accuracy of our method through numerical simulations in several noise scenarios of interest. Finally, as an application, we demonstrate how the logical estimator pinpoints the selection of a suitable error correcting code for differing noise environments.

3.1 Methods

While the special setting of Pauli errors drastically simplifies the predictability problem, realistic noise processes are nonetheless poorly described by Pauli error models. To circumvent this problem, we recall a straightforward application of RC [WE16] to FT circuits, that allows us to model the effect of complex noise processes by simple Pauli errors. In other words, RC ensures that there is no effect on the logical error rate from parameters of the physical channel other than the Pauli error probabilities. The physical twirling gates required to do RC can be absorbed into the logical gadgets of FT circuits at no additional cost in overhead.

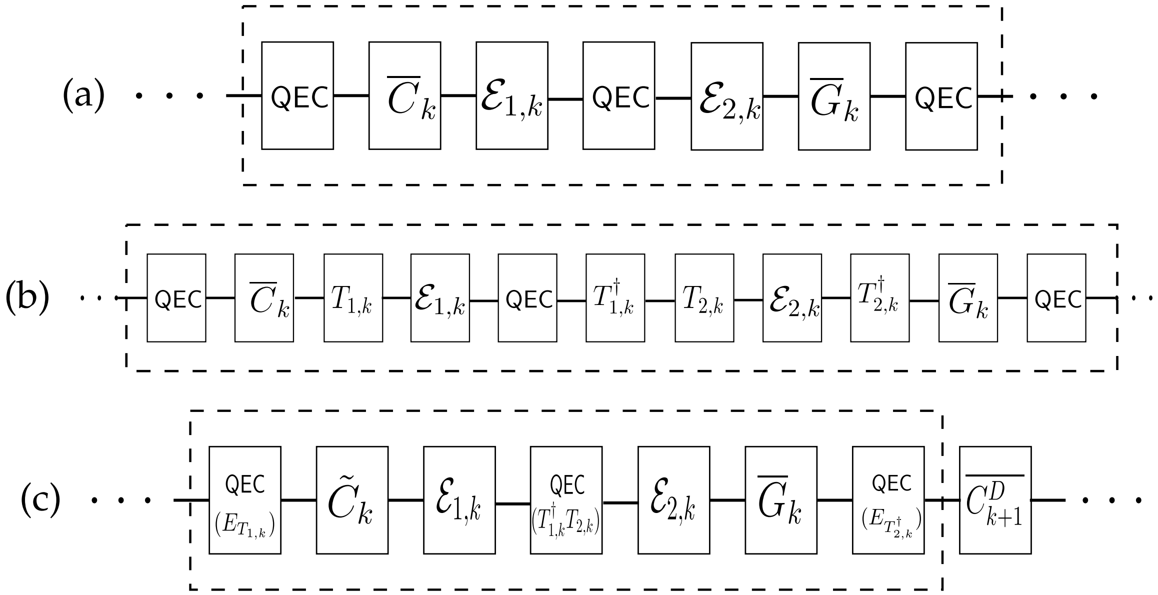


Figure 3.1: Compiling twirling (random physical Pauli) gates into fault tolerant gadgets. Figure (a) shows the noisy gates in the $k-$th clock cycle of a fault tolerant quantum algorithm and is an adaptation of the standard form prescribed in [WE16]. Twirling gates are inserted in figure (b) to tailor the noise processes to Pauli errors. These gates are compiled into existing gates by replacing easy gates by their dressed versions in figure (c).

3.1.1 Quantum error correction with RC

We now show how randomized compiling (RC) can be performed in fault tolerant circuits. Note that a Pauli error P can be decomposed with reference to a stabilizer code: $P = \overline{P}\ S_P\ E_P$, where S_P is an element of the stabilizer group $\mathcal{S}$, $\overline{P}$ is a logical Pauli operator in $\mathcal{L} = \mathcal{N}(\mathcal{S})/\mathcal{S}$, and E_P is an element of $\mathcal{N}(\mathcal{L})/\mathcal{S}$, usually called a pure error [LB13; Pou06]. Unlike pure errors, stabilizers and logical operators commute with QEC routines. A Pauli error P can be compiled into QEC resulting in a new quantum error correction routine $QEC(P)$ in which the input to the decoder corresponding to a syndrome outcome s is $s \oplus s(P)$ [DA07; CIP18].

In fault tolerant circuits, each logical gate is sandwiched between QEC routines. Following the prescription in [WE16], we divide logical gates into two sets: $\mathcal{S}_1$ and $\mathcal{S}_2$, calling them easy and hard gates respectively. A crucial requirement for $\mathcal{S}_1$ and $\mathcal{S}_2$ is

$$\overline{G}\ T\ \overline{G}^\dagger\ QEC = QEC(T)\ \overline{C} \tag{3.1}$$

for some easy logical gate $\overline{C} \in \mathcal{S}_1$, $n-$qubit Pauli gates T and all hard gates $\overline{G}$. Recall that $QEC(T)$ refers to the compilation of the Pauli gate T in the QEC routine, discussed in the background section. The previous requirement follows from

$$\overline{G}\; T\; \overline{G}^{\dagger}\; QEC = \overline{G}\; \overline{T}\; \overline{G}^{\dagger}\; S_T\; E_T\; QEC \tag{3.2}$$

$$= \overline{G}\; \overline{T}\; \overline{G}^{\dagger}\; QEC(E_T)\;, \tag{3.3}$$

where, in eq. 3.2 we have used the decomposition of Pauli gates with reference to a stabilizer code. Note that the expression $\overline{G}\;\overline{T}\;\overline{G}^{\dagger}$ in eq. 3.3 is guaranteed to be an easy gate for a choice of easy and hard gate sets in [WE16].

Fig. 3.1(a) shows a canonical presentation of a quantum circuit, where the k-th clock cycle is composed of an easy gate $\overline{C}_k$ and a hard gate $\overline{G}_k$, sandwiched between QEC routines. Noise processes affecting easy and hard gates are denoted by $\mathcal{E}_{1,k}$ and $\mathcal{E}_{2,k}$ respectively. These complex processes can be tailored to Pauli errors by inserting Pauli gates $T_{1,k}, T^{\dagger}_{1,k}, T_{2,k}, T^{\dagger}_{2,k}$. However, to guarantee that they be applied in a noiseless fashion, we compile them into the existing gates in the fault tolerant circuit. This is achieved in two steps. First, $T^{\dagger}_{1,k}$ and $T_{2,k}$ are compiled into QEC following $\mathcal{E}_{1,k}$, resulting in $QEC(T^{\dagger}_{1,k}T_{2,k})$. Second, $T^{\dagger}_{2,k}$ is propagated across $\overline{G}_k$, and compiled with $QEC\;\overline{C}_{k+1}T_{1,k+1}$, resulting in a *dressed gate* $\overline{C^{D}_{k+1}}$ such that $QEC(T^{\dagger}_{2,k})\overline{C^{D}_{k+1}} = \overline{G}_k\; T^{\dagger}_{2,k}\; \overline{G}^{\dagger}_k\; QEC\; \overline{C}_{k+1}T_{1,k+1}$. It follows from eq. 3.1 that $\overline{C^{D}_{k+1}}$ is equivalent to quantum error correction followed by an easy gate.

Fig. 3.1(c) shows the result of compiling all of the twirling gates into the easy gates and quantum error correction routines. Note that the compiled circuit is logically equivalent to the original circuit in the absence of noise. However, in the presence of noise, the average output of the circuit is dictated by the performance of $QEC(T)$ averaged over the different choices of Pauli gates T. This is what we refer to as QEC in the *RC setting*. In practice, this average performance can be achieved by repeating every iteration (shot) of the algorithm with a different Pauli operation compiled into the constituent QEC routines. With RC, the average logical performance of a QEC scheme over several compilations with random Pauli gates can be well approximated by the performance of the QEC scheme under an effective Pauli error model. For the purpose of numerical simulations in this thesis, we have used the performance of the QEC routine under the twirled noise process, as a proxy to the performance of QEC in the RC setting.

3.1.2 Logical estimator for concatenated codes

With a noise model described by Pauli errors, we first develop the background needed to define a *logical estimator* that can accurately predict the logical error rate. Uncorrectable errors cause the quantum error correction scheme to fail. We adopt the notation p_c to denote the total probability of correctable errors:

$$p_c = \sum_{E \in \mathcal{E}_C} \chi_{E,E} \, , \tag{3.4}$$

where $\mathcal{E}_C$ is the set of correctable errors discussed in section 2.5.2. We also use the notation p_u to denote the total probability of uncorrectable errors: $p_u = 1 - p_c$. It is easy to note that p_u is an upper bound to the standard infidelity metric which is measured by randomized benchmarking, i.e., $r = 1 - \chi_{0,0}$:

$$p_u = r - \sum_{\substack{E \in \mathcal{E}_C \\ E \neq \mathbb{I}}} \chi_{E,E} \, . \tag{3.5}$$

In particular, for Pauli noise processes the following equations show that p_u is exactly the average *logical* infidelity $\overline{r}$. This follows from the fact that the off-diagonal terms in the χ-matrix representation of Pauli noise processes i.e., $\chi_{E,E'} = 0$ in Eq.(3.8) for all $E \neq E'$.

$$\begin{aligned} \overline{r} &= 1 - \sum_{\substack{E,E' \in \mathcal{E}_C \\ s(E)=s(E')\,,\,\overline{E}=\overline{E}'}} \chi_{E,E'} && (3.6) \\ &= r - \sum_{\substack{E,E' \in \mathcal{E}_C\,,\,E,E' \neq \mathbb{I} \\ s(E)=s(E')\,,\,\overline{E}=\overline{E}'}} \chi_{E,E'} && (3.7) \\ &= p_u - \sum_{\substack{E,E' \in \mathcal{E}_C\,,\,E \neq E' \\ s(E)=s(E')\,,\,\overline{E}=\overline{E}'}} \chi_{E,E'} \, . && (3.8) \end{aligned}$$

A detailed derivation of eq. 3.6 is presented in section A.1 of the appendix. The expressions in eqs. 3.7 and 3.8 point out a conceptual difference between infidelity and the uncorrectable error probability. While on the one hand, r accounts for the effect of only the trivial correctable error $\mathbb{I}$, p_u on the other hand captures many more degrees of freedom – including all other correctable errors in $\mathcal{E}_C$. Hence, we expect r to be a worse predictor of the logical infidelity than p_u.

It is generally infeasible to enumerate all the $\mathcal{O}(4^{n-k})$ correctable errors for an $[[n,k]]$ stabilizer code; to compute p_u exactly. Our logical estimator is the result of an efficient heuristic to approximate p_u, particularly for concatenated code families. In particular, we use a coarse grained estimate of the probability of a syndrome outcome – a joint probability distribution over $\mathcal{O}(n^\ell)$ syndrome bits – calculated as a product of marginal probability distributions over the n code blocks at level $(\ell-1)$. This procedure is recursed through the ℓ levels of the concatenated code. Furthermore, its accuracy is provably high for uncorrelated Pauli error models.

While for concatenated codes, the number of physical qubits itself grows exponentially in the size of a code block n, we can exploit its encoding structure to simplify the complexity of computing p_u. However, it turns out that despite this simplification we cannot exactly compute p_u efficiently, i.e., in time that scales polynomially in the number of physical qubits. This leads us to resort to a heuristic method for a reasonable approximation of p_u for concatenated codes. Here we present a method to measure and compute an approximation, denoted by $\widetilde{p}_u(\mathcal{C}^\star_\ell)$, to the probability of uncorrectable errors for a concatenated code $\mathcal{C}^\star_\ell$: $p_u(\mathcal{C}^\star_\ell)$. For ease of notation we also define the quantities $p_c(\mathcal{C}^\star_\ell) = 1 - p_u(\mathcal{C}^\star_\ell)$ and $\widetilde{p}_c(\mathcal{C}^\star_\ell) = 1 - \widetilde{p}_u(\mathcal{C}^\star_\ell)$.

An error E_ℓ for the level ℓ concatenated code $\mathcal{C}^\star_\ell$ can be expressed as a tensor product of Pauli errors $E_{\ell-1,i}$ for the level $\ell-1$ codes $\mathcal{C}^\star_{\ell-1,i}$:

$$E_\ell = \bigotimes_{i=1}^{n} E_{\ell-1,i} \; . \tag{3.9}$$

Let us define E_ℓ to be a correctable pattern if the above tensor product corresponds to an encoded version of a correctable error for the code block $\mathcal{C}_\ell$. For example, $E_2 = \overline{X} \otimes \overline{\mathbb{I}}^{\otimes 6}$ is a correctable pattern for the $\ell = 2$ concatenated Steane code since $X \otimes \mathbb{I}^{\otimes 6}$ is a correctable error for the Steane code block.

A correctable error E_ℓ for the concatenated code $\mathcal{C}^\star_\ell$ is either (i) corrected within the lower level code-blocks $\mathcal{C}^\star_{\ell-1,1}, \ldots, \mathcal{C}^\star_{\ell-1,n}$, or (ii) has a non-trivial correction applied by the decoder of the level$-\ell$ code-block $\mathcal{C}_{\ell,1}$. Let us denote the contribution to $p_c(\mathcal{C}^\star_\ell) = 1 - p_u(\mathcal{C}^\star_\ell)$ from case (i) by Λ, while that from case (ii) by Γ; so that

$$p_c(\mathcal{C}^\star_\ell) = \Lambda(\mathcal{C}^\star_\ell) + \Gamma(\mathcal{C}^\star_\ell) \; . \tag{3.10}$$

Case (i) implies that each of the errors $E_{\ell-1,i}$ are correctable errors for the codes $\mathcal{C}^\star_{\ell-1,i}$. Therefore, the total probability of correctable errors in case (i) admits a recursive definition:

$$\Lambda(\mathcal{C}^\star_\ell) = p_c(\mathcal{C}^\star_{\ell-1,1}) p_c(\mathcal{C}^\star_{\ell-1,2}) \ldots p_c(\mathcal{C}^\star_{\ell-1,n}) \; . \tag{3.11}$$

Recall that case (ii) is the total probability of non-trivial correctable patterns for $\mathcal{C}^{\star}_{\ell}$, i.e.,

$$\Gamma(\mathcal{C}^{\star}_{\ell}) = \sum_{E \,\in\, \mathcal{E}_{\mathcal{C}} \backslash \mathbb{I}} \mathsf{Pr}(E_{\ell}) \;, \tag{3.12}$$

$$= \sum_{E \,\in\, \mathcal{E}_{\mathcal{C}} \backslash \mathbb{I}} \mathsf{Pr}(E_{\ell-1,1} \otimes E_{\ell-1,2} \otimes \ldots \otimes E_{\ell-1,n}) \tag{3.13}$$

where we have used the fact that each correctable error corresponds to a pattern according to eq. 3.9. A logical error $E_{\ell-1,i}$ occurs on the code-block $\mathcal{C}_{\ell-1,i}$ whenever the decoder fails in correcting the physical errors in such a way that the residual effect of the physical noise process affecting the qubits of $\mathcal{C}^{\star}_{\ell-1,i}$ and the recovery operation applied by the decoder results in $E_{\ell-1,i}$. Let us denote the probability of the decoder for $\mathcal{C}^{\star}_{\ell-1,i}$ to leave a residual $E_{\ell-1,i}$, conditioned on the syndrome measurements by $\mathsf{Pr}_{\mathcal{D}}(E_{\ell-1,i} \,|\, s(\mathcal{C}^{\star}_{\ell-1,i}))$. We can rewrite eq. 3.13 as

$$\Gamma(\mathcal{C}^{\star}_{\ell}) = \sum_{E \,\in\, \mathcal{E}_{\mathcal{C}} \backslash \mathbb{I}} \; \sum_{s(\mathcal{C}^{\star}_{\ell})} \mathsf{Pr}(s(\mathcal{C}_{\ell}) s(\mathcal{C}^{\star}_{\ell-1,1}) \ldots s(\mathcal{C}^{\star}_{\ell-1,n})) \prod_{j=1}^{n} \mathsf{Pr}_{\mathcal{D}}(E_{\ell-1,j} | s(\mathcal{C}^{\star}_{\ell-1,j})) \;, \tag{3.14}$$

$$= \sum_{E \,\in\, \mathcal{E}_{\mathcal{C}} \backslash \mathbb{I}} \; \sum_{s(\mathcal{C}^{\star}_{\ell})} \mathsf{Pr}(s(\mathcal{C}_{\ell}) | s(\mathcal{C}^{\star}_{\ell-1,1}) \ldots s(\mathcal{C}^{\star}_{\ell-1,n}))$$

$$\prod_{j=1}^{n} \mathsf{Pr}_{\mathcal{D}}(E_{\ell-1,j} | s(\mathcal{C}^{\star}_{\ell-1,j})) \mathsf{Pr}(s(\mathcal{C}^{\star}_{\ell-1,j})) \;, \tag{3.15}$$

where $\mathsf{Pr}(s(\mathcal{C}_{\ell}) | s(\mathcal{C}^{\star}_{\ell-1,1}) \ldots s(\mathcal{C}^{\star}_{\ell-1,n}))$, is the conditional probability of measuring the syndrome outcomes $s(\mathcal{C}_{\ell})$ on the code-block $\mathcal{C}_{\ell}$ when the outcomes on the lower level code blocks $\mathcal{C}^{\star}_{\ell-1,1}, \ldots, \mathcal{C}^{\star}_{\ell-1,n}$ are $s(\mathcal{C}^{\star}_{\ell-1,1}), \ldots, s(\mathcal{C}^{\star}_{\ell-1,n})$, respectively. Equivalently,

$$\mathsf{Pr}(s(\mathcal{C}_{\ell}) | s(\mathcal{C}^{\star}_{\ell-1,1}) \ldots s(\mathcal{C}^{\star}_{\ell-1,n})) = \mathsf{Pr}(s(\mathcal{C}_{\ell}) | \mathcal{E}^{s(\mathcal{C}^{\star}_{\ell-1,1})}_{\ell-1,1} \ldots \mathcal{E}^{s(\mathcal{C}^{\star}_{\ell-1,n})}_{\ell-1,n})) \;. \tag{3.16}$$

A major hurdle in computing Γ using eq. 3.15 is the sum over an exponentially large set of syndrome outcomes for the concatenated code. To circumvent this difficultly, we will apply an efficient heuristic to approximate the probability in eq. 3.16. In essence, we will replace the conditional channel $\mathcal{E}^{s(\mathcal{C}^{\star}_{\ell-1,i})}_{\ell-1,i}$ by the average logical channel $\hat{\mathcal{E}}_{\ell-1,i}$, which is defined as

$$\hat{\mathcal{E}}_{\ell-1,i} = \sum_{s(\mathcal{C}_{\ell-1,i})} \mathsf{Pr}(s(\mathcal{C}_{\ell-1,i})) \mathcal{E}^{s(\mathcal{C}_{\ell-1,i})}_{\ell-1,i} \left[\hat{\mathcal{E}}_{\ell-2,1} \otimes \ldots \otimes \hat{\mathcal{E}}_{\ell-2,n} \right] \;. \tag{3.17}$$

Note that $\hat{\mathcal{E}}_{0,j}$ is the physical noise model while $\hat{\mathcal{E}}_{1,j}$ is the exact average logical channel $\overline{\mathcal{E}}_{1,j}$. However, in general for $\ell \geq 2$, $\hat{\mathcal{E}}_\ell$ is a coarse-grained approximation for the exact average logical channel $\overline{\mathcal{E}}_\ell$. In other words, $\hat{\mathcal{E}}_{\ell-1,i}$ is computed using the knowledge of the syndrome bits measured only at level $\ell - 1$, while assuming the noise model: $\hat{\mathcal{E}}_{\ell-2,1} \otimes \ldots \otimes \hat{\mathcal{E}}_{\ell-2,n}$, that accounts for the average effect of all syndrome measurements at lower levels.

Replacing the conditional channel $\mathcal{E}^{s(\mathcal{C}^\star_{\ell-1,i})}_{\ell-1,i}$ in eq. 3.15 by the average channel $\hat{\mathcal{E}}_{\ell-1,i}$ defined in eq. 3.17 allows us to approximate Γ by $\widetilde{\Gamma}$ defined as follows:

$$\widetilde{\Gamma}(\mathcal{C}^\star_\ell) = \sum_{E\,\in\,\mathcal{E}_\mathcal{C}\setminus\mathbb{I}} \sum_{s(\mathcal{C}_\ell)} \sum_{s(\mathcal{C}^\star_{\ell-1,1})} \cdots \sum_{s(\mathcal{C}^\star_{\ell-1,n})} \mathsf{Pr}(s(\mathcal{C}_\ell)|\hat{\mathcal{E}}_{\ell-1,1}\ldots\hat{\mathcal{E}}_{\ell-1,n}) \prod_{j=1}^{n} \mathsf{Pr}_\mathcal{D}(E_{\ell-1,j}\,|\,\hat{\mathcal{E}}_{\ell-1,j})\mathsf{Pr}(s(\mathcal{C}^\star_{\ell-1,j}))\;, \tag{3.18}$$

$$= \sum_{E\,\in\,\mathcal{E}_\mathcal{C}\setminus\mathbb{I}} \prod_{j=1}^{n} \mathsf{Pr}_\mathcal{D}(E_{\ell-1,j}\,|\,\hat{\mathcal{E}}_{\ell-1,j}))\;. \tag{3.19}$$

Denote $\mathcal{R}(s(\mathcal{C}_{\ell-1,i}), P)$ to be the set of $n-$qubit errors on which a lookup table decoder for the code block $\mathcal{C}_{\ell-1,i}$ leaves a residual logical error P when the error syndrome $s(\mathcal{C}_{\ell-1,i})$ is encountered. Now $\mathsf{Pr}_\mathcal{D}(E_{\ell-1,i}\,|\,\hat{\mathcal{E}}_{\ell-1,j}))$ can be computed recursively:

$$\mathsf{Pr}_\mathcal{D}(E_{\ell-1,i}\,|\,\hat{\mathcal{E}}_{\ell-1,i})) = \sum_{Q\in\mathcal{R}(s(\mathcal{C}_{\ell-1,i}),E_{\ell-1,i})} \prod_{j=1}^{n} \mathsf{Pr}_\mathcal{D}(Q_{\ell-2,j}\,|\,\hat{\mathcal{E}}_{\ell-2,j}))\;. \tag{3.20}$$

Note that the probability of leaving a residual error at level 0 is simply specified by the physical noise model, i.e., $\mathsf{Pr}_\mathcal{D}(P|\hat{\mathcal{E}}_{0,j})$ is the probability of the Pauli error P on the physical qubit j. This concludes the method to efficiently compute $\widetilde{\Gamma}$, an approximation to Γ.

Recall that the total probability of correctable errors is given by eq. 3.10. An approximation to $p_c(\mathcal{C}^\star_\ell)$, is given by

$$\widetilde{p}_c(\mathcal{C}^\star_\ell) = \widetilde{\Lambda}(\mathcal{C}^\star_\ell) + \widetilde{\Gamma}(\mathcal{C}^\star_\ell)\;, \tag{3.21}$$

where $\widetilde{\Gamma}$ defined in eq. 3.19 while $\widetilde{\Lambda}$ is defined in a similar fashion to eq. 3.11:

$$\widetilde{\Lambda}(\mathcal{C}^\star_\ell) = \widetilde{p}_c(\mathcal{C}^\star_{\ell-1,1})\widetilde{p}_c(\mathcal{C}^\star_{\ell-1,2})\ldots\widetilde{p}_c(\mathcal{C}^\star_{\ell-1,n})\;. \tag{3.22}$$

Using the approximation in eq. 3.21, we can efficiently estimate the logical estimator $\widetilde{p}_u$ for concatenated codes. We now summarize the procedure to calculate the logical estimator.

Protocol 2 Logical estimator for concatenated codes

Input: A level ℓ-concatenated code $\mathcal{C}^\star_\ell$ with constituent codes $\mathcal{C}_i$ and corresponding correctable error sets $\mathcal{E}_{\mathcal{C}_i}$ for each level $1 \leq i \leq l$; Probabilities of Pauli errors corresponding to the twirl of the noise $\mathcal{E}_{0,j}$ acting on each code block $1 \leq j \leq N/n_{\mathcal{C}_1}$ i.e., diagonal entries of $\chi(\mathcal{E}_{0,j})$ obtained from Cycle Error Reconstruction experiments, where $n_{\mathcal{C}_i}$ denotes the number of physical qubits used to encode one logical qubit using the code $\mathcal{C}_i$ and $N = \Pi_{k=1}^{\ell} n_{\mathcal{C}_k}$ is the total number of physical qubits.

Output: The logical estimator - $\widetilde{p}_u(\mathcal{C}^\star_\ell)$.

Procedure:

```
// $\hat{\mathcal{E}}_{i,j}$ - Average logical channel at level $i$, code-block $j$
// Iterate over all the levels of the code.
for $i \leftarrow 1$ to $\ell$ do
    // Iterate over all the code blocks in each level.
    for $j \leftarrow 1$ to $N/(\Pi_{k=1}^{i} n_{\mathcal{C}_k})$ do
        if $i == 1$ then
            // For the first level, use Cycle Error Reconstruction data.
            $\chi(\hat{\mathcal{E}}) \leftarrow \chi(\mathcal{E}_{0,j})$
        else
            // Use previous level's average logical channels.
            $\chi(\hat{\mathcal{E}}) \leftarrow \otimes_{\text{ind}=1}^{n_{\mathcal{C}_i}} \chi(\hat{\mathcal{E}}_{i-1,(j-1)*n_{\mathcal{C}_i}+\text{ind}})$
        end
        // Calculate the average logical channel for each code block.
        for $m \leftarrow 0$ to $3$ do
            $\chi_{m,m}(\hat{\mathcal{E}}_{i,j}) \leftarrow \sum_{\substack{E,E' \in \mathcal{E}_{\mathcal{C}_i} \\ s(E)=s(E'),\, \overline{E}=\overline{E}'}} \phi(E,m)\, \phi^\star(E',m)\, \chi_{E\overline{P}_m,\overline{P}_m E'}(\hat{\mathcal{E}})$,
            where $R_{s(E)}$ is the recovery operator for the syndrome $s(E)$; $R_{s(E)}\ |E\ \overline{P}_m| =$
              $\phi(E,m)\ S\ |\overline{P}_m|$ for any Pauli error $E$ and some stabilizer $S$; $|P|$ denotes the
              bare Pauli without any associated global phase; $\overline{P}$ is the logical component.
        end
    end
end
$\widetilde{p}_u(\mathcal{C}^\star_\ell) \leftarrow 1 - \chi_{0,0}(\hat{\mathcal{E}}_{\ell,1})$
```

For i.i.d Pauli error models with sufficiently small single-qubit infidelity r_0, the quality of approximation is: $|\overline{r}_\ell - \widetilde{p}_u| \leq n_C^{\ell+1} r_0^{2+\lfloor (d_C+1)/2 \rfloor}$. Here, d_C and n_C describe the distance and the size of a code-block of a level$-\ell$ concatenated code. For instance, using an i.i.d depolarizing error model with $r_0 = 10^{-3}$ and the level-2 concatenated Steane code, the above expression yields $|\overline{r}_2 - \widetilde{p}_u| \leq 5 \times 10^{-10}$. This is validated by numerics: $\widetilde{p}_u = 4.24 \times 10^{-9}$ and $\overline{r}_2 = 4.20 \times 10^{-9}$. A detailed derivation of quality of approximation is provided in section A.2 of the appendix.

Notably, the time complexity of computing $\widetilde{p}_u$ for the concatenated code: $\mathcal{O}(4^{n_C+\ell}\, n^\ell)$, scales polynomially in the total number of physical qubits n^ℓ, whereas an exact computation of p_u would scale doubly exponentially in ℓ. We will now prove this statement.

Recall that $\widetilde{p}_u = 1 - \widetilde{p}_c(\mathcal{C}_\ell^\star)$, where $\widetilde{p}_c(\mathcal{C}_\ell^\star)$ is an approximation to the total probability of correctable errors. Note that $\widetilde{p}_c(\mathcal{C}_\ell^\star) = \widetilde{\Lambda} + \widetilde{\Gamma}$ where both $\widetilde{\Lambda}$ and $\widetilde{\Gamma}$ are defined recursively. So, if computing $\widetilde{p}_c(\mathcal{C}_\ell^\star)$ takes time τ_ℓ and computing $\widetilde{\Gamma}$ takes time κ_ℓ, we have

$$\tau_\ell = n\, \tau_{\ell-1} + \kappa_\ell \,. \tag{3.23}$$

The recurrence relation in eq. 3.19 for computing $\widetilde{\Gamma}(\mathcal{C}_\ell^\star)$ implies

$$\begin{aligned} \kappa_\ell &= 4n\, \kappa_{\ell-1} + \mathcal{O}(4^n)\,, & (3.24)\\ &= \mathcal{O}(4^{n+\ell}\, n^\ell)\,. & (3.25) \end{aligned}$$

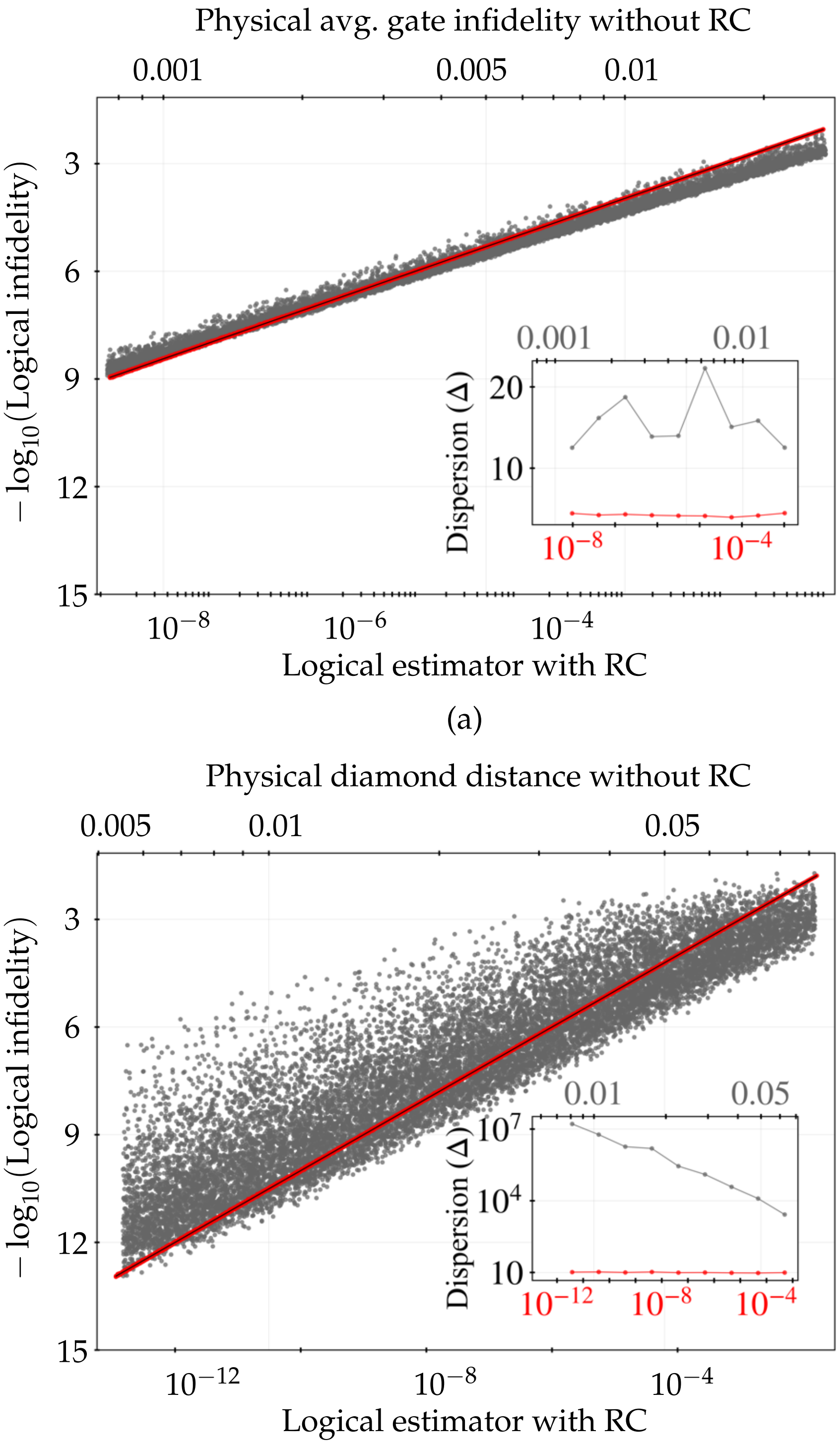
Using the above solution in eq. 3.23, we find that

$$\tau_\ell = \mathcal{O}(4^{n+\ell}\, n^\ell)\,. \tag{3.26}$$

The last equation establishes that computing logical estimator is linear in the total number of qubits i.e, n^ℓ.

3.2 Results and discussion

We provide numerical evidence to highlight the improvement offered by our methods developed for optimizing FT schemes. We begin with the task of accurately predicting the performance of concatenated Steane codes. We perform numerical simulations of QEC in the RC and non-RC settings under a large ensemble of random CPTP maps applied to the physical qubits. Following Ref. [IP18], we generate a single qubit CPTP



(a)

(b)

Figure 3.2: The figure compares the predictive power of the (exact) logical estimator (red) against two standard error metrics (gray): the average gate infidelity (a) and the diamond distance (b), under a large ensemble of CPTP maps. Each point $p = (x_p, y_p)$ corresponds to a noise process; x_p is its physical error metric and y_p, its logical error rate. The dispersion quantified as Δ in the insets indicates the predictive power of the physical metric. While the logical error rates can vary over several orders of magnitude for standard error metrics, logical estimator is strongly correlated with the logical error rate.

map $\mathcal{E}$ from its Stinespring dilation: a random unitary matrix U of size (8×8), given by $U = e^{-iHt}$ for a complex Hermitian matrix H whose entries are sampled from a Gaussian distribution of unit variance, centred at 0. We vary the time parameter t between 0.001 and 0.1 to vary the noise strength.

Figure 3.2 shows that logical error rates can vary wildly across physical noise processes with fixed infidelity and diamond distance in agreement with [IP18]. The variation, captured by the amount of dispersion in the scatter plots, is quantified using the ratio of the maximum and the minimum logical error rates across channels of similar physical error rate, denoted by Δ. In other words, we partition the range of physical error rates into bins b_i and use $\Delta(b_i)$ to quantify the amount of dispersion: $\Delta(b_i) = (1/|b_i|)\ (\max_{p\in b_i} y_p)/(\min_{p\in b_i} y_p)$, where $|b_i|$ is the number of channels in the bin b_i. The large fluctuations in the logical error rates can be attributed to two extreme features of the error-metrics. While infidelity controls only one parameter out of the many that specify a noise process, diamond distance suffers from being sensitive to the details of a noise process that are irrelevant to the logical error rate. In addition, standard error metrics can only reveal intrinsic properties of the underlying noise process that are agnostic to the choice of an error correcting code.

Logical estimator with RC, in contrast, is highly correlated with the logical error rate. This improvement can be attributed to two features. First, RC provides a drastic reduction from $\mathcal{O}(12^n)$ parameters that specify an $n-$qubit Markovian noise process to $\mathcal{O}(4^n)$ Pauli error probabilities. Second, unlike standard error metrics, $\widetilde{p}_u$ carefully accounts for Pauli error probabilities that contribute to the logical error rate. Numerical evidence for drastic gains in predictability using the logical estimator with RC for the class of coherent errors is presented in section A.3 of the appendix.

The special setting of i.i.d noise hides the drastic advantages provided by $\widetilde{p}_u$ in predicting logical infidelity because the dominant contribution to $\widetilde{p}_u$ comes from $\chi_{0,0}$, which is also well captured by r. However, for correlated error-models, given only $\chi_{0,0}$, the uncertainty on the logical error rate ranges between extremities 0 and 1. While r is completely insensitive to either of these scenarios, $\widetilde{p}_u$ in contrast helps distinguish between them, thereby providing a far more accurate estimate of the logical error rate.

We support the above argument by numerical studies of correlated Pauli error models generated from a convex combination of an i.i.d process of infidelity r_0 and multi-qubit interactions. While the i.i.d component $\mathcal{E}_{\mathsf{iid}}$ is specified by single qubit error probabilities, multi-qubit interactions are specified by an arbitrary subset $\mathfrak{S}$, so, $\mathcal{E}_{\mathsf{cor}}(\rho) = \sum_{P\in\mathfrak{S}} \chi_{P,P} P\,\rho\,P$, where $\chi_{P,P}$ is sampled from the normal distribution with mean and variance $4^n r_0$. The combined Pauli error model is therefore given by $\mathcal{E}(\rho) = q\mathcal{E}_{\mathsf{iid}}(\rho) +$

$(1-q)\mathcal{E}_{\mathsf{cor}}(\rho)$, where $0 \leq q \leq 1$. Explicitly setting $\chi_{0,0}$ followed by appropriate normalization, ensures that the infidelity of the above noise model is r_0. Figure 3.3 highlights the importance of the $\widetilde{p}_u$ over r for predicting the performance of the concatenated Steane code under correlated Pauli noise processes.

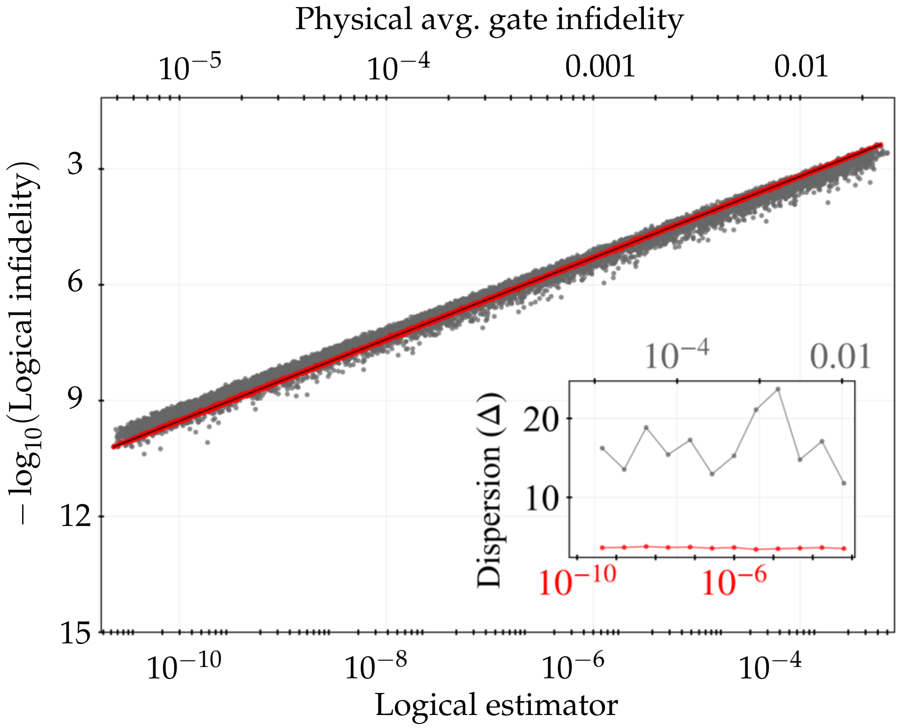


Figure 3.3: Predictability of logical infidelity for level-2 concatenated Steane code. The figure compares the predictive powers of logical estimator (red) against two standard error metrics (gray): the average gate infidelity (a) and the diamond distance (b), under correlated Pauli maps. Each point $p = (x_p, y_p)$ corresponds to a noise process; x_p is its physical error metric and y_p, its logical error rate. The dispersion quantified as Δ in the insets indicates the predictive power of the physical metric. While the logical error rates can vary over several orders of magnitude for standard error metrics, logical estimator is strongly correlated with the logical error rate.

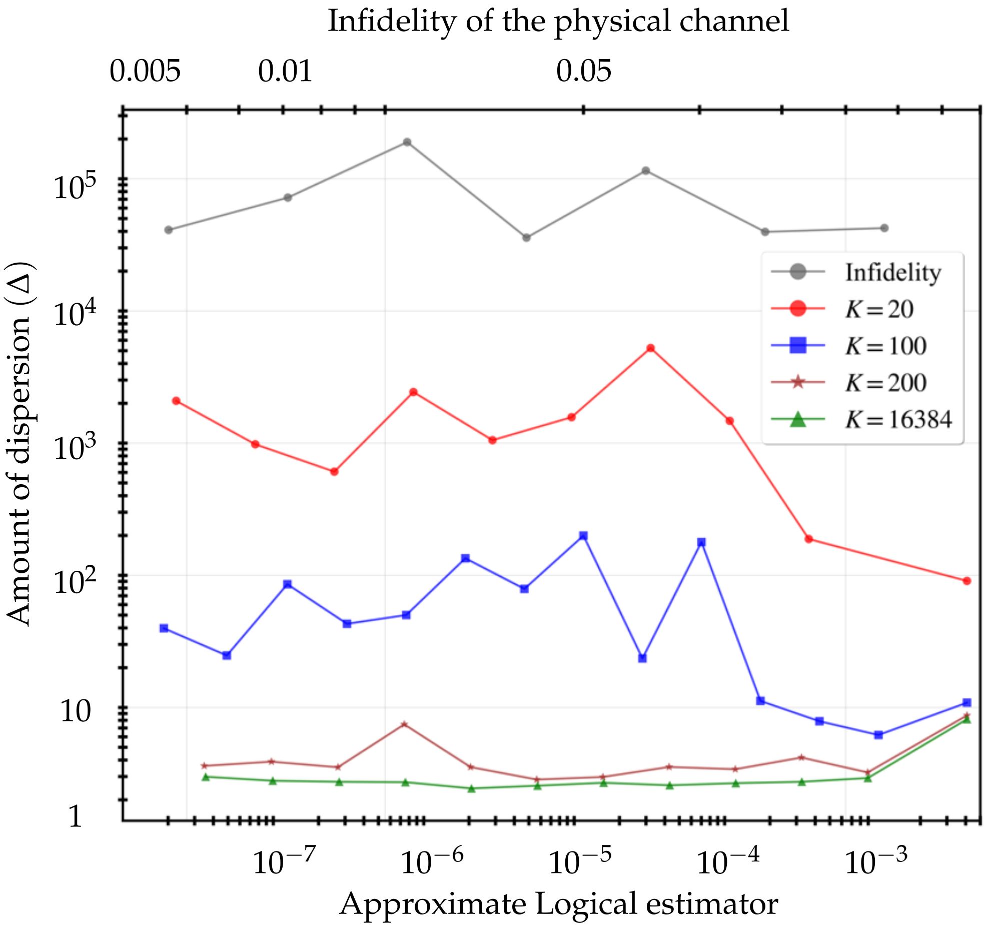


Figure 3.4: Accuracy of the logical estimator based on limited CER data, using a level-2 concatenated Steane code for an ensemble of about 15000 random correlated Pauli channels. The accuracy, quantified by Δ, improves sharply with the number of Pauli error rates (K) extracted using CER. We observe that for $K = 200$, which is about 1.2% of all Pauli error rates on the Steane code block, the accuracy closely matches the logical estimator computed using all CER data, i.e., $K = 4^7$.

3.2.1 Limited cycle error reconstruction data

Even in the absence of correlations across the $n-$qubit code blocks of a concatenated code, we require $\mathcal{O}(4^n)$ Pauli error rates from CER to compute $\widetilde{p}_u$. Extracting this exponential sized CER data set is a challenge for experimentalists. Refs. [HYF21; CDDH⁺23] describe how to extract the leading K Pauli error probabilities in a noise process, where

$K \ll 4^n$. We want to combine a handful of leading Pauli error rates extracted by CER with a simple method to extrapolate the remaining ones. For a Pauli error Q that is not given in the CER dataset we set

$$\mathsf{Pr}(Q) = (1 - r_0)^{n - \mathsf{wt}(Q)}\,(r_0/3)^{\mathsf{wt}(Q)}\ , \tag{3.27}$$

where $\mathsf{wt}(Q)$ is the Hamming weight of Q, and r_0 is derived from the infidelity of the noise process: $r = 1 - (1 - r_0)^n$. We construct an adversarial error model where the above extrapolation is unlikely to perform well by setting some multi-qubit error probabilities that violate eq. (3.27). Furthermore, when errors are sampled uniformly from the set of correctable and uncorrectable errors, we observe maximum fluctuations in the logical error rate. However, Fig. 3.4 presents strong numerical evidence indicating that the simple extrapolation works well in practice even for the adversarial example.

3.2.2 Code selection

Selecting a quantum error correcting code that has the smallest logical error rate under an existing physical noise process is a crucial step in optimizing resources for fault tolerance. To demonstrate the efficacy of the logical estimator for this problem, we consider an example of an error model and two different error-correcting codes: (i) concatenated Steane code and (ii) concatenated version of a $[[7, 1, 3]]$ code used in Ref. [RGB+17] that we refer to as a *Cyclic code*. The error model is obtained from a Pauli twirl on the i.i.d application of the CPTP map $\mathcal{E} : \rho \mapsto p_I \rho + \sum_{Q \in \{X,Y,Z\}} p_Q e^{-i\theta Q} \rho e^{i\theta Q}$, where $p_X = r_X(1 - r_Z), p_Z = r_Z(1 - r_X), p_Y = r_X r_Z, p_I = 1 - p_X - p_Y - p_Z$ and set a bias specified by $\eta = r_Z/r_X$. Based on Ref. [RGB+17], we expect the Steane code to outperform the Cyclic code in one noise regime, and the converse in a different regime. Our tool is successful if it produces a lower value of $\widetilde{p}_u$ for the code with lower logical infidelity, for any noise rate. Lastly, to compute the logical estimator as well as the logical error rate estimates, we use a bias-adapted minimum-weight decoder that assigns weights η, η, and 1 to each Pauli error of type X, Y, and Z, respectively.

Figure 3.5 shows that the logical estimator correctly identifies the optimal code for all values of the physical error rate (bias). It also replicates the functional form of the logical error rate, showing that the performance gain from the Cyclic code over the Steane code increases with the bias.

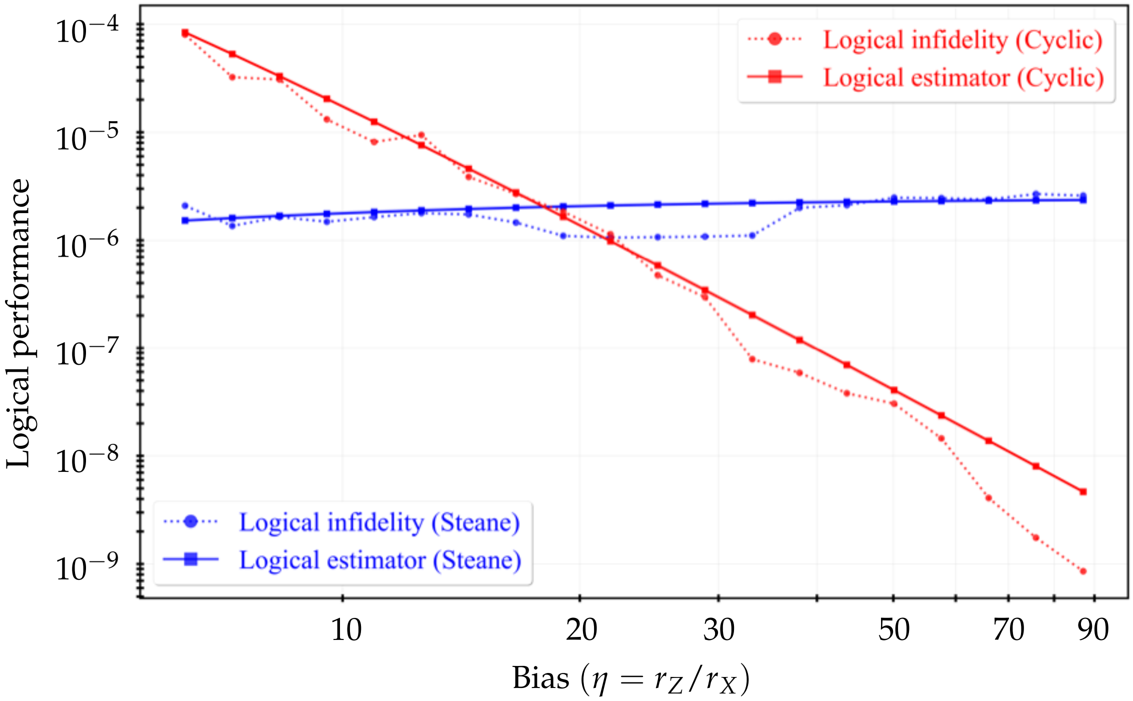


Figure 3.5: Using the logical estimator to select an optimal code. The above figure demonstrates the use of our tool in selecting an optimal error correcting code under a biased Pauli error model. The choices of codes include level-3 concatenated versions of the Steane code and the cyclic code. While the solid lines depict the values of the logical estimator, the dashed lines correspond to logical error rates estimated using numerical simulations. We observe that $\widetilde{p}_u$ accurately selects the optimal code for all noise rates.

3.3 Conclusion

We have shown how experimental data from CER, even limited data, can be used to successfully predict the logical performance of FT architectures based on concatenated codes. It can be used to precisely and efficiently estimate the resource overhead required to achieve a target logical error rate [JRO+17; RGB+17; NDD+19] for implementing quantum algorithms. Along with informing the choice of an optimal code for an underlying physical noise process, the logical estimator provides directives for other components in a FT scheme, such as a decoder. Different lookup table decoders can be compared using our logical estimator, similar to the work in Refs. [CR18a; SJG20; DPM+20].

Our scheme relies on RC to yield a Pauli error model, and although in theory this requires twirling with the full Pauli group, it has been observed that a handful of random compilations of the original circuit are sufficient in practice [HNM+21; SLS+21]. A natural question that follows is whether RC also mitigates the impact of physical noise on the logical qubit. There is no persistent trend across the general class of Markovian noise processes, and in some cases, RC degrades the performance of the code. Developing noise tailoring techniques that guarantee an improvement to the performance as well as predictability is an interesting problem for future research.

Although the methods and techniques presented in the chapter address generic noise processes, there are a number of roadblocks in broadening the scope of this study beyond concatenated codes, where the complexity of computing the logical estimator grows exponentially with the size of the code. We have proposed an application for surface codes in section A.4 of the appendix. While these results are preliminary, they demonstrate that our method may find broader application beyond concatenated codes. Also, further research is needed to extend these ideas to the context of multiple logical qubits.

Chapter 4

Improved quantum error correction with randomized compiling

> This chapter consists of a literal transcription of [JIB+23] for which my contribution was major. Some notation and stylistic changes have been done to be consistent with the rest of thesis.

Noise is pervasive in present-day quantum computation. The theory of fault tolerance was developed to guarantee reliable computations in the presence of noise. However, fault tolerant constructions demand a large overhead in terms of additional resources required to encode a logical computation in a way that is resilient to errors. Achieving the target logical error rates as required by various applications with the limited amount of resources in terms of the number of physical qubits is a challenging task. Along with designing better error correcting codes, decoders and high quality hardware components of a quantum computer, there are other ways of reducing logical error rates. Active noise tailoring by randomized compiling (RC) [WE16] is a potential candidate for two key reasons. First, RC significantly simplifies the form of the noise on the encoded quantum information. Second, RC can be used to transform an unknown error model into one that is adapted to the error correction capabilities of a particular code.

Randomized compiling tools were leveraged to accurately predict the performance of quantum error correction schemes in Ref. [IJB+22]. Although simplifying the form of the noise makes the performance more predictable, it was observed that RC can sometimes degrade the performance of an error correcting code. We can understand this effect by using the χ-representation [WBC15] of a physical noise process. In this representation, the action of noise on a quantum state ρ is given by: $\mathcal{E}(\rho) = \sum_{i,j} \chi_{i,j} P_i \rho P_j$ where P_i denote Pauli matrices in the $n-$qubit Pauli group $\mathcal{P}_n$ without phases, i.e., $P_i \in \mathcal{P}n/\{\pm 1, \pm i\}$. Noise tailoring methods such as RC can transform the elements of the χ-matrix, for example by removing off-diagonal elements $\chi_{i,j}\ \forall\ i \neq j$. This mathematical transformation is commonly referred to as twirling [BDS+96; BBP+96; CB19]. If one were to remove the contribution of $\chi_{i,j}$ corresponding to Pauli errors that are correctable by the decoder, this could have a negative impact of the code's performance. In general, noise tailoring methods are oblivious to the details of what error terms are relevant for quantum error correction.

Related work. The impact of twirling the noise on the performance of error correction schemes has been explored in the literature under various settings. The performance of surface codes under coherent and incoherent error models have been compared in Ref. [BEK+18], and using numerical studies it was noted that while the threshold is similar in both cases, the sub-threshold performance of the twirled channel is significantly better than the original coherent error model. In another setting, analytical calculations of the logical error rate of repetition codes under rotation errors reveal that coherent errors can accumulate faster, leading to worse logical error rates than their corresponding Pauli approximations [GD17]. The necessity of active coherence-suppression methods for codes with large distances was also noted, but their impact on the code's performance was not explored. For the Toric code under coherent error models, a laborious analysis has shown that the effective logical channel approaches an incoherent channel provided the noise decreases with increasing code size [IP20]. However, in the scenario where the error rate remains constant independent of the code size, there are several challenges to arriving at a similar conclusion. A recent study computed thresholds for the surface code under coherent rotations by mapping the problem first to a (complex) Ising model, and then to a corresponding scattering network [VBB22]. The error measure they used for maximum likelihood decoding measures how close the logical channel is to a Pauli channel (regardless of the decoder used). They discovered that for rotation angles below the threshold, the logical channel approaches a Pauli channel at a rate that scales exponentially with the code distance. In Ref. [GSL+16], the poor predictability of the logical error rate and the code's pseudo threshold under coherent errors provided by their twirled counterparts was identified, reinforcing the need for active noise tailoring. The impact of twirling the noise for complex error models, such as combinations of stochastic errors and rotations around an arbitrary non-Pauli axis, is unknown. The scaling of the potential gains from twirling with increased code-concatenation levels remains unexplored.

Our contributions. In this chapter, we analyze the impact of RC on the performance of quantum error correction. In particular, we show that RC improves the performance of a concatenated Steane code under a coherent noise model (specifically, a tensor product of arbitrary identical unitary errors). This positive result demonstrates that RC tools can play a key role in achieving fault tolerance. We present a detailed study of the performance gains with respect to changes in the axis of rotation and the number of levels of concatenation. We identify a special axis of rotation for a given concatenation level where maximum gains from RC are achieved. We note that this axis can be different from the axes of rotation for which the best pseudo-threshold for the code is achieved. It has been observed, in previous studies, that randomized compiling can also degrade logical performance [BWG+18]. Our study shows that a wide class of physically motivated error models do not exhibit such behaviour. However, we identify some complex noise models where such degradation can occur and provide numerical results for the same.

The chapter is structured as follows. Section 4.1 discusses the methods used to study the impact of randomized compiling on the logical performance. In section 4.2, we present analytical studies for gains offered by randomized compiling using realistic error models. Finally, in section 4.3 we provide concluding remarks and describe some interesting open problems.

4.1 Methods

The goal of this chapter is twofold. First, we want to identify important scenarios for physical errors wherein RC can be leveraged to improve the performance of quantum error correcting codes. Second, identify settings under which such performance gains cannot be guaranteed. For the first goal, we study the performance of concatenated Steane code under realistic error models. We start off by simple rotations about $Z-$axis and progressively move to arbitrary rotations followed by a combination of coherent and stochastic error models. For the second goal, we generate numerical results for a large ensemble of noise processes belonging to more complex noise models which involve random rotations on different qubits and arbitrary CPTP maps. All the performance metrics in this chapter are derived in the memory model and assume perfect syndrome extraction. Simulations with gate dependent errors can be pursued in the future.

For both the goals, it is crucial to understand how RC can be applied alongside quantum error correction in practice. We follow the methods of Ref. [IJB+22]. The main idea can be summarized as follows. Recall that noise tailoring by randomized compiling is achieved by inserting random Pauli gates in a circuit such that its net effect does not change the logical output of the circuit. Consequently, the average output distribution of the circuit over all possible Pauli random gates can be understood by studying the response of the original circuit against Pauli noise on the individual components. In the same spirit, we insert random Pauli gates around all the individual components of a quantum error correction circuit. There is no need to account for sources of noise in the extra Pauli random gates because they can be absorbed into the original elements of the quantum error correcting circuit. In practice, only a handful compilations are sufficient to achieve the twirling effect [HNM+21]. We assume an ideal application of RC in this chapter for simplicity. The details of this procedure are described in section 3.1.1.

We now have two variations of the average fidelity. First, the standard notion – average fidelity over all syndrome outcomes, $r(\overline{\mathcal{E}}_1)$, defined in Eq. (2.111). Second, the average fidelity over syndrome outcomes as well as logically equivalent compilations of the quantum error correction circuit, which we will denote $r_{\rm rc}$. Note that the number of random compilations for a circuit with n elements grows as $\mathcal{O}(4^n)$. In the ideal case, where we have considered all of these compilations in $r_{\rm rc}(\overline{\mathcal{E}}_1)$, it reduces to $r(\overline{\mathcal{E}}_1^T)$.

While Eq. (2.111) addresses the logical channel of a block code, we can easily extend these definitions for a concatenated code assuming a hard decoder [CWB+17; GSL+16]. In this case, the logical channel at level$-\ell$ can be recursively defined in as a function whose input physical channels are the logical channels at level$-(\ell-1)$. We will use the notation $r(\overline{\mathcal{E}}_\ell)$ and $r(\overline{\mathcal{E}}_\ell^T)$ to denote the error rates corresponding to logical channels of a level$-\ell$ concatenated code without RC and with RC, respectively. Their ratio, denoted by δ_ℓ, where

$$\delta_\ell = \frac{r(\overline{\mathcal{E}}_\ell)}{r(\overline{\mathcal{E}^T}_\ell)} \,, \tag{4.1}$$

is an indicator of the performance gain due to RC, which we will estimate for various error models. Note that $\delta_\ell > 1$ indicates a performance gain whereas $\delta_\ell < 1$ denotes a performance loss.

4.2 Results and discussion

This section is devoted to case studies of performance gains from RC for the concatenated Steane code, under various interesting classes of error models, and inferences we can draw from these studies. Markovian errors can be broadly classified into unital and non-unital maps. Since non-unital components of a noise map do not impact the error rate significantly [Wal15; GD17], we restrict our attention to unital maps in this chapter. In particular, we choose coherent rotations which form an important class of unital maps. In practice, these typically arise from imperfect pulses used to implement quantum gates in the hardware. Interestingly, these are also the class of errors on which randomized compiling has the maximum effect of turning them into purely incoherent noise.

4.2.1 Rotation about $Z-$axis

While we ideally want to study the impact of RC on the performance of a quantum error correcting code under general coherent errors, let us first start with a simple yet interesting model – rotations about the $Z-$axis. Although the RC process tailors the underlying physical noise, irrespective of the choice of the code, through this example we show that in fact the gains produced from RC can be arbitrarily increased by choosing codes of increasing distances.

Recall that the rotation about $Z-$axis is specified by $\rho \to R_Z(\omega)\rho R_Z(-\omega)$ where

$$R_Z(\omega) = \cos(\omega/2)\ I + i\sin(\omega/2)\ Z\ . \tag{4.2}$$

Applying the rotation independently across all $n = 7$ the physical qubits of the Steane code, is specified by the map

$$\mathcal{E}(\bar{\rho}) = R_Z^{\otimes n}(\omega)\ \bar{\rho}\ R_Z^{\otimes n}(-\omega). \tag{4.3}$$

The performance of the Steane code under the above error model, can be inferred from Eq. (2.111), where the correctable errors $\mathcal{E}_{\mathcal{C}}$ can be defined with respect to the minimum weight decoder. Explicitly enumerating all correctable errors, we find that there are 22 correctable errors of weight at most one, and 42 two-qubit ones. Since we are confined to rotations about the $Z-$axes, we can limit ourselves to the correctable errors of $Z-$type. Reserving the details of our derivation to Appendix B.1, we find

$$r(\overline{\mathcal{E}}_1) \approx 63\ (\omega/2)^4 - 476\ (\omega/2)^6 + \mathcal{O}(\omega^8)\ . \tag{4.4}$$

In comparison, the logical infidelity for quantum error correction with randomized compiling is

$$r(\overline{\mathcal{E}^T}_1) \approx 21\ (\omega/2)^4 - 112\ (\omega/2)^6 + \mathcal{O}(\omega^8)\ . \tag{4.5}$$

Finally, the performance gain from RC quantified using the metric δ_1 defined in eq. 4.1 can now be estimated as

$$\delta_1 = \frac{r(\overline{\mathcal{E}}_1)}{r(\overline{\mathcal{E}^T}_1)} \approx 3 - \frac{5}{3}\,\omega^2 + \mathcal{O}(\omega^4)\ . \tag{4.6}$$

We now show that the above modest performance gains can be made arbitrarily large by concatenating the Steane code with itself. It is possible to extend the analysis above via recursion to approximate the effective logical channel for a level ℓ concatenated Steane code for $\ell > 1$. The details of this procedure can be found in Appendix B.2. The approximate logical channel allows us to estimate the performance of level ℓ concatenated Steane code and study the impact of randomized compiling on it. To understand the impact of RC with the number of levels, we can do a leading order analysis of the recursive relations used to construct the average logical channel, described in Appendix B.2. We find that for small rotation angle ω, the average infidelity of the logical channel scales as

$$\begin{aligned} r(\overline{\mathcal{E}}_\ell) &\approx 63^{2^\ell - 1}(\omega/2)^{2^{\ell+1}}\ , \\ r(\overline{\mathcal{E}^T}_\ell) &\approx 21^{2^\ell - 1}(\omega/2)^{2^{\ell+1}}\ . \end{aligned} \tag{4.7}$$

Subsequently, the scaling of gain δ_ℓ with the levels of concatenation is given by

$$\delta_\ell \approx 3^{2^\ell - 1} - (5 \times 2^{l-1} \times 3^{2^l - 3})\omega^2 + O(\omega^4)\ . \tag{4.8}$$

Figure 4.1 corroborates this scaling law for the exact value of the logical error rates of the concatenated Steane code, in other words, showing that $\log(\log(\delta_\ell)))$ is approximately a linear function of ℓ. Note that the above analysis is accurate for small rotation angles. Varying the rotation angles leads us to another important discovery. Figure 4.2 shows the gains from randomized compiling for a range of rotation angles for levels $1 \leq \ell \leq 5$. The gains from RC grow significantly with increase in number of levels of the code. The figure suggests the presence of a threshold rotation angle $\omega_\star$ below which arbitrary gains from RC can be achieved by increasing the size of the code (levels of concatenation). On the contrary, for rotations $\omega > \omega_\star$, the trend reverses.

We now turn to more general noise models, where we will find that the presence of a threshold in the case of rotations about the $Z-$axis, extends to the general case.

4.2.2 Rotation about an arbitrary axis

While the above analysis considered coherent error models described by rotations about the $Z-$axis, it is straightforward to apply these ideas to rotations about any of the Pauli

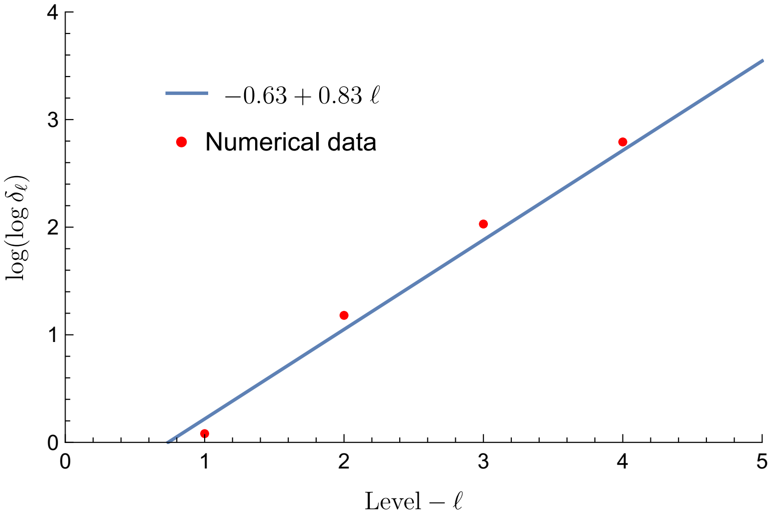


Figure 4.1: The above figure shows that the gain at level ℓ , δ_ℓ, scales doubly exponentially with ℓ. The rotation angle used here is $\omega = \pi/20$.

axes. We now investigate average gains due to RC for a rotation about an arbitrary axis.

We consider a general error model where the physical qubits of a code undergo rotations about an arbitrary axes of the Bloch sphere, described by the unitary matrix U, i.e., $\mathcal{E}(\bar{\rho}) = U^{\otimes n}\bar{\rho}(U^{\dagger})^{\otimes n}$. The following parameterization for U [BRS+09] is useful for our analysis:

$$\begin{pmatrix} \cos(\omega/2) + i\sin(\omega/2)\cos(\theta) & ie^{-i\phi}\sin(\omega/2)\sin(\theta) \\ ie^{-i\phi}\sin(\omega/2)\sin(\theta) & \cos(\omega/2) - i\sin(\omega/2) \end{pmatrix}.$$

where $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$ define the axis (in polar angles) about which each qubit is rotated, and ω gives the magnitude of the rotation. For example, $\theta = \phi = 0$ can be identified with rotations about the $Z-$axis. The performance gain from RC can be defined following Eq. 4.6, as a function of the parameters $\delta(\theta, \phi, \omega)$. The average gain for an unknown axis is computed as

$$\overline{\delta}_\ell(\omega) = \frac{1}{2\pi}\int_0^{2\pi} d\phi \int_0^{\pi} \sin(\theta)\ d\theta\ \delta_\ell(\theta, \phi, \omega)\ , \tag{4.9}$$

for $\ell = 1$. Likewise, for concatenated codes, $\overline{\delta}_\ell$ denotes the average gain in performance for level ℓ. This is similar to the conclusion drawn for the case of rotations about the $Z-$axis. First of all we see that for all coherent errors RC improves the performance

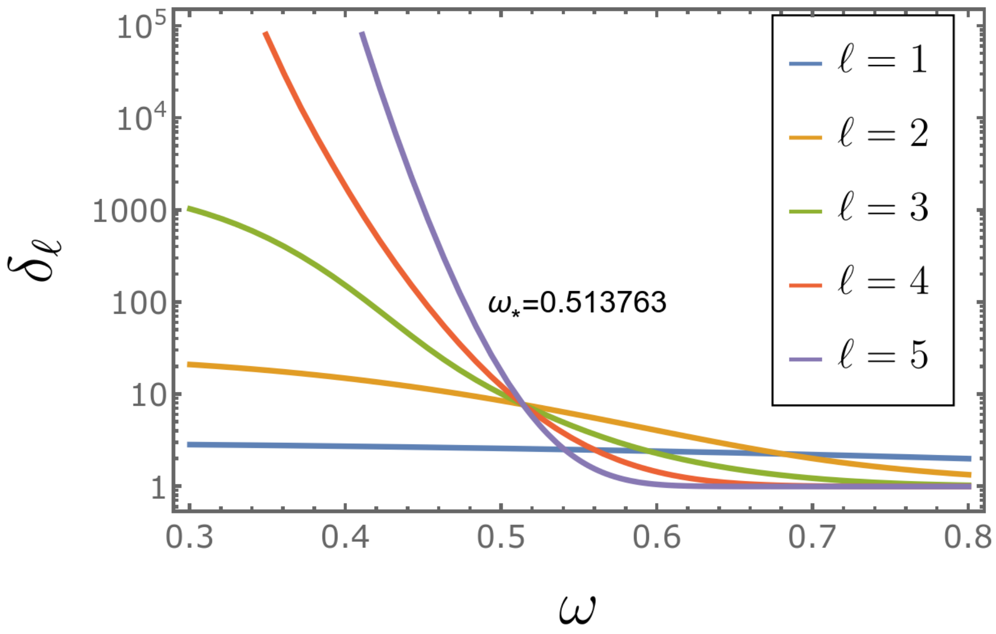


Figure 4.2: Gains in logical performance, δ_ℓ, of a level ℓ concatenated Steane code for rotations by angle ω about the $Z-$axis. The common crossover point lies at $\omega_\star \approx 0.51$, which corresponds to a rotation angle of about 15°, below which gains from RC can be amplified by increasing the number of levels of concatenation.

of the Steane code. Furthermore, performance gains are largest for coherent errors that correspond to rotations about the X, Y or Z axes.

Using the general techniques developed in the appendix to approximate the effective logical channel of a level$-\ell$ concatenated code, we can estimate the gains $\overline{\delta}_\ell$ in average performance due to RC over the various rotation axes. Similar to the case of $Z-$rotations, Fig. 4.3 suggests the presence of a threshold $\overline{\omega}_\star$ wherein for rotation angles $\omega \leq \overline{\omega}_\star$ the gains can be arbitrarily increased by choosing codes of larger distance, whereas the trend reverses for $\omega > \overline{\omega}_\star$.

Note that threshold angle $\overline{\omega}_\star$ for rotations about an unknown axis is higher the threshold for rotations about the $Z-$axes, i.e., $\overline{\omega}_\star > \omega_\star$. This can be explained as follows. In the case of a generic non-Pauli axis, the twirled noise model, i.e., is in the presence of RC, is composed of a probabilistic mixture of X, Y and Z type errors. Whereas, in the case of a fixed Pauli axis, we only have errors of one type (either X, Y or Z). For a fixed error budget, specified by fidelity, the case of a non-Pauli axis results in the error strength spread over a larger number of correctable errors than the case of a fixed Pauli

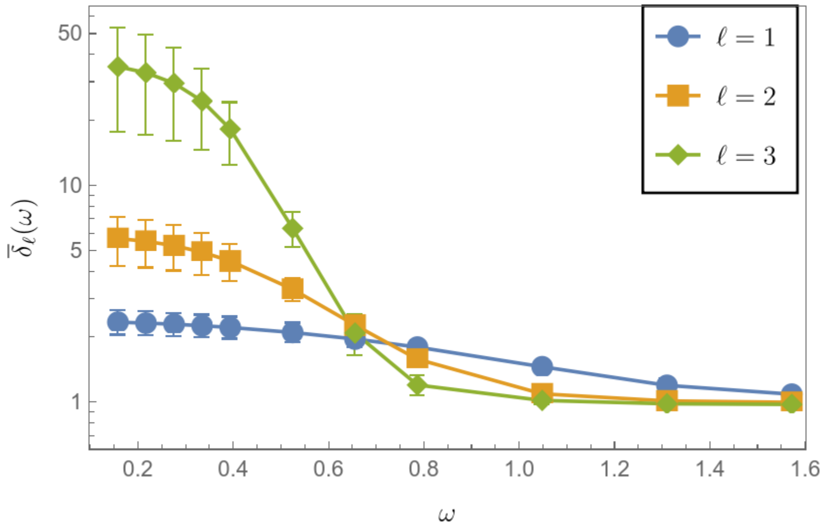


Figure 4.3: The average gain in performance from RC, using the Haar average over all axes of rotation, for the level ℓ concatenated Steane code. The average gains are larger for small magnitudes of rotation. We observe that the gains increase significantly with the number of levels for $\omega \leq \overline{\omega}_\star \approx 0.65$, which corresponds to a rotation angle of about $19°$.

axis which would include relatively higher weight Pauli errors of one type. Hence, the Steane code has better error correction capability. Figure 4.4 provides evidence to our argument by showing that the threshold angle for performance gains from RC under rotations about various axes, is higher for non-Pauli axes compares to the Pauli ones. As a consequence, we also note that for rotation angles $\omega_\star < \omega < \overline{\omega}_\star$, the largest gains from RC are achieved for rotations axes that lie between the X, Y and Z axes as opposed to the individual Pauli axes.

4.2.3 Composition of coherent and stochastic map

So far, we have shown that RC always improves the performance of quantum error correcting codes under coherent errors. Although theoretically we understand that RC only impacts the coherent components, experimental characterization work in the past [CDDH+23; HNM+21] has revealed that noise profiles in quantum devices typically comprise decoherent and coherent components. In what follows, we show that the gains are persistent even in cases where the noise comprises both coherent and decoherent

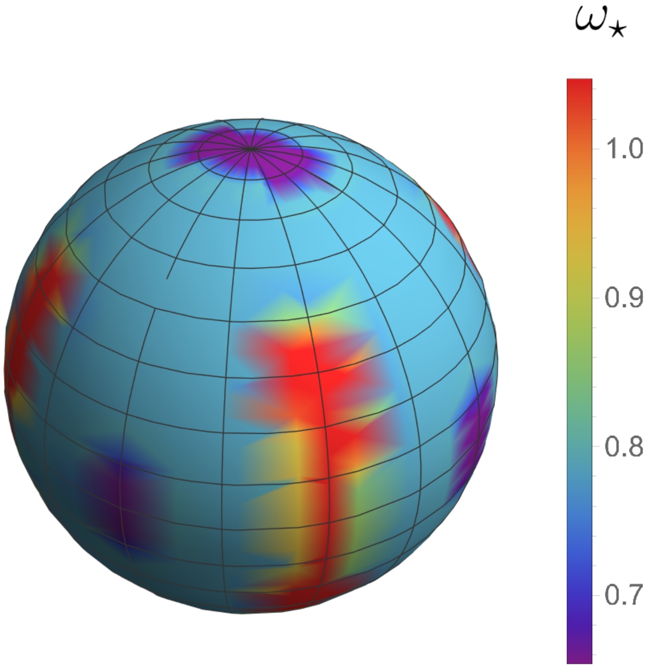


Figure 4.4: The threshold angle $\omega_\star$ for which $\delta_2 < \delta_1$ for rotations about an axis parameterized by θ, ϕ. Each point on the sphere has coordinates $\{\sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi), \cos(\theta)\}$ and the color denotes the threshold value of angle ω for which the above condition holds. It shows that the cardinal axes do not have the highest threshold.

effects. In our numerical studies, we adopt a model motivated by Ref. [CDAE19], which shows that any non-catastrophic quantum channel i.e., a channel with unitarity and fidelity at-least one-half, can be expressed as a composition of a unitary process and a decoherent process. Choosing the unbiased [CDAE19] depolarizing channel as the decoherent component, the overall noise can be expressed as

$$\mathcal{E} \simeq (\mathcal{E}_{dep} \circ \mathcal{E}_{coh})^{\otimes n}, \tag{4.10}$$

where

$$\begin{aligned} \mathcal{E}_{coh}(\rho) &= U\rho U^\dagger, \\ \mathcal{E}_{dep}(\rho) &= (1-p)\rho + \frac{p}{2}\mathbb{I}. \end{aligned} \tag{4.11}$$

and U can be parameterized using Eq. (4.9). In what follows, we will study the impact of RC under the approximation given by Eq. (4.10). Note that both the coherent as well as the incoherent parts of the error model contribute to the strength of noise, for instance, the average gate fidelity. While RC only affects the coherent part of the error process, we expect that for a fixed noise strength, the performance gain due to RC under the error model described above will diminish with increasing p. This expectation is supported by the numerical simulations presented in Fig. 4.5, where we present numerical estimates of $\overline{\delta}_\ell(\omega, p)$ for several depolarizing strengths p. Here, $\overline{\delta}_\ell(\omega, p)$ is defined analogous to Eq. (4.9) as

$$\overline{\delta}_\ell(\omega, p) = \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\pi} \delta_\ell(\theta, \phi, \omega, p) \ \sin(\theta) \ d\theta \ d\phi \ . \tag{4.12}$$

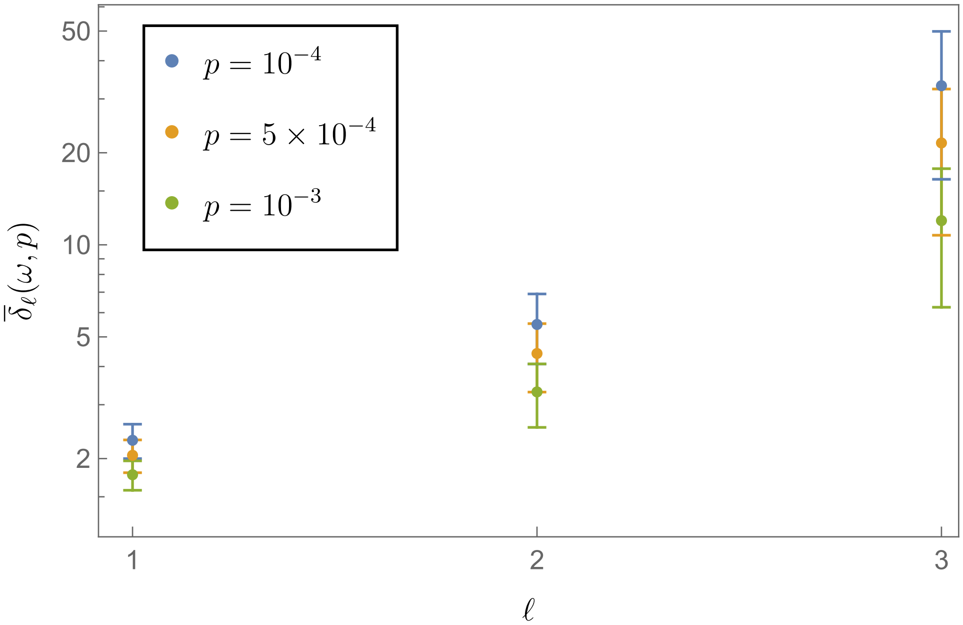


Figure 4.5: The impact of the depolarizing component on the gains from RC. We fix the average infidelity per qubit to be $r \approx 0.003$ and increase the value of the depolarizing strength from $p = 10^{-4}$ to $p = 10^{-3}$. The value of ω corresponding to each value of p is chosen such that the total physical infidelity of the qubit remains constant. We observe that the gains from RC diminish with increase in depolarizing strength. This is because RC does not impact the stochastic component of the noise model.

Note that in all of the error models considered so far, we have only observed gains in performance due to RC. However, amongst the most general CPTP maps including the unital as well as non-unital types, we have identified cases under which RC can lead to a loss in the performance. Some examples of these maps are mentioned in Appendix B.3.

4.3 Conclusion

The application of randomized compiling in fault tolerance is attractive for two reasons. First, amongst the exponentially growing number of parameters controlling a physical noise process, RC effectively eliminates the impact of most of them on a QEC scheme. Second, since RC removes multiple noise sources, we expect the code to perform better. This chapter provides concrete evidence to show that RC improves the performance of quantum error correction under a wide class of coherent errors. We have identified noise regimes where gains are drastic for the case of concatenated Steane codes. In particular, it grows doubly exponentially with the number of levels, under small rotations about a Pauli axis. Our results can be extended to guarantee performance gains under generic unital noise processes, leveraging tools from [CDAE19; CDWE19] that approximate a unital noise process as a composition of a coherent and an incoherent error model. These observations strengthen the need for active noise tailoring methods as a crucial component of a fault tolerant scheme.

Performance gains offered by RC also depend on the strength of errors affecting the physical qubits. We stumbled upon an interesting observation that indicates gains decrease when the amount of coherent rotation error passes beyond a threshold value. To the best of our knowledge a threshold of this nature hasn't been reported in earlier works. The threshold helps estimate the maximum possible noise that can be alleviated on a hardware device by leveraging RC tools. We also carried out extensive studies to analyze the variation of this threshold with the features of the underlying coherent error model.

Beyond the paradigm of identical unital maps across all physical qubits, we argue that unilateral conclusions about performance gains due to RC cannot be made, i.e., it depends strongly on the microscopic details of the underlying physical noise process. Our arguments are strengthened by numerical studies of complex physical noise processes that revealed some cases where the code's performance can also degrade in the presence of RC. In Ref. [CWB+17], it was shown that twirled noise processes may improve or degrade thresholds depending on the code and noise properties. In this chapter we arrive at a similar conclusion by exploring different error models for the minimum weight decoder.

Obtaining efficiently computable estimates for performance gains due to RC in different experimental setups would be crucial to optimizing fault tolerance resources in near-term applications. In the absence of exact values, it would be useful to provide bounds for the impact of RC on the code's performance. Although RC's impact on performance depends strongly on the underlying noise process, it is still interesting to see that it can provide significant gains for a wide variety of realistic error models and relevant error regimes.

To ensure a performance gain from a noise tailoring technique, such as RC, ideally, we want to cancel the impact of those terms in the underlying noise process, which correspond to uncorrectable errors – since these add to the logical infidelity. It would be worthwhile to explore ways of controlling physical noise sources to ensure that RC always offers a gain in performance. It would also be interesting to explore different Twirling gate sets that can tailor the noise process to suppress terms that contribute negatively to the logical channel's fidelity. Although we identified a handful of cases where a performance loss is observed, it will be noteworthy to develop cheap experimental protocols to ascertain whether performing error correction with RC will be significantly beneficial for a given device.

Chapter 5

Testing distance of codes

> This chapter consists of work done in collaboration with Sathyawageeswar Subramanian and Tom Gur. At the time of writing this chapter, its contents aren't posted anywhere online. I am leading this project with inputs and assistance from the aforementioned collaborators.

In the thesis so far, we have explored how to use noise tailoring and characterization tools to efficiently characterize and improve the logical fidelity of quantum stabilizer codes. Another extremely fundamental and important performance metric for codes which has a rich mathematical structure and theory behind it is *minimum distance*. Minimum distance has been widely studied both in the classical and quantum error correction literature. Inspired by the recent work on relaxed decision problems in the last two decades, we close off the thesis by exploring the capabilities of ultra-fast algorithms to estimate the minimum distance of codes.

The chapter is structured as follows. Section 5.1 discusses some key results in the literature related to the computational complexity of finding the minimum distance of a classical binary linear code. In section 5.2, we introduce concepts and notation that are useful for this chapter. Section 5.3 derives classical and quantum query complexity bounds for estimating the minimum distance. In section 5.4, we introduce the property testing version of this problem and provide query complexity bounds for it. Finally, in section 5.5 we provide concluding remarks and describe some interesting open problems.

5.1 Related work

In a seminal paper in 1978 [BMT78], Berlekamp, McEliece, and van Tilborg initiated the study of the *computational complexity* of tasks in coding theory. Where prior works focused on information theoretic aspects [Gol49; Ham50] and applications in communication [Sha48] and compression [ANR74], the authors of Ref. [BMT78] proved NP-hardness of some fundamental tasks in error correction. Specifically, they showed that MAXIMUM LIKELIHOOD DECODING and computing the WEIGHT DISTRIBUTION for binary linear codes are NP-complete. In general, maximum-likelihood decoding for classical codes is concerned with finding the least weight error which is consistent with a given error syndrome. An error *syndrome* is a bit string $s \in \mathbb{F}_2^{n-k}$ which contains information about the error. Specifically, if $x \in \mathbb{F}_2^n$ is corrupted with errors, $Hx^T = s \neq \mathbf{0}$, where $H \in \mathbb{F}_2^{n-k \times n}$ is the parity check matrix of some linear code. Weight distribution problem involves finding the Hamming weights of the codewords. The decision version of these problems are defined below. Please note that the original paper [BMT78] named these problems COSET WEIGHTS and SUBSPACE WEIGHTS respectively.

MAXIMUM LIKELIHOOD DECODING
Input: A binary matrix $H \in \mathbb{F}_2^{n-k \times n}$, a vector $s \in \mathbb{F}_2^{n-k}$, and an integer $w > 0$.
Output: ACCEPT if $\exists\, x \in \mathbb{F}_2^n$, $|x| \leq w$ such that $Hx^T = s$, REJECT otherwise.

WEIGHT DISTRIBUTION
Input: A binary matrix $H \in \mathbb{F}_2^{n-k \times n}$ and an integer $w > 0$.
Output: ACCEPT if $\exists\, x \in \mathbb{F}_2^n$, $|x| = w$ such that $Hx^T = \mathbf{0}$, REJECT otherwise.

In Ref. [BMT78], the authors conjectured that finding the minimum distance of a linear code was NP-complete too. Nearly two decades later, Alexander Vardy made an important advancement where they proved that estimating the minimum distance of a linear code is NP-complete [Var97]. In particular, the following decision version of estimating the minimum distance was shown to be NP-complete.

MINDIST$_{\leq}$
Input: A parity check matrix $H \in \mathbb{F}_2^{n-k \times n}$ corresponding to an $[n, k]$ linear code, and a code-distance threshold $w \in [n-k+1]$.
Output: ACCEPT if there is a nonzero vector $x \in \mathbb{F}_2^n$ of weight $\leq w$ such that $Hx^T = \mathbf{0}$, REJECT otherwise.

Note that, MINDIST$_{\leq}$ can be seen as a special case of MAXIMUM LIKELIHOOD DE-

CODING with the syndrome s fixed i.e., $s = \mathbf{0}$. The following problems which are close variations of the above problems were also shown [NH81] to be NP-complete before MINDIST$_{\leq}$.

CODEWORD$_{\geq}$
Input: A binary matrix $H \in \mathbb{F}_2^{n-k\times n}$ and an integer $w > 0$.
Output: ACCEPT if there is a vector $x \in \mathbb{F}_2^n$ of weight $\geq w$ such that $Hx^T = \mathbf{0}$, REJECT otherwise.

CODEWORD (MOD k)
Input: A binary matrix $H \in \mathbb{F}_2^{n-k\times n}$, an integer $w > 0$ and an integer $k \geq 2$.
Output: ACCEPT if there is a vector $x \in \mathbb{F}_2^n$ of weight $\leq w$ such that $Hx^T = \mathbf{0}$ and $|x| \not\equiv 0(\text{mod } k)$, REJECT otherwise.

CODEWORD RANGE
Input: A binary matrix $H \in \mathbb{F}_2^{n-k\times n}$, and integers $w_2 > w_1 > 0$
Output: ACCEPT if there is a vector $x \in \mathbb{F}_2^n$ such that $Hx^T = \mathbf{0}$ and $w_1 \leq |x| \leq w_2$, REJECT otherwise.

Let us pause for a moment and appreciate the deep significance of proving the hardness of finding minimum distance. It not only implies the hardness of a host of other coding theory problems, but also makes designing codes with high minimum distance challenging. If finding minimum distance were easy, one could iterate over a large number of random linear codes and pick the best one by efficiently checking their distance. We mention the following example of a computational problem whose hardness follows from the fact that MINDIST$_{\leq}$ is NP-hard. This problem is related to determining the trellis complexity of a linear code where the task is to find a permutation of coordinates that minimizes the number of vertices in the minimal trellis for a binary linear code [HK96]. It is important in the theory of block-code trellises [Var98]. The trellis complexity decides the amount of resources need to implement maximum likelihood decoding for a given code. Although permuting symbols leaves the properties of the code invariant, it does impact the time and resources required to decode them using a *Viterbi* algorithm [Vit67] . One should imagine trellises as the different graph representations of equivalent codes which have their unique time and resource complexities for decoding. For more details on trellises and their relation with linear codes, we refer the readers to Ref. [Mas78]. The NP-hardness of the following problem was first proved in Ref. [HK96] via a reduction from the SIMPLE MAX CUT problem.

PARTITION RANK

Input: A binary matrix $H \in \mathbb{F}_2^{m \times n}$, and integers $i, w > 0$

Output: ACCEPT if there is a column permutation that takes H into $H' = [A_i | B_{n-i}]$ such that $A_i \in \mathbb{F}_2^{m \times i}, B_i \in \mathbb{F}_2^{m \times n-i}$ and $\text{rank}(A_i) + \text{rank}(B_{n-i}) \leq w$, REJECT otherwise.

Here $\text{rank}(A)$ is defined as the maximal number of linearly independent columns in A. We will now describe a reduction [Var97] from MINDIST$_{\leq}$ to PARTITION RANK. Note that the smallest integer i for which

$$\text{rank}(A_i) + \text{rank}(B_{n-i}) < \text{rank}(H) + i \tag{5.1}$$

is equal to $\min\{d, d^{\perp}\}$, where d is the minimum distance of the binary linear code defined by H, and $d^{\perp}$ is the minimum distance of the corresponding dual code. Note that since we are taking a minimum of d and $d^{\perp}$, H can be interpreted both as a parity check matrix and a generator matrix. An intuition for the previous statement is as follows. The statement

$$\text{rank}(A_i) + \text{rank}(B_{n-i}) \leq \text{rank}(H) + i \tag{5.2}$$

holds for all $i \in [n]$ since $\text{rank}(A_i) \leq i$ and $\text{rank}(B_{n-i}) \leq \text{rank}(H)$. First, let us think of H as the parity check matrix. Note that A_i will be a full rank matrix for all values of $i < d$ since all column subsets of H of size $< d$ are linearly independent. This follows from the definition of minimum distance. Therefore, d is the minimum number of columns for which $\text{rank}(A_i) < d$. The same argument can be recycled by thinking of H as the generator matrix for the dual code leading to the above conclusion. Next, we will describe a recipe to use the above fact to compute the minimum distance of a code using access to a (hypothetical) polynomial time algorithm for PARTITION RANK.

Let $\mathcal{C}$ be an $[n, k, d]$ code whose distance d we wish to determine. We construct a binary Reed-Muller code $\mathcal{C}'$ with parameters (n', k', d'), where

$$\begin{aligned} n' &= 2^{2\lceil \lg n \rceil + 1}, \\ k' &= n'/2 \leq 4n^2, \text{ and} \\ d' &= 2^{\lceil \lg n \rceil + 1} \geq 2n. \end{aligned}$$

One can then use the Kronecker product construction [MS77] to obtain the generator matrix for the code $\mathcal{C}^{\star} = \mathcal{C}^{\perp} \otimes \mathcal{C}'$. The parameters for $C^{\star}$ obey the following inequalities:

$$\begin{aligned} n^{\star} &= nn' \leq 8n^3, \text{and} \\ d^{\star} &= d^{\perp} d' \geq 2nd^{\perp} \geq n > d. \end{aligned}$$

The dual distance of $\mathcal{C}^\star$ is the minimum of the dual distances of $\mathcal{C}^\perp$ and $\mathcal{C}'$ i.e., $\min\{d, d'\} = d$. Therefore, we can use the generator matrix for $\mathcal{C}^\star$ in the PARTITION RANK algorithm for determining d. Since computing minimum distance in NP-hard, this reduction proves the NP-hardness of the PARTITION RANK problem. Note that the clever construction of $\mathcal{C}^\star$ ensures that the the minimum distance of the input code is the minimum of the distances $d^\star = \mathrm{d}_{\mathcal{C}^\star}$ and $(d^\star)^\perp = \mathrm{d}_{(\mathcal{C}^\star)^\perp}$ i.e., $\min\{d^\star, (d^\star)^\perp\} = d$. Moreover, the reduction constructs a code with parity check matrix of size $\mathsf{O}\left((\mathsf{n}^\star)^2\right) = \mathsf{O}(\mathsf{n}^6)$ and hence takes $poly(n)$ time.

More connections to linear algebra. As a side note, a quantity similar to minimum distance in a parity check matrix with respect to the rational field is called the *spark* of a matrix. Spark is defined to be the minimum number of columns in a matrix required to form a dependent set. It is known that computing spark of a matrix is NP-complete [TP14]. It was also shown under some complexity theoretic assumptions that the minimum distance cannot be approximated up to within any constant factor in random polynomial time (RP) and to within an additive error that is linear in n [DMS99]. The hardness of computing minimum distance in the context of low-density parity-check (LDPC) codes was explored in Ref. [HFE04].

Equivalent problems for quantum codes. One can define quantum equivalents of all the problems mentioned so far. The analogue of maximum likelihood decoding for quantum stabilizer codes involves finding the most probable error coset. This is because we only care about correcting errors up to a stabilizer. It was shown in Refs. [HLG11; Fuj12] that decoding stabilizer codes is NP-hard under different assumptions for the distance metric. Ref. [KL20] proves that the hardness results hold even when the error model is restricted to the depolarizing model and the class of codes is restricted to a small class of stabilizer codes with low full-rank check matrices. A stronger hardness result was shown in Ref. [IP15], where the authors proved that finding the most likely equivalence class of errors in degenerate quantum maximum likelihood decoding is #P-complete. Although there was significant progress on the decoding of quantum stabilizer codes, the hardness of computing the minimum distance for these codes was unknown until recently. In Ref. [KK22], the authors show that computing the minimum distance of a quantum stabilizer code is NP-hard. In particular, they show that computing or even approximating (to an additive/multiplicative precision) the minimum distance of a CWS code is NP-complete. CWS codes are a class of quantum codes specified by a classical code and a graph. When the classical code is linear, they are exactly equivalent to the class of stabilizer codes. For details, please see Ref. [CSS+09]. Their reduction from classical to quantum involves the construction of a non-degenerate code. Therefore, this implies that the hardness results hold true even for the case when restricted to the subclass of non-degenerate codes. Finally, using the mapping in Ref. [BTL10], which takes an $[[n,k,d]]$ stabilizer code to $[[4n,2k,2d]]$ CSS code in $poly(n)$ time, their results can be extended to show that computing/approximating the minimum distance of CSS codes is NP-hard as well. CSS codes are a special class of stabilizer codes where the stabilizer group is a product of two subgroups S_X and S_Z containing Pauli operators of only X-type and Z-type respectively.

Our setting. We reimagine the aforementioned line of work by first investigating the query complexity of finding the minimum distance of a code given access to the adjacency list oracle for the parity check matrix of the code. Further, we extend the problem to the setting of *property testing* algorithms for the distance of classical linear codes. We started this project wanting to address this problem for quantum codes but soon discovered that the simpler case of classical codes has not been addressed. Therefore, in this chapter we will restrict ourselves to studying the classical and quantum complexities of the problem of testing minimum distance of classical linear codes. In particular, the theme of our work is captured by the following central question: Is it possible to test whether a given classical linear code has large distance by looking at only a small fraction of the input representing the code?

We remark that Ref. [Kir18] has also previously considered a similar problem, in the context of information set decoding, under the name of the k-list matching problem. Another closely related problem is the k-list problem or generalized birthday problem [Wag02; GNPS18], which has been studied in the context of cryptanalysis, and primarily in a probabilistic setting where average-case computational complexity is the primary interest. However these settings do not concern us at the moment in this chapter.

Comparison with locally testable codes. At this point, it is crucial to mention and distinguish our setting from the one of locally testable codes [FS95; RS96; GS06; GGK19]. In the decoding step of error correction, checking whether a given string is a valid codeword is an important task. In case it is corrupted by errors, suitable correction needs to be applied. Locally testable codes are error correcting codes that allow super fast testing of this property. In particular, they have testers which can check with high probability of success whether a string is a valid codeword or it is far (in Hamming distance) from all codewords. These testers run in sub-linear time and are *local* i.e., they only look at small number of bits of the given input string. In this setting, we *fix* a code and test whether a given input string is a valid codeword. However, in the setting we consider, we are testing a global *property of the code* itself i.e., the minimum distance, where the *input* is *the code* itself. The code is specified using an oracle to the columns of its parity check matrix.

Our contributions. We study the query complexity of the decision version of finding the minimum distance of a code i.e., MinDist$_{\leq}$, under the model where each query returns a column of the parity check matrix of the code. One can iterate this procedure to calculate the distance using a binary search. We provide classical and quantum lower and upper bounds for this task. We also define a property testing version of the same problem and study the query complexity under the same model. We provide non-trivial lower bounds for this variant. Although the upper bound from the decision version holds for this variant, we conjecture that it can be improved using the extra structure in the testing version. We discuss some potential approaches and leave this open for future research. The results are summarized in table 5.1. In this chapter, we will restrict ourselves to binary codes. However, some of the methods discussed can be generalized to codes defined over higher dimensions.

Query complexity of DepSet$_w$		
	Lower bound	Upper bound
Classical	$\Omega(n)$ [Lemma 6]	$O(n)$ [trivial]
Quantum	$\Omega(n^{2/3})$ [Lemma 6]	$O(n^{w/(w+1)})$ [Lemma 7]

Query complexity of Test MinDist$_{\geq}$		
	Lower bound	Upper bound
Classical	$\Omega(n^{1/2})$ [Lemma 9]	$O(n)$ [trivial]
Quantum	$\Omega(n^{1/3})$ [Lemma 9]	$O(n^{w/(w+1)})$ [Lemma 8]

Table 5.1: The above tables summarize the results presented in this chapter for the query complexity of estimating and testing the minimum distance of binary linear codes. The classical and quantum bounds for DepSet$_w$ build upon results from Refs. [Yao94; GKH+96; BSS+03] and [BKT18; Amb04] respectively. The bounds for the Test MinDist$_{\leq}$ problem use results from Refs. [Aar02; Shi02; Kut05; Amb05; Amb04].

5.2 Preliminaries and notation

Classical query complexity. Informally, query complexity is a way to measure the amount of information about the input required to compute any function on it. The access to the input is provided via an oracle. For instance, oracle access to n-bit input string $x \in \{0,1\}^n$ is a simple map $i \to x_i$. The *classical randomized query complexity* of a function $f : \{0,1\}^n \to \{0,1\}$ is the total number of oracle calls made by the best (randomized) classical algorithm to compute f with error $\epsilon = 1/3$, and will be denoted by $R(f)$.

Quantum query complexity. We will adopt the notion of quantum query complexity as described in, e.g Ref. [Kot14]. In quantum algorithms, the oracle access to the input is facilitated by a unitary whose action is described by $Q_x|i,b\rangle \rightarrow |i, b \oplus x_i\rangle$. Any quantum algorithm with query complexity T can be described by $T+1$ unitaries $U_0, U_1, \ldots U_T$ which act on $m \geq (\log n + 1)$ qubits. It can perform any unitary $V_x = U_T Q_x U_{T-1} Q_x \ldots U_1 Q_x U_0$, where we assume Q_x is implicitly tensored with identity if any U_i acts on more qubits than Q_x. A quantum algorithm is said to compute a function $f : \{0,1\}^n \rightarrow \{0,1\}$ on an input x i.e., $f(x)$ with error ϵ if the probability of obtaining $f(x)$ by measuring the first qubit of $V_x|0^m\rangle$ is at least $(1-\epsilon)$. The bounded-error quantum error complexity of a function f is defined to be the quantum query complexity of the best quantum algorithm that computes f with error $\epsilon = 1/3$. We will denote the quantum query complexity of f by $Q(f)$. Note that $\mathsf{O(n)}$ is a trivial upper bound to both classical and quantum query complexities i.e., $R(f)$ and $Q(f)$. In this chapter, we will use more complex oracles to access our input but the essential concept to measure the query complexity remains identical.

Input models and distance measures. We briefly describe some options for the input models and some distance measures between parity check matrices. The parity check matrix $H \in \mathbb{F}_2^{n-k \times n}$ can be queried in the following ways:

1. *Adjacency matrix model*: In this model, each query simulates an access to one entry of the matrix. The action of the classical oracle can be described as

$$O^H_{mat}(i,j) = H_{ij} \; \forall \; i \in [n-k], j \in [n], \tag{5.3}$$

where H_{ij} is the entry of the matrix at the i^{th} row and j^{th} column. The action of the analogous quantum oracle is given by

$$O^H_{mat}(|i,j\rangle|t\rangle|z\rangle) = |i,j\rangle|H_{ij} \oplus t\rangle|z\rangle \; \forall i \in [n-k], j \in [n], t \in \{0,1\}, \tag{5.4}$$

where $|i,j\rangle$ is the index register, $|t\rangle$ is the target register, and $|z\rangle$ is the work register that is not affected by the query operation.

2. *Adjacency list model*: In this model, each query simulates an access to one column of the matrix. The action of the classical oracle can be described as

$$O^H_{col}(i) = H[i] \; \forall \; i \in [n], \tag{5.5}$$

where $H[i]$ is the i^{th} column of the matrix. When queried by a quantum algorithm, the oracle is a unitary whose action is given by

$$O^H_{col}(|i\rangle|t\rangle|z\rangle) = |i\rangle|H[i] + t \; (\bmod 2^{n-k})\rangle\rangle|z\rangle \; \forall \; i \in [n], t \in [2^{n-k}], \tag{5.6}$$

where $|z\rangle$ is the work register that is not affected by the query operation. It would be helpful to imagine the columns of H as explicit bit strings as opposed to integers in $[2^{n-k}]$. An equivalent definition for the above oracle is:

$$O_{col}^{H}(|i\rangle|t_1,\ldots,t_{n-k}\rangle|z\rangle) = |i\rangle|H_{1i}\oplus t_1,\ldots,H_{(n-k)i}\oplus t_{n-k}\rangle|z\rangle \;\forall\; i\in[n], t\in\{0,1\}^{n-k}. \tag{5.7}$$

Please note that the quantum oracles can potentially be queried in superposition. When we talk about quantum query complexity, we will refer to the number of queries made to the quantum versions of the oracles whereas for the classical query complexity, we will count the number of queries made to classical versions of the same oracle.

The distance between two parity check matrices $H, H' \in \mathbb{F}_2^{n-k\times n}$ can be measured in the following ways:

1. *Hamming distance*: It is defined as the number of dissimilar entries in the two matrices and calculated as:

$$\delta_{Ham}(H,H') = \sum_{i\in[n-k],j\in[n]} H_{ij}\oplus H'_{ij}. \tag{5.8}$$

2. *Discrete or column distance*: We imagine the parity check matrices of the codes to be multisets of their columns i.e., $H = \{H[i]\}$ and $H' = \{H'[i]\}$, where $1 \le i \le n$. The column distance between them is defined as:

$$\delta_{col}(H,H') = \frac{|H\setminus H'| + |H'\setminus H|}{2}, \tag{5.9}$$

where $A\setminus B = \{a : (a\in A)\wedge(a\notin B)\}$ is the set difference along with multiplicities. For example, when $H = \{v_1, v_2, \ldots, v_n\}$ and $H' = \{v_1, v_1, \ldots, v_1\}$ (v_1 repeated n times), $\delta_{col}(H,H') = (n-1)$. Note that when H, H' are of the same size, we have $\delta_{col}(H,H') = |H\setminus H'| = |H'\setminus H|$. We will overload the notation H to mean the parity check matrix as well as the multiset (to be inferred from the context). Given three parity check matrices H_1, H_2, H_3 *of the same size* imagined as multisets of columns, the following statements about the column distances between any pair are true.

   (a) $\delta_{col}(H_1,H_2) = 0$ if and only if $H_1 = H_2$.

   (b) It is symmetric i.e., $\delta_{col}(H_1,H_2) = \delta_{col}(H_2,H_1)$.

(c) The triangle inequality holds i.e.,

$$\delta_{col}(H_1, H_3) \leq \delta_{col}(H_1, H_2) + \delta_{col}(H_2, H_3). \tag{5.10}$$

An easy way to see this is to observe that to go from H_1 to H_3, one (potentially sub-optimal) strategy is to first go from H_1 to H_2 by changing $\delta_{col}(H_1, H_2)$ columns and then go from H_2 to H_3 by changing $\delta_{col}(H_2, H_3)$ columns.

In this chapter, we will primarily use the *adjacency list oracle* and the *column distance* measure. The other options will be considered in future research. Going forward, we will drop the subscript $\mathcal{C}$ from the notation $d_{\mathcal{C}}$ to denote the minimum distance of the code $\mathcal{C}$. We will instead overload the notation and use d_H to denote the minimum distance of the code whose parity check matrix is H.

5.3 Estimating the minimum distance of a linear code

Recall that if for two problems A, B there exists a many-one reduction such that A reduces to B (denoted $A \leq B$), i.e. every instance of problem A can be transformed into an instance of problem B, then the solution to the former can be obtained from the solution to the latter. If the reduction itself has query complexity q, then it follows that any query complexity upper bound q_B for B translates into an upper bound $q_A = q + q_B$ for A; similarly, a lower bound $\underline{q_A}$ for A translates into a lower bound $\underline{q_B} = \underline{q_A} - q$ for B.

In this section, we will explore some relations between the variants of the MINDIST problem defined in section 5.1, and well-known problems in the literature, with the aim of obtaining upper and lower bounds on the classical and quantum query complexities of the former. Towards that aim, we first identify some problems of interest, before proceeding to elucidate local many-one reductions that map between these problems and variants of MINDIST.

Recall that the problem of determining whether a given parity check matrix H, thought of as a list of its column vectors, contains a dependent subset of columns of cardinality at most w exactly captures the problem of deciding whether the code described by H has distance at least w. Accordingly, we first define the following problem.

DEPSET$_w$
Input: Adjacency list oracle O^H_{col} for a parity check matrix $H \in \mathbb{F}_2^{n-k \times n}$, threshold parameter $w \in [3, n-k+1]$.
Output: ACCEPT if $\exists$ a dependent set of size $< w$, REJECT otherwise.

A common source of problems used to prove lower bounds on query complexity comes from the rich theory of communication complexity. To motivate the first problem of interest, consider the case of a code that has distance two. The parity check matrix of such a code must have a dependent subset of columns of size two. That is, there must exist two identical columns (over $\mathbb{F}_2$). A central problem in the communication complexity setting is detecting whether two players hold disjoint sets of inputs. Using the intuition about parity check matrices of distance two, we can achieve a (weak) lower bound on DEPSET$_w$ by solving an instance of the famous SET-DISJOINTNESS problem in the two-way communication model defined below.

t-SET-DISJOINTNESS
Input: $A, B \subset U$ for some ground set with $|U| = n$ and $|A| = |B| = t$.
Output: ACCEPT if $A \cap B = \phi$, REJECT otherwise.

In the two-way communication model, Alice and Bob receive sets A, B of size t from a universe of size n in the form of bit strings $a, b \in \{0,1\}^n$ such that $a_i = 1$ iff the element of U indexed by i is included in A and so on, under some predecided ordering. The players have access to shared random bits, are allowed two-way communication, and must decide whether or their inputs are disjoint. We refer to [She14] for a review of results on disjointness. We have the following warm up reduction.

Lemma 5. t-SET-DISJOINTNESS $\leq$ DEPSET$_3$, *and hence* $R($DEPSET$_w) = \Omega(n/\log n)$.

Let $U = [n]$ and let the indicator bit strings for Alice's set be $x \in \{0,1\}^n$ such that $x[i] = 1 \iff$ Alice's set contains the element i. Similarly, let the corresponding string for Bob be y. Let $S_A = \{i_1, i_2, \ldots, i_t\}, S_B = \{j_1, j_2, \ldots, j_t\}$ be sets of indices such that $x[s] = 1\ \forall s \in S_A$ and $y[s] = 1\ \forall s \in S_B$, where $1 \leq t \leq n/2$ is the number of elements in each set. Given these input strings, Alice and Bob can privately construct matrices $H_A = \{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_t\}$ and $H_B = \{\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_t\}$ where for each $a, b \in [t]$,

$$\vec{v}_a = \vec{\mathbf{e}}_{i_a}, \quad \vec{w}_b = \vec{\mathbf{e}}_{j_b},$$

where $\vec{\mathbf{e}}_i$ is the i^{th} standard basis vector, having 1 in the i^{th} co-ordinate and zero everywhere else. Alice and Bob can both construct another matrix $H_C = \{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_{n-2t+1}\}$ where $\vec{u}_i = \sum_{a=2}^{i+1} \vec{\mathbf{e}}_a$, and the input matrix for DEPSET$_w$ is defined as $H = [H_A|H_B|H_C]$.

Claim: Alice and Bob have overlapping sets $\iff$ $\exists$ a dependent set of size two in H. Moreover, the number of queries made to H is $\Omega(\mathsf{t}/\log(\mathsf{n}))$.

Proof. We describe the protocol first and analyze its correctness and complexity below. Alice and Bob sample column index $i \in [2t]$ uniformly at random. The following cases are possible:

1. $1 \leq i \leq t$: Alice queries H and conveys the index for the non-zero element in the vector to Bob using $\log(n)$ bits.
2. $t+1 \leq i \leq 2t$: Bob queries H and conveys the index for the non-zero element in the vector to Alice using $\log(n)$ bits.

Note that all the columns of H are distinct when Alice and Bob have disjoint sets. Therefore, there does not exist any dependent set of size two. However, if there is an overlapping element, it'll lead to two identical columns in H, thereby creating a dependent set of size two. This proves the first part of the claim.

The number of bits to be communicated to solve SET-DISJOINTNESS is at most $\log(n)$ times the number of queries made to DEPSET$_w$. Since, it is known [KS92; BYJK+04; Raz92] that t-SET-DISJOINTNESS requires $\Omega(\mathsf{t})$ bits, the number of queries to solve DEPSET$_w$ is $\Omega(\mathsf{t}/\log(\mathsf{n}))$. Considering the extreme value of $t = {}^{n}/_{2}$ establishes the lower bound in statement of the lemma.

□

Another problem which involves detecting whether a list of items contains a duplicate is the well-studied element distinctness problem.

ELEMENT DISTINCTNESS
Input: Function $f : [n] \to [N], n \leq N$.
Output: ACCEPT if $\exists\, x_1, x_2 \in [n]$ distinct such that $f(x_1) = f(x_2)$, REJECT otherwise.

A reduction from this problem helps improve the classical lower bound achieved using t-SET-DISJOINTNESS previously to $\Omega(\mathsf{n})$, which is optimal.

Lemma 6. ELEMENT DISTINCTNESS $\leq$ DEPSET$_3$. *Hence* $R(\text{DEPSET}_w) = \Omega(\mathsf{n})$, *and* $Q(\text{DEPSET}_w) = \Omega(\mathsf{n}^{2/3})$.

Proof. Assume $N = 3n/2$ and let $f : [n] \to [N]$ be the input instance of ELEMENT DISTINCTNESS. We construct an $N \times 2n$ matrix $[\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_N, \vec{v}_{N+1}, \ldots \vec{v}_{2n}]$ where each column $\vec{v}_i \in \mathbb{F}_2^N$ and

$$\vec{v}_i = \begin{cases} \vec{\mathbf{e}}_{f(i)} & 1 \leq i \leq n \\ \displaystyle\sum_{a=1}^{i-n+1} \vec{\mathbf{e}}_a & n+1 \leq i \leq 2n \end{cases}$$

Here $\vec{\mathbf{e}}_i$ is the i^{th} standard basis vector, having 1 in the i^{th} co-ordinate and zero everywhere else. Notice that by construction, for $n+1 \leq i \leq 2n$, the Hamming weight of $\vec{v}_i$ is exactly $2 \leq i-n+1 \leq n+1$. Hence none of these vectors can form a dependent subset of size less than three with any other set of vectors in H.

We can see that $\exists i_1, i_2 \in [N]$ distinct indices such that $f(i_1) = f(i_2)$ iff $\exists$ a dependent set of size two in $H = \{\vec{v}_1, \ldots, \vec{v}_{2n}\}$ —

$$\begin{aligned} &\exists i_1 \neq i_2 \in [N] \; s.t. \; f(i_1) = f(i_2) \\ &\qquad\qquad \iff \vec{v}_{i_1} = \vec{v}_{i_2} \\ &\qquad\qquad \iff \{\vec{v}_{i_1}, \vec{v}_{i_2}\} \text{ is a dependent set,} \end{aligned} \tag{5.11}$$

since as observed above, any pair of vectors $\{\vec{v}_{i_1}, \vec{v}_{i_2}\}$ in H that is dependent must have $i_1, i_2 \leq n$, because $\{v_i \; \forall i \in [n+1, 2n]\}$ are distinct and unique by construction.

The lower bounds on the query complexities of DEPSET$_w$ follow from the above reduction and known classical [Yao94; GKH+96; BSS+03] and quantum results [Amb04]. □

A more general variant of the element distinctness problem is that of detecting whether a list of items contains one item that repeats at least k times.

K-ELEMENT DISTINCTNESS
Input: Function $f : [n] \to [N]$, $n \leq N$.
Output: ACCEPT if $\exists \; x_1, x_2, \ldots, x_k \in [n]$ distinct such that $f(x_1) = f(x_2) = \ldots = f(x_k)$, REJECT otherwise.

In an important paper, Ambainis [Amb04] showed that ELEMENT DISTINCTNESS, which has a worst-case randomized query complexity of $\Theta(n)$ [Yao94; GKH+96; BSS+03], can be solved with quantum query complexity $\Theta(n^{2/3})$. In the same paper, he extended his quantum walk algorithm to the case of k-ELEMENT DISTINCTNESS for any constant $k = O(1)$ that does not scale with n, obtaining an $O(n^{k/k+1})$ upper bound on its quantum query complexity. There is a long and ongoing line of work on upper [BL11; Bel12a; Bel12b] and lower bounds [BKT18; MTZ20; She20; JZ22] on the quantum query and time complexities of the k-ELEMENT DISTINCTNESS problem. We note that all known lower bounds and algorithms are given for the regime of constant k, and little is known when k is allowed to vary with n. It will become clear below that for our MINDIST problem, k can naturally depend on n, and in fact regimes of large k are of particular interest to us.

However k-Element Distinctness does not directly capture the case of a code with distance at most k, since k repeating columns in H would still mean that the distance of the code is only two. On the other hand, what we need to detect is the presence of a linearly dependent subset of at most k columns. This is, as is k-distinctness itself, an instantiation of the k-subset finding problem.

k-Subset Finding
Input: (i) Function $f : D \to R$, for some finite domain D and range R, and $|D| = n$ defines the problem size; (ii) a property $P \subset (D \times R)^k$.
Output: accept if $\exists\, x_1, x_2, \ldots, x_k \in D$ such that $((x_1, f(x_1)), \ldots, (x_k, f(x_k))) \in P$, and reject otherwise.

It was realized soon after Ambainis presented his algorithm for k-Element Distinctness that the same algorithm can in fact address any problem that has *1-certificate complexity* equal to k [CE05]. Loosely speaking, the 1-certificate complexity of a function f is the minimal size of a certificate that shows that an input satisfies f; taking the example of f being the k-Element Distinctness decision function, notice that a subset of size k of domain elements that all map to the same point in the range definitionally constitutes a 1-certificate for the problem. With this observation, it becomes clear that searching for size k dependent subsets of a given set of vectors has 1-certificate complexity k and can be solved by Ambainis' quantum walk algorithm. Using this intuition, we have the following reduction. Although proven using $\mathbb{F}_2$, the statement and the proof for the following lemma is true for $\mathbb{F}_q$ for all integers $q \geq 2$.

Lemma 7. DepSet$_w$ $\leq$ k-Subset Finding. *Hence* $Q($DepSet$_w) = \mathsf{O}(\mathsf{n}^{w/(\mathsf{w}+1)})$ *for* $w = \mathsf{O}(1)$.

Proof. Given a matrix $H = [\vec{v}_1 \vec{v}_2 \ldots \vec{v}_n]$ where each column $\vec{v}_i \in \mathbb{F}_2^{n-k}$ is a vector of length $(n-k)$, interpret H to also mean the multiset $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$ and let $f : [n] \to H$ be an indexing function that maps $i \mapsto \vec{v}_i$. Let $g : 2^H \to \{0,1\}$ be a function that maps a subset $S \subseteq H$ of $|S| = w$ vectors to $g(S) = 1$ if S is dependent, i.e. there exist nonzero coefficients $\alpha_1, \ldots, \alpha_w \in \mathbb{F}_2$ such that $\sum_{i=1}^{w} \alpha_i s_i = 0$. Otherwise $g(S) = 0$. Notice in particular that such a dependent subset S of size w exists if and only if H contains a dependent subset of size *at most* w; in other words, every dependent subset $S' \subset H$ with $|S'| \leq w$ gives rise to (multiple) dependent subsets S with $|S| = w$.

Hence, to solve an instance of DepSet$_w$ we solve the instance f of w-Subset Finding. Since we are not concerned with the computational complexity of checking whether a given set of vectors is dependent, we bundle this complexity aside inside the oracular function g. We thus get an upper bound of $\mathsf{O}(\mathsf{n}^{w/(\mathsf{w}+1)})$ by applying Ambainis' quantum walk algorithm [Amb04; CE05]. □

The $\mathsf{O}(\mathsf{n}^{\mathsf{k}/(\mathsf{k}+1)})$ query complexity of Ambainis' quantum walk algorithm is tight for some kinds of problems with 1-certificate complexity of constant size—for instance, the k-sum problem [BS13].

K-SUM
Input: A list of $n+1$ elements $t, x_1, \ldots, x_n \in G$ of a finite Abelian group G, and an arbitrary but fixed and constant positive integer k.
Output: ACCEPT if there exists a subset of k elements in $x_1, \ldots, x_n$ that sum up to t, and REJECT otherwise.

The K-SUM problem almost captures the essence of DEPSET$_w$ problem. If we fix the group G to be $\mathbb{F}_2^n$ and set $t = \mathbf{0}$, the subset of k elements that add up to $\mathbf{0}$ will represent a dependent set in the matrix $H = \{x_1, x_2, \ldots, x_n\}$. Therefore, the instances that are accepted for K-SUM would be accepted by DEPSET$_k$ under this reduction. However, there can exist dependent sets of size $< k$ which will lead to acceptance of erroneous instances for the K-SUM problem. Therefore, the analogous problem that would exactly capture DEPSET$_k$ is the following problem.

≤K-SUM
Input: A list of $n+1$ elements $t, x_1, \ldots, x_n \in G$ of a finite Abelian group G, and an arbitrary but fixed and constant positive integer k.
Output: ACCEPT if there exists a subset of up to k elements in $x_1, \ldots, x_n$ that sum up to t, and REJECT otherwise.

Deriving non-trivial lower bounds for the quantum query complexity of ≤K-SUM is an interesting open problem which will also help close the gap in bounds for DEPSET$_w$.

5.4 Testing the minimum distance of a linear code

In this section, we will define two property testing variants of estimating the minimum distance of a linear code and discuss subtle differences between them.

TEST MINDIST$_{\geq}$

Input: Adjacency list oracle O^H_{col} for a parity check matrix $H \in \mathbb{F}_2^{n-k\times n}$, a code-distance threshold $w \in [3, n-k+1]$ and a proximity parameter $\epsilon > 0$.

Promise: Either $\mathsf{d}_H \geq w$ or $\mathsf{d}_H < w$ with the condition that $\delta_{col}(H, H') \geq \epsilon n$ $\forall H'$ s.t. $\mathsf{d}_{H'} \geq w$.

Output: ACCEPT if $\mathsf{d}_H \geq w$, REJECT otherwise.

TEST MINDIST$_{\leq}$

Input: Adjacency list oracle O^H_{col} for a parity check matrix $H \in \mathbb{F}_2^{n-k\times n}$, a code-distance threshold $w \in [2, n-k]$ and a proximity parameter $\epsilon > 0$.

Promise: Either $\mathsf{d}_H \leq w$ or $\mathsf{d}_H > w$ with the condition that $\delta_{col}(H, H') \geq \epsilon n$ $\forall H'$ s.t. $\mathsf{d}_{H'} \leq w$.

Output: ACCEPT if $\mathsf{d}_H \leq w$, REJECT otherwise.

Note that a tester for TEST MINDIST$_{\leq}$ can trivially accept all inputs since by changing just one column of any given parity check matrix, the distance of the code it generates can be changed to two. Therefore by changing only one column, one can obtain a code with low distance. In other words, every code is *close* to a code with distance two for $\epsilon = \mathsf{O}(1)$ under the distance metric δ_{col} and for $\epsilon = \mathsf{O}(1/\mathsf{n})$ under the Hamming distance metric δ_{Ham}. Therefore, this testing variant is not interesting and we will focus on TEST MINDIST$_{\geq}$ for the rest of this chapter.

The main results in this section are summarized below:

1. We provide a tester for TEST MINDIST$_{\geq}$ with quantum query complexity $\mathsf{O}(\mathsf{n}^{w/(\mathsf{w}+1)})$.
2. We show that any classical algorithm to solve TEST MINDIST$_{\geq}$ will require $\Omega(\mathsf{n}^{1/2})$ queries to the classical oracle and any quantum algorithm requires $\Omega(\mathsf{n}^{1/3})$ queries to the corresponding quantum oracle.

We will design a tester for TEST MINDIST$_{\geq}$ using tools similar to the one used for DEPSET$_w$. The core idea is to use Ambainis' quantum walk algorithm to search for a dependent set of size *at most* $(w-1)$ in a subset of columns queried uniformly at random. If the search is successful, the set found serves as a certificate that the minimum distance is *at most* $(w-1)$. If the search is unsuccessful, we conclude that with high probability, the minimum distance of the given code is $\geq w$. Using this intuition, we have the following lemma.

Lemma 8. *Q(*TEST MINDIST$_{\geq}$*)* = $\mathsf{O}(\mathsf{n}^{w/(\mathsf{w}+1)})$ *for* $w = \mathsf{O}(1)$.

Proof. A tester for TEST MINDIST$_{\geq}$ is described by the following steps.

1. Let $r = \mathsf{O}(\mathsf{n}^{w/(\mathsf{w}+1)})$.
2. Run Ambainis' quantum walk algorithm [Amb04] on the Johnson graph $J(n,r)$ whose nodes are subsets of size r.
3. If the walk finds a dependent set of size $< w$ REJECT, else ACCEPT.

Correctness of the algorithm. Note that the tester always accepts inputs H with $\mathrm{d}_H \geq w$ because, by definition, it is impossible to find a dependent set of columns of size $< w$ in such instances. In other words, this tester has *one-sided error* as described in section 2.6. Note that having one sided error is an important feature. For instance, they appear in the context of proximity oblivious testers [Gol17] and are useful for amplifying the success probability. Using the reduction in section 5.3 without the additional promise of the testing variant, we see that the tester will also reject inputs H with $\mathrm{d}_H < w$ with high probability.

Query complexity. Note that the number of queries made by the tester to the quantum version of the oracle O^H_{col} is $\mathsf{O}(\mathsf{n}^{w/(\mathsf{w}+1)})$ and therefore Q(DEPSET$_w$)=$\mathsf{O}(\mathsf{n}^{w/(\mathsf{w}+1)})$ for $w = \mathsf{O}(1)$. □

It is important to pause here and note that we have not used the additional promise provided by the testing variant i.e, TEST MINDIST$_{\geq}$ compared to the DEPSET$_w$ problem. Therefore, we conjecture that the upper bound in the previous lemma can be improved. Next, we establish lower bounds for the TEST MINDIST$_{\geq}$ problem.

Lower bounds. We now show that any classical algorithm for TEST MINDIST$_{\geq}$ requires $\Omega(\mathsf{n}^{1/2})$ queries whereas any quantum algorithm requires $\Omega(\mathsf{n}^{1/3})$ queries to the adjacency list oracle. We will achieve this by reducing COLLISION problem [Aar02; Shi02; Kut05; Amb05] to TEST MINDIST$_{\geq}$. The COLLISION problem is defined as follows:

COLLISION
Input: A function $f : [n] \to [n]$.
Promise: Either f is one-to-one or f is two-to-one.
Output: ACCEPT if f is one-to-one, REJECT otherwise.

Lemma 9. COLLISION $\leq$ TEST MINDIST$_{\geq}$. *Hence R(*TEST MINDIST$_{\geq}$*) =* $\Omega(n^{1/2})$*, and Q(*TEST MINDIST$_{\geq}$*) =* $\Omega(n^{1/3})$.

Proof. Given an input function f for the COLLISION problem, we describe a recipe to construct the corresponding instance for TEST MINDIST$_{\geq}$.

Consider any binary linear code $\mathcal{C}$ with parameters $n_{\mathcal{C}}, k_{\mathcal{C}}, d_{\mathcal{C}}$ where $d_{\mathcal{C}} > 2$. Let $H_{\mathcal{C}} \in \mathbb{F}_2^{n_{\mathcal{C}}-k_{\mathcal{C}} \times n_{\mathcal{C}}}$ be the parity check matrix for $\mathcal{C}$. There are several options for picking such a code ranging from the Hamming code [Ham50] with distance 3 to asymptotically good codes (Justesen code [Jus72] for instance) with constant relative distance. We choose any one such code with $n_{\mathcal{C}} = n$ (the length of the input sequence in the collision problem). Let $H_{\mathcal{C}}[i]$ denote the i^{th} column of the matrix $H_{\mathcal{C}}$ where $1 \leq i \leq n_{\mathcal{C}}$. The parity check matrix corresponding to the input function f is given by H_f where $H_f[j] = H_{\mathcal{C}}[f(j)]$ for all $1 \leq j \leq n_{\mathcal{C}} = n$. Now, we provide H_f to a tester for TEST MINDIST$_{\geq}$ with the parameter $w = d_{\mathcal{C}}$. If the tester accepts the input H_f, we conclude that f is one-to-one. If the tester rejects the input, we conclude that f is two-to-one.

Correctness of the reduction. Suppose the input sequence f is one-to-one. In this case, H_f is equivalent to $H_{\mathcal{C}}$ up to a permutation of columns. Since permuting the columns of a parity check matrix does not change the distance of the code, we have $d_{H_f} = d_{\mathcal{C}}$. Therefore, the tester will accept this input with certainty. On the other hand, when f is two-to-one, H_f will contain $({}^{n}\!/_{2})$ columns repeated twice. This implies $d_{H_f} = 2$ for this case. To show that the H_f constructed in this case is a valid REJECT instance for the testing problem, we need to show that $\delta_{col}(H_f, H') \geq \epsilon n_{\mathcal{C}}$ for some $\epsilon = O(1) > 0$ and for all H' such that $\mathrm{d}_{H'} \geq d_{\mathcal{C}}$.

Consider the set of pairs of indices $\{(i_1, j_1), (i_2, j_2), \ldots (i_{n/2}, j_{n/2})\}$ for which H_f has identical columns i.e., $H_f[i_l] = H_f[j_l]$ for all $1 \leq l \leq {}^{n}\!/_{2}$. The column distance, defined in section 5.2, between H_f and H' follows

$$\begin{aligned}
\delta_{col}(H_f, H') &= \sum_{l=1}^{n_{\mathcal{C}}/2} \left(|H_f[i_l] - H'[i_l]| + |H'[j_l] - H_f[j_l]|\right) \\
&= \sum_{l=1}^{n_{\mathcal{C}}/2} \left(|H_f[i_l] - H'[i_l]| + |H'[j_l] - H_f[i_l]|\right) \\
&\qquad\qquad (\because \text{ due to collision, } H_f[i_l] = H_f[j_l]) \\
&\geq \sum_{l=1}^{n_{\mathcal{C}}/2} |H'[i_l] - H'[j_l]| \qquad \text{(using the triangle inequality)} \\
&\geq \frac{n_{\mathcal{C}}}{2}.
\end{aligned} \tag{5.12}$$

Therefore, H_f is $1/2$-far with respect to *column distance* from all matrices H' for which $\mathrm{d}_{H'} \geq d_{\mathcal{C}}$. Combined with the fact that $d_{H_f} = 2 < d_{\mathcal{C}}$ makes it a valid REJECT instance. The lower bounds follow from the corresponding classical [Mun77; DM89; BH07; Fel91] and quantum [Aar02; Shi02; Kut05; Amb05] results in the literature. □

We have established that the testing version of the minimum distance problem is at least as hard as the collision problem. Although the collision problem comes with the promise of containing $n/2$ matching pairs, finding any one of these is not trivial. To find a match for any entry, one has to search through $(n-1)$ elements. Therefore, although the corresponding parity check matrix has $n/2$ dependent sets of size two each, finding them requires $\Omega(\mathsf{n}^{1/2})$ classical queries and $\Omega(\mathsf{n}^{1/3})$ quantum queries. Note that this is weaker than the $\Omega(\mathsf{n}^{2/3})$ quantum lower bound for the decision version i.e., DEPSET$_w$, where we derived the lower bound via reduction from ELEMENT DISTINCTNESS. These two bounds are related to each other. For further discussion on this, please see Refs. [AS04; Amb05].

In this section, we derived lower and upper bounds for the TEST MINDIST$_{\geq}$ problem. We believe that these bounds can be improved from both ends. Specifically, for the upper bounds, we need to find a way to use the extra promise structure imposed by the testing variant whereas for the lower bound, it'd be good to derive a bound that depends on the threshold weight parameter w. These are interesting directions for future research.

5.5 Conclusion and future work

In this chapter, we explored the query complexity of estimating the minimum distance of a linear code given access to the parity check matrix as an adjacency list oracle. We showed that this is maximally hard classically and requires $\Omega(\mathsf{n}^{2/3})$ queries using a quantum algorithm. We also show that $O(\mathsf{n}^{w/(\mathsf{w}+1)})$ queries are sufficient for this task when using a quantum algorithm. We define and explore a property testing variant of the same task where we wish to certify if a given code has high distance or is far from all codes having high distance for a natural notion of distance between codes defined via their parity check matrices. We show that $\Omega(\mathsf{n}^{1/2})$ queries are required classically and $\Omega(\mathsf{n}^{1/3})$ queries are required when using a quantum algorithm to solve the testing variant with bounded-error. The upper bound for the non-testing variant holds trivially but we conjecture that this can be improved.

The quantum query complexity upper bound obtained for $\textsc{DepSet}_w$ only holds for constant w. It would be interesting to generalize the lower and upper bounds to the setting of $w = polylog(n)$ and even $w = \Theta(n)$. This would enable testing codes with distance linear in the number of bits. Some preliminary work in this direction has been done in Refs. [BKT18; MTZ20] but the problem remains largely open. This problem has likely not received much attention in the field of mathematics due to the lack of an important application. We now highlight, through this work, that resolving these questions for non-constant w would help answer fundamental questions in coding theory.

It would also be interesting to consider promise versions of this problem whereby we are guaranteed that either a certain threshold number of dependent sets exist or there are no dependent sets of size *at most* w. Formally, the following problem is interesting and comes close in some sense to the property testing setting.

Pr-$\textsc{DepSet}(w, \epsilon)$

Input: $H \in \mathbb{F}_2^{n-k\times n}$, threshold parameter $w \in [3, n-k+1]$, threshold parameter $\epsilon > 0$.

Promise: Either $\exists$ at least ϵn^{w-1} dependent sets of size$< w$ or $\nexists$ any dependent set of size$< w$.

Output: ACCEPT if $\exists$ a dependent set of size $< w$, REJECT otherwise.

The motivation to solve the above promise version comes from a belief that the instances which are far away from the set of codes having high distance will likely contain many dependent sets of size *at most* $(w-1)$. Therefore, solving the above promise version will lead to taking advantage of the additional structure in the $\textsc{Test MinDist}_{\geq}$ problem, and improving the upper bound we derived in this chapter. The major obstacle in this approach is translating the Hamming/column distance promise in $\textsc{Test MinDist}_{\geq}$ to the guaranteed existence of a large number of dependent sets of size *at most* $(w-1)$. We believe this is a promising direction to explore nonetheless.

We feel that this chapter serves as the beginning of posing and answering a long line of interesting property testing questions for classical codes, quantum codes and quantum fault-tolerant gadgets more generally.

Chapter 6

Summary and conclusion

While quantum computing is one of the most exciting endeavours of this century, scaling quantum systems to achieve fault tolerance and perform meaningful quantum computation is one of the biggest challenges. The methods developed in the first part of the thesis attempt to push our understanding of how noise tailoring and noise characterization methods can be used to improve and diagnose the performance of quantum error correction schemes. These methods not only help us in validating fault tolerant systems efficiently but also reduce the resource overhead required to implement them in practice. The second part aims to understand the query complexity of testing fundamental properties of an error correcting code. This study opens a wide range of open problems relevant to the study of error correcting codes. It also opens some interesting problems around property testing of structured functions.

In chapter 3, we describe how to use noise tailoring, *randomized compiling* in particular, in the context of quantum error correction to enforce a stochastic noise model. Furthermore, we develop an efficient metric called the *logical estimator* which can use the data obtained from scalable noise characterization protocols such as *cycle error reconstruction* to reliably predict the performance of concatenated stabilizer codes. We show that this reliable prediction can not only to be used to assess the resource overhead required to achieve a target logical error rate, but can also be utilized to guide the optimal selection of the different components of a quantum error correction scheme such as the code and the decoder. This opens up avenues to further explore the role of modern noise characterization and noise tailoring tools for efficient diagnostics of other families of codes and fault-tolerant gadgets in general. In particular, it would be interesting to develop similar approximations for other families of codes such as the topological codes. Combining ideas from Ref. [DCP10] and the tools developed in this chapter seems to be a promising direction. Moreover, going beyond the memory model for quantum error correction for a single logical qubit, and developing such benchmarks for logical operations that include multiple logical qubits would be useful.

Chapter 4 explores the role of using *randomized compiling* to improve the *performance* of concatenated codes. We show that under a wide range of physically motivated error models, randomized compiling can improve the performance of concatenated Steane code by several orders of magnitude. The noise models we study include pure Z-rotations, rotations about arbitrary axes and a combination of stochastic and unitary noise. For the simple case of Z-rotations, we analytically show that the gain in performance increases doubly exponentially with the increase in levels of concatenation for small rotation angles. For other complex noise models, we show the existence of a threshold error rate below which the gains from randomized compiling can be arbitrarily magnified. We also study the variation of the gain and the threshold angle with the rotation axis. Since randomized compiling is a code agnostic method, this study acts as a necessary precursor to the interesting problem of designing code-dependent noise tailoring strategies to further maximize gains and consequently bring down the resource overheads. It will be worthwhile to perform realistic simulations of randomized compiling to assess how many compilations are enough in practice. Moreover, adding some noise to all the components of the quantum error correction scheme including the encoding and syndrome extraction circuits in the simulations will move these studies closer to a real-world application.

In chapter 5, we study the classical and quantum query complexity of finding the minimum distance of large binary linear codes. We prove non-trivial lower and upper bounds for this task by reducing to and from other well known problems in the literature. Furthermore, we explore a property testing variant of the same problem and provide an efficient (quantum) tester. We derive non-trivial lower bounds by reducing the COLLISION problem to the task of testing the minimum distance of a given code. To the best of our knowledge, this is a first attempt at exploring property testing of properties of codes. This opens up several interesting testing questions both for classical and quantum codes, and fault-tolerant gadgets in general. In particular, one can extend all the results presented in this chapter to the context of quantum stabilizer codes. Using recent results on the hardness of minimum distance for quantum stabilizer codes derived in Ref. [KK22] is a natural link for this problem. For testing more general fault-tolerance structures, results obtained in Ref. [YS22], which test large scale graph states, can serve as a good starting point.

References

[AAB$^+$19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019 (p. 1).

[Aar02] Scott Aaronson. Quantum lower bound for the collision problem. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, 635–642, Montreal, Quebec, Canada. Association for Computing Machinery, 2002 (pp. 84, 94, 96).

[ABJ$^+$03] J. B. Altepeter, D. Branning, E. Jeffrey, T. C. Wei, P. G. Kwiat, R. T. Thew, J. L. O'Brien, M. A. Nielsen, and A. G. White. Ancilla-assisted quantum process tomography. *Phys. Rev. Lett.*, 90:193601, 19, 2003 (p. 22).

[ABO08] D. Aharonov and M. Ben-Or. Fault-tolerant quantum computation with constant error rate. *SIAM Journal on Computing*, 38(4):1207–1282, 2008. eprint: `https://doi.org/10.1137/S0097539799359385` (p. 20).

[AGP07] Panos Aliferis, Daniel Gottesman, and John Preskill. Accuracy threshold for postselected quantum computation. *Quantum Information and Computation*, 8(3&4):0181–0244, 2007 (pp. 2, 38, 48).

[Amb04] A. Ambainis. Quantum walk algorithm for element distinctness. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 22–31, Los Alamitos, CA, USA. IEEE Computer Society, 2004 (pp. 84, 90, 92, 94).

[Amb05] Andris Ambainis. Polynomial degree and lower bounds in quantum complexity: collision and element distinctness with small range. *Theory of Computing*, 1(3):37–46, 2005 (pp. 84, 94, 96).

[ANR74] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974 (p. 79).

[AP08] Panos Aliferis and John Preskill. Fault-tolerant quantum computation against biased noise. *Phys. Rev. A*, 78:052331, 5, 2008 (p. 48).

[AP09] Panos Aliferis and John Preskill. Fibonacci scheme for fault-tolerant quantum computation. *Physical Review A*, 79(1), 2009 (p. 20).

[AS04] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, 2004 (p. 96).

[AS08] Noga Alon and Asaf Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM Journal on Computing*, 37(6):1703–1727, 2008. eprint: `https://doi.org/10.1137/06064888X` (p. 47).

[BATB$^+$21] J Pablo Bonilla Ataides, David K Tuckett, Stephen D Bartlett, Steven T Flammia, and Benjamin J Brown. The xzzx surface code. *Nat. Commun.*, 12:2172, 2021 (p. 48).

[BBP$^+$96] Charles H. Bennett, Gilles Brassard, Sandu Popescu, Benjamin Schumacher, John A. Smolin, and William K. Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Phys. Rev. Lett.*, 76:722–725, 5, 1996 (p. 66).

[BDS$^+$96] Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed-state entanglement and quantum error correction. *Phys. Rev. A*, 54:3824–3851, 5, 1996 (p. 66).

[BEK$^+$18] Sergey Bravyi, Matthias Englbrecht, Robert König, and Nolan Peard. Correcting coherent errors with surface codes. *npj Quantum Information*, 4(1), 2018 (pp. 37, 39, 66, 126).

[Bel12a] Aleksandrs Belovs. Learning-graph-based quantum algorithm for k-distinctness. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 207–216, 2012 (p. 90).

[Bel12b] Aleksandrs Belovs. Span programs for functions with constant-sized 1-certificates: extended abstract. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '12, 77–84, New York, New York, USA. Association for Computing Machinery, 2012 (p. 90).

[BH07] Mario Cortina Borja and John Haigh. The Birthday Problem. *Significance*, 4(3):124–127, August 2007. eprint: `https://academic.oup.com/jrssig/article-pdf/4/3/124/49109395/sign\_4\_3\_124.pdf` (p. 96).

[BKGN$^+$17] Robin Blume-Kohout, John King Gamble, Erik Nielsen, Kenneth Rudinger, Jonathan Mizrahi, Kevin Fortier, and Peter Maunz. Demonstration of qubit operations below a rigorous fault tolerance threshold with gate set tomography. *Nature Communications*, 8:EP–, 2017. Article (p. 3).

[BKT18] Mark Bun, Robin Kothari, and Justin Thaler. The polynomial method strikes back: tight quantum query bounds via dual polynomials. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, 297–310, Los Angeles, CA, USA. Association for Computing Machinery, 2018 (pp. 84, 90, 97).

[BL05] Samuel L. Braunstein and Peter van Loock. Quantum information with continuous variables. *Rev. Mod. Phys.*, 77:513–577, 2, 2005 (p. 1).

[BL11] Aleksandrs Belovs and Troy Lee. Quantum algorithm for k-distinctness with prior knowledge on the input, 2011. arXiv: `1108.3022 [quant-ph]` (p. 90).

[BL17] Daniel J Bernstein and Tanja Lange. Post-quantum cryptography. *Nature*, 549(7671):188–194, 2017 (p. 1).

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993 (p. 44).

[BMT78] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems (corresp.) *IEEE Transactions on Information Theory*, 24(3):384–386, 1978 (p. 79).

[BRS$^+$09] Stephen D. Bartlett, Terry Rudolph, Robert W. Spekkens, and Peter S. Turner. Quantum communication using a bounded-size quantum reference frame. *New Journal of Physics*, 11(6):063013, June 2009 (p. 71).

[BS13] Aleksandrs Belovs and Robert Spalek. Adversary lower bound for the k-sum problem. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, 323–328, Berkeley, California, USA. Association for Computing Machinery, 2013 (p. 92).

[BSS$^+$03] Paul Beame, Michael E. Saks, Xiaodong Sun, and Erik Vee. Time-space trade-off lower bounds for randomized computation of decision problems. *J. ACM*, 50:154–195, 2003 (pp. 84, 90).

[BTL10] Sergey Bravyi, Barbara M Terhal, and Bernhard Leemhuis. Majorana fermion codes. *New Journal of Physics*, 12(8):083039, 2010 (p. 83).

[BV13] Sergey Bravyi and Alexander Vargo. Simulation of rare events in quantum error correction. *Phys. Rev. A*, 88:062308, 6, 2013 (p. 133).

[BV93] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 11–20, 1993 (p. 1).

[BW23] Stefanie J. Beale and Joel J. Wallman. Randomized compiling for subsystem measurements, 2023. arXiv: `2304.06599 [quant-ph]` (p. 27).

[BWG$^+$18] Stefanie J. Beale, Joel J. Wallman, Mauricio Gutiérrez, Kenneth R. Brown, and Raymond Laflamme. Quantum error correction decoheres noise. *Phys. Rev. Lett.*, 121:190501, 19, 2018 (pp. 37, 67, 139).

[BYJK$^+$04] Ziv Bar-Yossef, T.S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004. Special Issue on FOCS 2002 (p. 89).

[CB19] Zhenyu Cai and Simon C. Benjamin. Constructing Smaller Pauli Twirling Sets for Arbitrary Error Channels. *Scientific Reports*, 9(1):11281, August 2019 (p. 66).

[CBB$^+$22] 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 preprint arXiv:2210.00921*, 2022 (p. 2).

[CDAE19] Arnaud Carignan-Dugas, Matthew Alexander, and Joseph Emerson. A polar decomposition for quantum channels (with applications to bounding error propagation in quantum circuits). *Quantum*, 3:173, August 2019 (pp. 74, 76, 141).

[CDDH$^+$23] 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, 2023. arXiv: `2303.17714 [quant-ph]` (pp. 2, 4, 27–29, 31, 49, 61, 73).

[CDWE19] Arnaud Carignan-Dugas, Joel J Wallman, and Joseph Emerson. Bounding the average gate fidelity of composite channels using the unitarity. *New Journal of Physics*, 21(5):053016, 2019 (p. 76).

[CE05] A. M. Childs and J. M. Eisenberg. Quantum algorithms for subset finding. *Quantum Information and Computation*, 5(7):593–604, 2005 (pp. 91, 92).

[Cho75] Man-Duen Choi. Completely positive linear maps on complex matrices. *Linear Algebra and its Applications*, 10(3):285–290, 1975 (pp. 7, 12).

[CIP18] Christopher Chamberland, Pavithran Iyer, and David Poulin. Fault-tolerant quantum computing in the Pauli or Clifford frame with slow error diagnostics. *Quantum*, 2:43, 2018 (p. 50).

[CMN$^+$18] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, 2018 (p. 2).

[CN97] Isaac L. Chuang and M. A. Nielsen. Prescription for experimental determination of the dynamics of a quantum black box. *Journal of Modern Optics*, 44(11-12):2455–2467, 1997. eprint: https://www.tandfonline.com/doi/pdf/10.1080/09500349708231894 (p. 22).

[CR18a] Christopher Chamberland and Pooya Ronagh. Deep neural decoders for near term fault-tolerant experiments. *Quantum Science and Technology*, 3(4):044002, 2018 (p. 63).

[CR18b] Rui Chao and Ben W Reichardt. Fault-tolerant quantum computation with few qubits. *NPJ Quantum Information*, 4(1):42, 2018 (p. 2).

[CSS$^+$09] Andrew Cross, Graeme Smith, John A. Smolin, and Bei Zeng. Codeword stabilized quantum codes. *IEEE Transactions on Information Theory*, 55(1):433–438, 2009 (p. 82).

[CTV17] Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. Roads towards fault-tolerant universal quantum computation. *Nature*, 549(7671):172–179, 2017 (pp. 2, 48).

[CWB$^+$17] Christopher Chamberland, Joel Wallman, Stefanie Beale, and Raymond Laflamme. Hard decoding algorithm for optimizing thresholds under general markovian noise. *Phys. Rev. A*, 95:042332, 4, 2017 (pp. 37, 39, 68, 76, 119).

[CZ95] J. I. Cirac and P. Zoller. Quantum computations with cold trapped ions. *Phys. Rev. Lett.*, 74:4091–4094, 20, 1995 (p. 1).

[DA07] David P. DiVincenzo and Panos Aliferis. Effective fault-tolerant quantum computation with slow measurements. *Phys. Rev. Lett.*, 98:020501, 2, 2007 (p. 50).

[DCE$^+$09] Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Phys. Rev. A*, 80:012304, 1, 2009 (pp. 3, 22).

[DCP10] Guillaume Duclos-Cianci and David Poulin. A renormalization group decoding algorithm for topological quantum codes. In *2010 IEEE Information Theory Workshop*, pages 1–5, 2010 (pp. 98, 126).

[Deu85] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985 (p. 1).

[DLP01] G. M. D'Ariano and P. Lo Presti. Quantum tomography for measuring experimentally the matrix elements of an arbitrary quantum operation. *Phys. Rev. Lett.*, 86:4195–4198, 19, 2001 (p. 22).

[DM89] Persi Diaconis and Frederick Mosteller. Methods for studying coincidences. *Journal of the American Statistical Association*, 84(408):853–861, 1989. eprint: `https://www.tandfonline.com/doi/pdf/10.1080/01621459.1989.10478847` (p. 96).

[DMS99] I. Dumer, D. Micciancio, and M. Sudan. Hardness of approximating the minimum distance of a linear code. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pages 475–484, 1999 (p. 82).

[DP17] Andrew S. Darmawan and David Poulin. Tensor-network simulations of the surface code under realistic noise. *Physical Review Letters*, 119(4), 2017 (p. 37).

[DPM+20] Poulami Das, Christopher A. Pattison, Srilatha Manne, Douglas Carmean, Krysta Svore, Moinuddin Qureshi, and Nicolas Delfosse. A scalable decoder micro-architecture for fault-tolerant quantum computing. *arXiv:2001.06598*, 2020 (p. 63).

[DZP19] David Davalos, Mario Ziman, and Carlos Pineda. Divisibility of qubit channels and dynamical maps. *Quantum*, 3:144, May 2019 (p. 3).

[EAŻ05] Joseph Emerson, Robert Alicki, and Karol Życzkowski. Scalable noise estimation with random unitary operators. *Journal of Optics B: Quantum and Semiclassical Optics*, 7(10):S347–S352, 2005 (pp. 3, 20, 22).

[EWP+19] Alexander Erhard, Joel J. 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. *Nature Communications*, 10(1), 2019 (pp. 4, 28, 49).

[Fel91] William Feller. *An introduction to probability theory and its applications, Volume 2*, volume 81. John Wiley & Sons, 1991 (p. 96).

[Fey82] Richard P Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6/7):467–488, 1982 (p. 1).

[FG99] C.A. Fuchs and J. van de Graaf. Cryptographic distinguishability measures for quantum-mechanical states. *IEEE Transactions on Information Theory*, 45(4):1216–1227, 1999 (p. 18).

[FHV+22] 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, 2022. arXiv: `2201.10672 [quant-ph]` (p. 27).

[FKD19] Samuele Ferracin, Theodoros Kapourniotis, and Animesh Datta. Accrediting outputs of noisy intermediate-scale quantum computing devices. *New Journal of Physics*, 21(11):113038, 2019 (p. 27).

[FKR+04] Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *Journal of Computer and System Sciences*, 68(4):753–787, 2004. Special Issue on FOCS 2002 (p. 45).

[Fla21] Steven T. Flammia. Averaged circuit eigenvalue sampling, 2021. arXiv: `2108.05803 [quant-ph]` (p. 28).

[FS95] K. Friedl and M. Sudan. Some improvements to total degree tests. In *Proceedings Third Israel Symposium on the Theory of Computing and Systems*, pages 190–198, 1995 (p. 83).

[Fuj12] Hachiro Fujita. Quantum mceliece public-key cryptosystem. *Quantum Info. Comput.*, 12(3–4):181–202, 2012 (p. 82).

[FW20] Steven T. Flammia and Joel J. Wallman. Efficient estimation of pauli channels. *ACM Transactions on Quantum Computing*, 1(1), December 2020 (pp. 30, 49).

[GB15] Mauricio Gutiérrez and Kenneth R. Brown. Comparison of a quantum error-correction threshold for exact and approximate errors. *Phys. Rev. A*, 91:022335, 2, 2015 (p. 119).

[GD17] Daniel Greenbaum and Zachary Dutton. Modeling coherent errors in quantum error correction. *Quantum Science and Technology*, 3(1):015007, 2017 (pp. 66, 69, 126).

[GGK19] Oded Goldreich, Tom Gur, and Ilan Komargodski. Strong locally testable codes with relaxed local decoders. *ACM Trans. Comput. Theory*, 11(3), 2019 (p. 83).

[GGR96] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 339–348, 1996 (p. 47).

[GKH$^+$96] Dima Grigoriev, Marek Karpinski, Friedhelm Meyer auf der Heide, and Roman Smolensky. A lower bound for randomized algebraic decision trees. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*, pages 612–619, 1996 (pp. 84, 90).

[GM19] Jérémie Guillaud and Mazyar Mirrahimi. Repetition cat qubits for fault-tolerant quantum computation. *Phys. Rev. X*, 9:041053, 4, 2019 (p. 48).

[GNPS18] Lorenzo Grassi, María Naya-Plasencia, and André Schrottenloher. Quantum algorithms for the $$k$$-xor problem. In *Lecture Notes in Computer Science*, pages 527–559. Springer International Publishing, 2018 (p. 83).

[Gol17] Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017 (pp. 41–44, 94).

[Gol49] Marcel JE Golay. Notes on digital coding. *Proc. IEEE*, 37:657, 1949 (p. 79).

[Got14] Daniel Gottesman. Fault-tolerant quantum computation with constant overhead. *Quantum Info. Comput.*, 14(15–16):1338–1372, 2014 (p. 2).

[Got97] Daniel Eric Gottesman. *Stabilizer Codes and Quantum Error Correction*. PhD thesis, California Institute of Technology, 1997 (pp. 23, 36, 39).

[GS06] Oded Goldreich and Madhu Sudan. Locally testable codes and pcps of almost-linear length. *J. ACM*, 53(4):558–655, 2006 (p. 83).

[GSL$^+$16] Mauricio Gutiérrez, Conor Smith, Livia Lulushi, Smitha Janardan, and Kenneth R. Brown. Errors and pseudothresholds for incoherent and coherent noise. *Phys. Rev. A*, 94:042338, 4, 2016 (pp. 37, 39, 66, 68).

[Gut12] Gus Gutoski. On a measure of distance for quantum strategies. *Journal of Mathematical Physics*, 53(3):032202, 2012. eprint: `https://doi.org/10.1063/1.3693621` (p. 19).

[Ham50] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950 (pp. 79, 95).

[Hay17] Masahito Hayashi. State evolution and trace-preserving completely positive maps. In *Quantum Information Theory*, pages 197–251. Springer, 2017 (p. 7).

[HDF19] Eric Huang, Andrew C. Doherty, and Steven Flammia. Performance of quantum error correction with coherent errors. *Phys. Rev. A*, 99:022313, 2, 2019 (pp. 37, 126, 137).

[HFE04] Xiao-Yu Hu, M.P.C. Fossorier, and E. Eleftheriou. On the computation of the minimum distance of low-density parity-check codes. In *2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577)*, volume 2, 767–771 Vol.2, 2004 (p. 82).

[HFW20] Robin Harper, Steven T. Flammia, and Joel J. Wallman. Efficient learning of quantum noise. *Nature Physics*, 16(12):1184–1188, 2020 (p. 4).

[HGT+21] Stuart Harwood, Claudio Gambella, Dimitar Trenev, Andrea Simonetto, David Bernal, and Donny Greenberg. Formulating and solving routing problems on quantum computers. *IEEE Transactions on Quantum Engineering*, 2:1–17, 2021 (p. 1).

[HJ85] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985 (p. 10).

[HK96] G.B. Horn and F.R. Kschischang. On the intractability of permuting a block code to minimize trellis complexity. *IEEE Transactions on Information Theory*, 42(6):2042–2048, 1996 (p. 80).

[HLG11] Min-Hsiu Hsieh and François Le Gall. NP-hardness of decoding quantum error-correction codes. *Physical Review A*, 83(5):052331, May 2011 (pp. 38, 39, 82).

[HNM+21] Akel Hashim, Ravi K. Naik, Alexis Morvan, Jean-Loup Ville, Bradley Mitchell, John Mark Kreikebaum, Marc Davis, Ethan Smith, Costin Iancu, Kevin P. O'Brien, Ian Hincks, Joel J. Wallman, Joseph Emerson, and Irfan Siddiqi. Randomized Compiling for Scalable Quantum Computing on a Noisy Superconducting Quantum Processor. *Physical Review X*, 11(4):041039, November 2021 (pp. 27, 64, 68, 73).

[HRP+06] T. Hime, P. A. Reichardt, B. L. T. Plourde, T. L. Robertson, C.-E. Wu, A. V. Ustinov, and John Clarke. Solid-state qubits with current-controlled coupling. *Science*, 314(5804):1427–1429, 2006. eprint: `https://www.science.org/doi/pdf/10.1126/science.1134388` (p. 1).

[HYF21] Robin Harper, Wenjun Yu, and Steven T. Flammia. Fast estimation of sparse quantum noise. *PRX Quantum*, 2:010322, 1, 2021 (p. 61).

[IJB+22] Pavithran Iyer, Aditya Jain, Stephen D. Bartlett, and Joseph Emerson. Efficient diagnostics for quantum error correction. *Phys. Rev. Res.*, 4:043218, 4, 2022 (pp. iv, 41, 48, 65, 67).

[IP15] Pavithran S. Iyer and David Poulin. Hardness of decoding quantum stabilizer codes. *IEEE Transactions on Information Theory*, 61(9):5209–5223, 2015 (pp. 38, 82).

[IP18] Pavithran Iyer and David Poulin. A small quantum computer is needed to optimize fault-tolerant protocols. *Quantum Science and Technology*, 3(3):030504, 2018 (pp. 3, 37, 39, 49, 57, 59, 131, 133, 137, 140).

[IP20] Joseph K. Iverson and John Preskill. Coherence in logical quantum channels. *New Journal of Physics*, 22(7):073066, July 2020 (p. 66).

[Iye18] Pavithran Iyer. *Une analyse critique de la correction d'erreurs quantiques pour du bruit rèaliste*. PhD thesis, Université de Sherbrooke, 2018 (pp. 3, 4, 131).

[JIB+23] Aditya Jain, Pavithran Iyer, Stephen D. Bartlett, and Joseph Emerson. Improved quantum error correction with randomized compiling. *Phys. Rev. Res.*, 5:033049, 3, 2023 (pp. iv, 65).

[JOL14] Tomas Jochym-O'Connor and Raymond Laflamme. Using concatenated quantum codes for universal fault-tolerant quantum gates. *Phys. Rev. Lett.*, 112:010505, 1, 2014 (p. 38).

[JRO+17] Peter D. Johnson, Jonathan Romero, Jonathan Olson, Yudong Cao, and Alán Aspuru-Guzik. Qvector: an algorithm for device-tailored quantum error correction. *arXiv:1711.02249*, 2017 (p. 63).

[Jus72] J. Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18(5):652–656, 1972 (p. 95).

[JZ22] Stacey Jeffery and Sebastian Zur. Multidimensional quantum walks, with application to k-distinctness, 2022. eprint: `arXiv:2208.13492` (p. 90).

[KBD+83] Karl Kraus, Arno Böhm, John D Dollard, and WH Wootters. *States, Effects, and Operations Fundamental Notions of Quantum Theory: Lectures in Mathematical Physics at the University of Texas at Austin*. Springer, 1983 (p. 14).

[Kir18] Elena Kirshanova. Improved quantum information set decoding. In *Post-Quantum Cryptography*, pages 507–527. Springer International Publishing, 2018 (p. 83).

[Kit97a] A Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191, 1997 (p. 19).

[Kit97b] A. Yu. Kitaev. *Quantum error correction with imperfect gates*. In *Quantum Communication, Computing, and Measurement*. O. Hirota, A. S. Holevo, and C. M. Caves, editors. Springer US, Boston, MA, 1997, pages 181–188 (p. 19).

[KK22] Upendra Kapshikar and Srijita Kundu. Diagonal distance of quantum codes and hardness of the minimum distance problem, 2022. arXiv: `2203.04262 [quant-ph]` (pp. 82, 99).

[KL20] Kao-Yueh Kuo and Chung-Chin Lu. On the hardnesses of several quantum decoding problems. en. *Quantum Information Processing*, 19(4):123, February 2020 (p. 82).

[KL96] Emanuel Knill and Raymond Laflamme. Concatenated quantum codes. *arXiv:quant-ph/9608012*, 1996 (p. 38).

[KLR+08] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland. Randomized benchmarking of quantum gates. *Phys. Rev. A*, 77:012307, 1, 2008 (p. 20).

[Kot14] Robin Kothari. Efficient algorithms in quantum query complexity, 2014 (p. 85).

[KRK+05] Navin Khaneja, Timo Reiss, Cindie Kehlet, Thomas Schulte-Herbrüggen, and Steffen J. Glaser. Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms. *Journal of Magnetic Resonance*, 172(2):296–305, 2005 (p. 126).

[KS92] Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992. eprint: `https://doi.org/10.1137/0405044` (p. 89).

[KSV02] A.Y. Kitaev, A. Shen, and M.N. Vyalyi. *Classical and Quantum Computation*. Graduate studies in mathematics. American Mathematical Society, 2002 (p. 19).

[Kut05] Samuel Kutin. Quantum lower bound for the collision problem with small range. *Theory of Computing*, 1(1):29–36, 2005 (pp. 84, 94, 96).

[LB13] D.A. Lidar and T.A. Brun. *Quantum Error Correction*. Cambridge University Press, 2013 (pp. 37, 50).

[MAB20] Swarnadeep Majumder, Leonardo Andreta de Castro, and Kenneth R. Brown. Real-time calibration with spectator qubits. *npj Quantum Information*, 6(1), 2020 (p. 126).

[Mag08] Magesan, Easwar. *Gaining Information About a Quantum Channel Via Twirling*. Master's thesis, 2008 (p. 22).

[Mas78] James L Massey. Foundation and methods of channel encoding. In *Proc. Int. Conf. Information Theory and Systems*, volume 65, pages 148–157. NTG-Fachberichte, 1978 (p. 80).

[MB14] J. True Merrill and Kenneth R. Brown. *Progress in compensating pulse sequences for quantum computation*. In *Quantum Information and Computation for Chemistry*. Stuart A. Rice Aaron R. Dinner, editor. Advances in Chemical Physics. John Wiley & Sons, Ltd, 2014, pages 241–294. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118742631.ch10 (p. 126).

[MC13] Easwar Magesan and Paola Cappellaro. Experimentally efficient methods for estimating the performance of quantum measurements. *Phys. Rev. A*, 88:022127, 2, 2013 (p. 49).

[MGE11] Easwar Magesan, J. M. Gambetta, and Joseph Emerson. Scalable and robust randomized benchmarking of quantum processes. *Phys. Rev. Lett.*, 106:180504, 18, 2011 (pp. 3, 20, 23, 24).

[MGE12] Easwar Magesan, Jay M. Gambetta, and Joseph Emerson. Characterizing quantum gates via randomized benchmarking. *Phys. Rev. A*, 85:042311, 4, 2012 (p. 20).

[MGJ+12] Easwar Magesan, Jay M. Gambetta, B. R. Johnson, Colm A. Ryan, Jerry M. Chow, Seth T. Merkel, Marcus P. da Silva, George A. Keefe, Mary B. Rothwell, Thomas A. Ohki, Mark B. Ketchen, and M. Steffen. Efficient measurement of quantum gate error by interleaved randomized benchmarking. *Phys. Rev. Lett.*, 109:080505, 8, 2012 (p. 23).

[MGS+13] Seth T. Merkel, Jay M. Gambetta, John A. Smolin, Stefano Poletto, Antonio D. Córcoles, Blake R. Johnson, Colm A. Ryan, and Matthias Steffen. Self-consistent quantum process tomography. *Phys. Rev. A*, 87:062119, 6, 2013 (p. 28).

[MS77] FJ McWilliams and NJA Sloane. The theory of error-correcting codes, 1977 (p. 81).

[MTZ20] Nikhil S. Mande, Justin Thaler, and Shuchen Zhu. Improved Approximate Degree Bounds for k-Distinctness. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*, volume 158 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 2:1–2:22, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020 (pp. 90, 97).

[Mun77] A. G. Munford. A note on the uniformity assumption in the birthday problem. *The American Statistician*, 31(3):119–119, 1977. eprint: https://www.tandfonline.com/doi/pdf/10.1080/00031305.1977.10479214 (p. 96).

[NBS+19] Murphy Yuezhen Niu, Sergio Boixo, Vadim N. Smelyanskiy, and Hartmut Neven. Universal quantum control through deep reinforcement learning. *npj Quantum Information*, 5(1), 2019 (p. 126).

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010 (pp. 1, 7, 14).

[NDD+19] Hendrik Poulsen Nautrup, Nicolas Delfosse, Vedran Dunjko, Hans J. Briegel, and Nicolai Friis. Optimizing Quantum Error Correction Codes with Reinforcement Learning. *Quantum*, 3:215, December 2019 (p. 63).

[NGR+21] Erik Nielsen, John King Gamble, Kenneth Rudinger, Travis Scholten, Kevin Young, and Robin Blume-Kohout. Gate set tomography. *Quantum*, 5:557, 2021 (p. 28).

[NH81] S. Ntafos and S. Hakimi. On the complexity of some coding problems (corresp.) *IEEE Transactions on Information Theory*, 27(6):794–796, 1981 (p. 80).

[Nie02] Michael A Nielsen. A simple formula for the average gate fidelity of a quantum dynamical operation. *Physics Letters A*, 303(4):249–252, 2002 (p. 22).

[Nie96] M. A. Nielsen. The entanglement fidelity and quantum error correction. *arXiv:quant-ph/9606012*, 1996 (p. 19).

[OIS+22] Thomas E. O'Brien, Lev B. Ioffe, Yuan Su, David Fushman, Hartmut Neven, Ryan Babbush, and Vadim Smelyanskiy. Quantum computation of molecular structure using data from challenging-to-classically-simulate nuclear magnetic resonance experiments. *PRX Quantum*, 3:030345, 3, 2022 (p. 1).

[OML19] Román Orús, Samuel Mugel, and Enrique Lizaso. Quantum computing for finance: overview and prospects. *Reviews in Physics*, 4:100028, 2019 (p. 1).

[PCZ97] 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, 2, 1997 (p. 22).

[Pou06] David Poulin. Optimal and efficient decoding of concatenated quantum block codes. *Physical Review A*, 74(5), 2006 (pp. 37, 39, 50).

[Pre23] John Preskill. Quantum computing 40 years later, 2023. arXiv: `2106.10522 [quant-ph]` (p. 1).

[Pre99] John Preskill. Lecture notes for physics 219: quantum computation. *Caltech Lecture Notes*:7, 1999 (pp. 1, 7).

[PRS02] Michal Parnas, Dana Ron, and Alex Samorodnitsky. Testing basic boolean formulae. *SIAM Journal on Discrete Mathematics*, 16(1):20–46, 2002. eprint: `https://doi.org/10.1137/S0895480101407444` (p. 45).

[PSL13] Gerardo A. Paz-Silva and D. A. Lidar. Optimally combining dynamical decoupling and quantum error correction. *Scientific Reports*, 3(1), 2013 (p. 126).

[Rag01] Maxim Raginsky. A fidelity measure for quantum channels. *Physics Letters A*, 290(1):11–18, 2001 (p. 19).

[Rau12] Robert Raussendorf. Key ideas in quantum error correction. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1975):4541–4565, 2012. eprint: `https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.2011.0494` (p. 38).

[Raz92] A.A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992 (p. 89).

[RDM02] Benjamin Rahn, Andrew C. Doherty, and Hideo Mabuchi. Exact performance of concatenated quantum codes. *Phys. Rev. A.*, 66:032304, 3, 2002 (pp. 37, 39, 119).

[RGB$^+$17] Alan Robertson, Christopher Granade, Stephen D. Bartlett, and Steven T. Flammia. Tailored codes for small quantum memories. *Phys. Rev. Applied*, 8:064004, 6, 2017 (pp. 48, 62, 63).

[RH12] Angel Rivas and Susana F Huelga. *Open quantum systems*, volume 10. Springer, 2012 (p. 6).

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996. eprint: `https://doi.org/10.1137/S0097539793255151` (p. 83).

[Sch96] Benjamin Schumacher. Sending entanglement through noisy quantum channels. *Phys. Rev. A*, 54:2614–2628, 4, 1996 (p. 19).

[SDT07] Krysta M. Svore, David P. Divincenzo, and Barbara M. Terhal. Noise threshold for a fault-tolerant two-dimensional lattice architecture. *Quantum Info. Comput.*, 7(4):297–318, 2007 (p. 20).

[SER+23] VV Sivak, Alec Eickbusch, Baptiste Royer, Shraddha Singh, Ioannis Tsioutsios, Suhas Ganjam, Alessandro Miano, BL Brock, AZ Ding, Luigi Frunzio, et al. Real-time quantum error correction beyond break-even. *Nature*, 616(7955):50–55, 2023 (p. 2).

[Sha48] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948 (pp. 2, 79).

[She14] Alexander A. Sherstov. Communication complexity theory: thirty-five years of set disjointness. In *Mathematical Foundations of Computer Science 2014*, pages 24–43. Springer Berlin Heidelberg, 2014 (p. 88).

[She20] Alexander A. Sherstov. Algorithmic polynomials. *SIAM Journal on Computing*, 49(6):1173–1231, 2020. arXiv: `1801.04607` (p. 90).

[Shi02] Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* Pages 513–519, 2002 (pp. 84, 94, 96).

[Sho99] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999 (p. 1).

[Sim97] Daniel R Simon. On the power of quantum computation. *SIAM journal on computing*, 26(5):1474–1483, 1997 (p. 1).

[SJG20] Milap Sheth, Sara Zafar Jafarzadeh, and Vlad Gheorghiu. Neural ensemble decoding for topological quantum error-correcting codes. *Phys. Rev. A*, 101:032338, 3, 2020 (p. 63).

[SLS+21] K. J. Satzinger et al. Realizing topologically ordered states on a quantum processor. *Science*, 374(6572):1237–1241, 2021. eprint: `https://www.science.org/doi/pdf/10.1126/science.abi8378` (p. 64).

[Ste06] Andrew M. Steane. A tutorial on quantum error correction. In D. L. Shepelyansky G. Casati and P. Zoller, editors, *Quantum Computers, Algorithms and Chaos*, pages 1–32. IOS Press, 2006 (p. 38).

[TBF18] David K. Tuckett, Stephen D. Bartlett, and Steven T. Flammia. Ultrahigh error threshold for surface codes with biased noise. *Phys. Rev. Lett.*, 120:050505, 5, 2018 (pp. 3, 48).

[TBF+20] David K Tuckett, Stephen D Bartlett, Steven T Flammia, and Benjamin J Brown. Fault-tolerant thresholds for the surface code in excess of 5% under biased noise. *Phys. Rev. Lett.*, 124(13):130501, 2020 (p. 48).

[TLG+18] Colin J Trout, Muyuan Li, Mauricio Gutiérrez, Yukai Wu, Sheng-Tao Wang, Luming Duan, and Kenneth R Brown. Simulating the performance of a distance-3 surface code in a linear ion trap. *New Journal of Physics*, 20(4):043038, 2018 (p. 133).

[TP14] Andreas M. Tillmann and Marc E. Pfetsch. The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. *IEEE Transactions on Information Theory*, 60(2):1248–1259, 2014 (p. 82).

[TS14] Yu Tomita and Krysta M. Svore. Low-distance surface codes under realistic quantum noise. *Phys. Rev. A*, 90:062320, 6, 2014 (p. 39).

[Tuc20] David Kingsley Tuckett. *Tailoring surface codes: Improvements in quantum error correction with biased noise*. PhD thesis, University of Sydney, 2020. (qecsim: `https://github.com/qecsim/qecsim`) (pp. 128, 129, 131).

[Var97] A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory*, 43(6):1757–1766, 1997 (pp. 79, 81).

[Var98] Alexander Vardy. Trellis structure of codes. *Handbook of coding theory*, 1998 (p. 80).

[VBB22] Florian Venn, Jan Behrends, and Benjamin Béri. Coherent error threshold for surface codes from majorana delocalization, 2022. arXiv: `2211.00655 [quant-ph]` (p. 66).

[Vit67] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967 (p. 80).

[Wag02] David Wagner. A generalized birthday problem. In *Advances in Cryptology — CRYPTO 2002*, pages 288–304. Springer Berlin Heidelberg, 2002 (p. 83).

[Wal15] Joel J. Wallman. Bounding experimental quantum error rates relative to fault-tolerant thresholds, 2015 (p. 69).

[Wal18] Joel J. Wallman. Randomized benchmarking with gate-dependent noise. *Quantum*, 2:47, January 2018 (p. 24).

[Wat09] John Watrous. Semidefinite programs for completely bounded norms. *Theory of Computing*, 5:217–238, 2009 (p. 19).

[Wat18] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018 (pp. 7, 14, 17, 19, 21).

[WBC15] Christopher J. Wood, Jacob D. Biamonte, and David G. Cory. Tensor networks and graphical calculus for open quantum systems. *Quant. Inf. Comp*, 11:0579–0811, 2015 (pp. 12, 13, 66).

[WBP15] Paul Webster, Stephen D. Bartlett, and David Poulin. Reducing the overhead for quantum computation when noise is biased. *Phys. Rev. A*, 92:062309, 6, 2015 (p. 2).

[WE16] Joel J. Wallman and Joseph Emerson. Noise tailoring for scalable quantum computation via randomized compiling. *Phys. Rev. A*, 94:052325, 5, 2016 (pp. 2, 4, 24–26, 49–51, 65).

[WF14] Joel J. Wallman and Steven T. Flammia. Randomized benchmarking with confidence. *New Journal of Physics*, 16(10):103032, October 2014 (p. 20).

[WG18] Christopher J. Wood and Jay M. Gambetta. Quantification and characterization of leakage errors. *Phys. Rev. A*, 97:032306, 3, 2018 (p. 23).

[WGH$^+$15] Joel Wallman, Chris Granade, Robin Harper, and Steven T Flammia. Estimating the coherence of noise. *New Journal of Physics*, 17(11):113020, 2015 (p. 23).

[WHE$^+$04] Yaakov S. Weinstein, Timothy F. Havel, Joseph Emerson, Nicolas Boulant, Marcos Saraceno, Seth Lloyd, and David G. Cory. Quantum process tomography of the quantum fourier transform. *The Journal of Chemical Physics*, 121(13):6117–6133, 2004. eprint: `https://doi.org/10.1063/1.1785151` (p. 22).

[Woo09] Christopher Wood. *Non-Completely Positive Maps: Properties and Applications*. PhD thesis, Macquarie University, 2009 (p. 3).

[Yao94] Andrew Chi-Chih Yao. Near-optimal time-space tradeoff for element distinctness. *SIAM Journal on Computing*, 23(5):966–975, 1994 (pp. 84, 90).

[YK22] Hayata Yamasaki and Masato Koashi. Time-efficient constant-space-overhead fault-tolerant quantum computation, 2022. arXiv: `2207.08826 [quant-ph]` (p. 2).

[YS22] Hayata Yamasaki and Sathyawageeswar Subramanian. Constant-time one-shot testing of large-scale graph states, 2022. arXiv: 2201.11127 [quant-ph] (p. 99).

[YWL10] Wen Yang, Zhen-Yu Wang, and Ren-Bao Liu. Preserving qubit coherence by dynamical decoupling. *Frontiers of Physics*, 6(1):2–14, 2010 (p. 126).

[ZDQ+21] Han-Sen Zhong et al. Phase-programmable gaussian boson sampling using stimulated squeezed light. *Phys. Rev. Lett.*, 127:180502, 18, 2021 (p. 1).

APPENDICES

Appendix A

Extending logical estimator to surface codes and faster simulations

The purpose of the appendix is to aid and enhance the understanding of the content in chapter 3. It is organized as follows. Section A.1 presents a derivation of an expression for the logical fidelity for a generic stabilizer code, as a function of the code properties as well as the parameters of the underlying physical noise process. In section A.2, we derive the accuracy of the logical estimator in estimating the logical error rate. In section A.3, we show how our tool improves the predictability of logical performance under coherent errors. While all of the above mentioned studies focus on the family of concatenated codes, section A.4 discusses how our studies can be applied to predict the performance of surface codes. Sections A.5 and A.6 describe techniques used for numerical simulations, including importance sampling to yield accurate estimates of average logical error rates with a reasonable number of syndrome samples.

A.1 Logical fidelity and correctable errors

The average logical channel $\overline{\mathcal{E}}_1$, defined in chapter 3, summarizes the effect of quantum error correction on a physical noise process $\mathcal{E}_0$ affecting an encoded state $\overline{\rho}$. In this section, we derive a closed form expression for the average logical channel in terms of the physical channel and the error correcting code parameters. Similar derivations have appeared in [CWB+17; GB15; RDM02], however, we present ours for the sake of completeness.

The action of the average logical channel on the logical state is

$$\begin{aligned}
\overline{\mathcal{E}}_1(\overline{\rho}) &= \sum_s \mathsf{Pr}(s)\mathcal{E}_1^s(\overline{\rho}), \\
&= \sum_s R_s \Pi_s \mathcal{E}_0(\overline{\rho}) \Pi_s R_s \\
&= \sum_s \sum_{i,j} \chi_{i,j} R_s \Pi_s P_i \overline{\rho} P_j \Pi_s R_s \\
&= \sum_s \sum_{\substack{i,j \\ s(P_i)=s(P_j):=s}} \chi_{i,j} R_s P_i \overline{\rho} P_j R_s \ ,
\end{aligned} \tag{A.1}$$

where in the last line we used the fact that $\Pi_s P_i = P_i \Pi_{s \oplus s(P_i)}$. In other words, whenever $s \neq s(P_i)$, the corresponding projector $\Pi_{s \oplus s(P_i)}$ annihilates the encoded state $\overline{\rho}$.

The chi matrix $\overline{\chi}$ of the effective logical channel defined by

$$\mathcal{E}_1(\overline{\rho}) = \sum_{l,m} \overline{\chi}_{lm} \overline{P}_l \overline{\rho} \overline{P}_m, \tag{A.2}$$

where $\overline{P}_l$ and $\overline{P}_m$ are logical operators of the code; can be extracted from eq. A.1.

The total probability of errors successfully corrected by the decoder: $\overline{\chi}_{00}$, can be estimated from the following observation. An error whose syndrome is s is corrected if the net effect of applying the error along with a recovery prescribed by the decoder results in an effective action of a stabilizer. In other words, all the terms in eq. A.1 where $R_s P_i$ and $P_j R_s$ are stabilizers contribute to $\overline{\chi}_{00}$. So,

$$\overline{\chi}_{0,0} = \sum_{\substack{E,E' \in \mathcal{E}_C \\ s(E)=s(E')\,,\,\overline{E}=\overline{E}'}} \phi(E)\ \phi^{\star}(E')\ \chi_{E,E'} \ , \tag{A.3}$$

where $\overline{E}$ is the logical component in the decomposition of E with respect to the Stabilizer group and $\phi(E)$ is specified by $R_{s(E)} E = \phi(E)\ S$ for any Pauli error E and some stabilizer S. The average logical infidelity $\overline{r}$ is then given by $1 - \overline{\chi}_{00}$.

When a Pauli error is not correctable, the effect of applying a recovery yields a logical operator. Hence, in general

$$\overline{\chi}_{l,m} = \sum_{\substack{E,E' \in \mathcal{E}_C \\ s(E)=s(E')\,,\,\overline{E}=\overline{E}'}} \phi(E,l)\ \phi^{\star}(E',m)\ \chi_{E\overline{P}_l, \overline{P}_m E'} \ . \tag{A.4}$$

where $R_{s(E)}\ |E\ \overline{P}_l| = \phi(E,l)\ S\ |\overline{P}_l|$, for $l \in \{0,1,2,3\}$, any Pauli error E and some stabilizer S. Here $|P|$ stands for the bare Pauli without any associated global phase.

A.2 Approximation quality for the uncorrectable error probability

In this section, we will quantity the accuracy of the approximating the uncorrectable error probability using $\widetilde{p}_u$ for concatenated codes. For simplicity, we will assume that the code-blocks in the concatenated code are all identical, and equal to a $[[n, 1, d]]$ quantum error correcting code, with $d \geq 3$. Recall that the distance of a level ℓ concatenated code scales as d^{ℓ}. We will use $t_{\ell} = \lfloor (d^{\ell} + 1)/2 \rfloor$ to denote the Hamming weight of the smallest uncorrectable error. Recall that $\widetilde{p}_u$ is defined recursively as the sum of two quantities: $\widetilde{Q}_1$ and $\widetilde{Q}_2$. We will use δ_{ℓ} to denote the inaccuracy in computing p_u for a level ℓ concatenated code:

$$\delta_{\ell} = |p_u(\mathcal{C}^{\star}_{\ell,1}) - \widetilde{p}_u(\mathcal{C}^{\star}_{\ell,1})| \;, \tag{A.5}$$

and γ_{ℓ} to denote the inaccuracy in computing Γ:

$$\gamma_{\ell} = |\widetilde{\Gamma}(\mathcal{C}^{\star}_{\ell}) - \Gamma(\mathcal{C}^{\star}_{\ell})| \;. \tag{A.6}$$

Then it follows that

$$\delta_{\ell} \leq n\delta_{\ell-1} + \gamma_{\ell} \;. \tag{A.7}$$

The most important ingredient in computing δ_{ℓ} is γ_{ℓ}, defined in eq. A.6. For simplicity we will compute γ_{ℓ} for the i.i.d depolarizing error model. However, for generic i.i.d Pauli error models, we can replace the depolarizing rate p in our analysis by the physical infidelity of the single qubit error model, r_0. The extension to correlated Pauli error models remains unclear.

An i.i.d application of the depolarizing channel on $n-$qubits can be described by

$$\mathcal{E}(\rho) = \sum_{P \in \mathcal{P}_n} \chi_{P,P} P \, \rho \, P \;,$$
$$\text{such that } \chi_{P,P} = (1-p)^{n-|P|} \left(\frac{p}{3}\right)^{|P|} \;, \tag{A.8}$$

where $\mathcal{P}_n$ is the $n-$qubit Pauli group, $0 \leq p \leq 1$ is the depolarizing rate and $|P|$ is the Hamming weight of the Pauli error P. In this case, we will show that

$$\gamma_{\ell} = \mathcal{O}(n^{\ell-1} p^{t_{\ell-1}+2}) \;, \tag{A.9}$$

for a level ℓ concatenated code.

Combining eq. A.9 with eq. A.7, we arrive at an expression for δ_ℓ:

$$\delta_\ell = \mathcal{O}(n^{\ell-1} p^{2+\lfloor (d+1)/2 \rfloor}) \;, \tag{A.10}$$

where d is the distance of a code block.

In the rest of this section, we will derive eq. A.9. Recall the following equation A.12 that outlines the approximation made by the heuristic to compute $\Gamma(\mathcal{C}^\star_\ell)$.

$$\widetilde{\Gamma}(\mathcal{C}^\star_\ell) = \sum_{E \,\in\, \mathcal{E}_\mathcal{C} \setminus \mathbb{I}} \; \sum_{s(\mathcal{C}_\ell)} \sum_{s(\mathcal{C}^\star_{\ell-1,1})} \cdots \sum_{s(\mathcal{C}^\star_{\ell-1,n})} \mathsf{Pr}(s(\mathcal{C}_\ell) | \hat{\mathcal{E}}_{\ell-1,1} \ldots \hat{\mathcal{E}}_{\ell-1,n}) \prod_{j=1}^{n} \mathsf{Pr}_\mathcal{D}(E_{\ell-1,j} \,|\, \hat{\mathcal{E}}_{\ell-1,j}) \mathsf{Pr}(s(\mathcal{C}^\star_{\ell-1,j})) \;, \tag{A.11}$$

$$= \sum_{E \,\in\, \mathcal{E}_\mathcal{C} \setminus \mathbb{I}} \; \prod_{j=1}^{n} \mathsf{Pr}_\mathcal{D}(E_{\ell-1,j} \,|\, \hat{\mathcal{E}}_{\ell-1,j})) \;. \tag{A.12}$$

It involves replacing the knowledge of conditional channels $\mathcal{E}^s_{\ell-1,j}$ by the average channel, $\hat{\mathcal{E}}_{\ell-1,j}$. We will prove the scaling in eq. A.9 two steps. First, is an observation that

$$\prod_{j=1}^{n} \mathsf{Pr}_\mathcal{D}(E_{\ell-1,j} | \hat{\mathcal{E}}_{\ell-1,j}) = \mathcal{O}(p^{t_{\ell-1}}) \;. \tag{A.13}$$

This follows from the fact that at least one of the errors $E_{\ell-1,j}$ in the error pattern $E_{\ell-1,1} \otimes \ldots \otimes E_{\ell-1,n}$ must be non-identity. Note that a non-identity logical error is left as a residual when the decoder for the subsequent lower level fails. Such an event will not occur for errors whose weight is below $t_{\ell-1}$.

Second, by showing that

$$\mathsf{Pr}(s(\mathcal{C}_{\ell,i}) | \mathcal{E}^{s(\mathcal{C}^\star_{\ell-1,1})}_{\ell-1,1} \ldots \mathcal{E}^{s(\mathcal{C}^\star_{\ell-1,n})}_{\ell-1,n}) \prod_{i=1}^{n} \mathsf{Pr}(s(\mathcal{C}^\star_{\ell-1,j})) = \mathsf{Pr}(s(\mathcal{C}_{\ell,i}) | \hat{\mathcal{E}}_{\ell-1,1} \ldots \hat{\mathcal{E}}_{\ell-1,n}) \prod_{i=1}^{n} \mathsf{Pr}(s(\mathcal{C}^\star_{\ell-1,j})) + \mathcal{O}(n^{\ell-1} p^2) \;. \tag{A.14}$$

Recall from the following equation that the average channel $\hat{\mathcal{E}}_{\ell,i}$ is defined recursively in terms of $\hat{\mathcal{E}}_{\ell-1,j}$.

$$\hat{\mathcal{E}}_{\ell-1,i} = \sum_{s(\mathcal{C}_{\ell-1,i})} \mathsf{Pr}(s(\mathcal{C}_{\ell-1,i})) \mathcal{E}^{s(\mathcal{C}_{\ell-1,i})}_{\ell-1,i} \left[\hat{\mathcal{E}}_{\ell-2,1} \otimes \ldots \otimes \hat{\mathcal{E}}_{\ell-2,n} \right] \;. \tag{A.15}$$

While the term corresponding to $s(\mathcal{C}_{\ell,i}) = 0$ describes the effect of stabilizers on the input state, the other terms include the effect of non-trivial errors. Note that the a non-trivial error E_ℓ has weight at least $t_{\ell-1}$, equal to the weight of the smallest uncorrectable error of the concatenated code $\mathcal{C}^\star_{\ell-1,j}$. Carrying this idea from level $\ell - 1$ to level 1, we find:

$$\hat{\mathcal{E}}_{\ell,i} = \mathcal{E}^{s(\mathcal{C}_{\ell,i})=0}_{\ell,i} \left[\hat{\mathcal{E}}_{\ell-1,1} \otimes \ldots \otimes \hat{\mathcal{E}}_{\ell-1,n}\right] + \mathcal{O}(p^{t_{\ell-1}}) \, , \tag{A.16}$$

$$= \left(\hat{\mathcal{E}}_{\ell-1,1} \otimes \ldots \otimes \hat{\mathcal{E}}_{\ell-1,n}\right) + \mathcal{O}(p^{t_{\ell-2}}) \, , \tag{A.17}$$

$$= \left(\hat{\mathcal{E}}_{1,1} \otimes \ldots \otimes \hat{\mathcal{E}}_{1,n^{\ell-1}}\right) + \mathcal{O}(p^{t_1}) \, , \tag{A.18}$$

where in eq. A.17 we have used the fact that the leading contribution to the conditional channel for the trivial syndrome, is the physical channel itself. Equation A.18 describes the recursion until level $\ell = 1$ where $\hat{\mathcal{E}}_{1,j} = \overline{\mathcal{E}}_{1,j}$.

Recall that the conditional channel for an error-syndrome $s(\mathcal{C}^\star_{\ell-1,i})$,

$$\mathcal{E}^{s(\mathcal{C}^\star_{\ell-1,i})}_{\ell-1,i} = \mathcal{E}^{s(\mathcal{C}_{\ell-1,i})s(\mathcal{C}_{\ell-2,1})\ldots s(\mathcal{C}_{\ell-2,n})\ldots s(\mathcal{C}_{1,1})\ldots s(\mathcal{C}_{1,n^{\ell-1}})}_{\ell-1,i} \, , \tag{A.19}$$

is defined by applying quantum error correction routines corresponding to the syndrome outcomes in the respective code-blocks of $\mathcal{C}^\star_{\ell-1,i}$. Note that an error is detected (by means of a non-trivial syndrome outcome) in a code block at level ℓ when the decoder operating on the code block at level $\ell - 1$ leaves a non-trivial residue. Hence, for a leading order analysis, we will consider conditional channels that correspond to trivial syndromes in all the code-blocks except for those at level one, i.e., $s(\mathcal{C}_{\ell,i}) = 0$ for all $\ell > 1$ in eq. A.19. In other words, we will consider errors that are corrected within the code blocks in level one:

$$\mathcal{E}^{s(\mathcal{C}_{\ell-1,i})=0,\, s(\mathcal{C}_{\ell-2,1})=0,\, \ldots,\, s(\mathcal{C}_{\ell-2,n})=0,\, \ldots,\, s(\mathcal{C}_{1,1})\ldots s(\mathcal{C}_{1,n^{\ell-1}})}_{\ell-1,i} = \mathcal{E}^{s(\mathcal{C}_{1,1})}_{1,1} \otimes \ldots \otimes \mathcal{E}^{s(\mathcal{C}_{1,n^{\ell-1}})}_{1,n^{\ell-1}} + \mathcal{O}(p^{t_1}) \, . \tag{A.20}$$

Using eqs. A.18 and A.20, we note that the quality of the approximation in eq. A.14

can be bounded as follows:

$$\left(\Pr(s(\mathcal{C}_\ell)|\mathcal{E}_1^{s(\mathcal{C}_{1,1})}\dots\mathcal{E}_1^{s(\mathcal{C}_{1,n^{\ell-1}})}) - \Pr(s(\mathcal{C}_\ell)|\hat{\mathcal{E}}_{1,1}\dots\hat{\mathcal{E}}_{1,n^{\ell-1}})\right)\prod_{j=1}^{n^{\ell-1}}\Pr(s(\mathcal{C}_{1,j}))$$

$$= \mathrm{tr}\left[\Pi_{s(\mathcal{C}_\ell)}\cdot\left((\mathcal{E}_1^{s(\mathcal{C}_{1,1})}\otimes\dots\otimes\mathcal{E}_1^{s(\mathcal{C}_{1,n^{\ell-1}})})(\rho) - (\hat{\mathcal{E}}_{1,1}\otimes\dots\otimes\hat{\mathcal{E}}_{1,n^{\ell-1}})(\rho)\right)\right]\prod_{j=1}^{n^{\ell-1}}\Pr(s(\mathcal{C}_{1,j}))\ , \tag{A.21}$$

$$= \sum_i\left[\left(\chi_{1,1}^{s(\mathcal{C}_{1,1})}\otimes\dots\otimes\chi_{1,n^{\ell-1}}^{s(\mathcal{C}_{1,n^{\ell-1}})}\right)_{i,i} - (\hat{\chi}_{1,1}\otimes\dots\otimes\hat{\chi}_{1,n^{\ell-1}})_{i,i}\right]$$
$$\mathrm{tr}\left[\Pi_{s(\mathcal{C}_\ell)}\cdot P_i\rho P_i\right]\prod_{j=1}^{n^{\ell-1}}\Pr(s(\mathcal{C}_{1,j}))\ , \tag{A.22}$$

$$\leq n^{\ell-1}\max_{s\,\in\,\mathbb{Z}_2^{n-k}}||\chi_1^s - \hat{\chi}_1||_\infty\Pr(s)\ , \tag{A.23}$$

where χ_1^s refers to the chi matrix of the conditional channel $\mathcal{E}_1^s$ while $\hat{\chi}_1$ refers to the chi matrix of the average channel $\hat{\mathcal{E}}_1$. In eq. A.23, we have used the matrix norm $||A||_\infty$ to refer to the maximum absolute value in the matrix.

To establish the scaling in eq. A.14 it remains to show that

$$\max_{s\,\in\,\mathbb{Z}_2^{n-k}}||\chi_1^s - \hat{\chi}_1||_\infty\Pr(s) = \mathcal{O}(p^2)\ . \tag{A.24}$$

Recall that the effective channel for a given syndrome s: $\mathcal{E}_1^s$, describes the composite effect of the physical noise process and quantum error correction conditioned on the measurement outcome s. Comparing eq. A.1 to the general form in eq. A.2, we find an expression similar to eq. A.4:

$$[\chi_1^s]_{i,i} = \frac{1}{\Pr(s)}\sum_{\substack{E\in\mathcal{E}_\mathcal{C}\\ s(E)=s}}\chi_{\overline{P}_iE,\overline{P}_iE}\ . \tag{A.25}$$

For the specific case of the depolarizing channel in A.8 we can express $[\chi_1^s]_{i,i}$, $\Pr(s)$

and $\hat{\chi}_{i,i}$ as polynomials in the depolarizing rate p:

$$[\chi_1^s]_{i,i} = \frac{1}{\mathsf{Pr}(s)} \sum_{w=1}^{n} A_{i,w}^s (1-p)^{n-w} \left(\frac{p}{3}\right)^w \,, \tag{A.26}$$

$$\mathsf{Pr}(s) = \sum_i \sum_{w=1}^{n} A_{i,w}^s (1-p)^{n-w} \left(\frac{p}{3}\right)^w \,, \tag{A.27}$$

$$\hat{\chi}_{i,i} = \sum_s \sum_{w=0}^{n} A_{i,w}^s (1-p)^{n-w} \left(\frac{p}{3}\right)^w \,, \tag{A.28}$$

where $A_{i,w}^s$ is the number of Pauli errors Q of Hamming weight w on which the action of the decoder leaves a residual logical error $\overline{P}_i$. In other words, $Q = \overline{P}_i R_s S$ where R_s is the recovery operation prescribed by the decoder for the error-syndrome s and S is any stabilizer. We can use two simple facts about errors to simplify the coefficients $A_{i,w}^s$. First, since the only error of Hamming weight zero is the identity which has $s = 0$, we find $A_{i,0}^s = \delta_{s,0}\delta_{i,0}$. Second, since all errors of Hamming weight up to $\lfloor (d-1)/2 \rfloor$ are correctable, we find $A_{i,w}^s = \delta_{i,0} A_{0,w}^s$ for all $w \leq \lfloor (d-1)/2 \rfloor$. Using these simplifications,

$$[\chi_1^s]_{i,i} = \frac{1}{\mathsf{Pr}(s)} \left[(1-p)^{n-1} \left(\frac{p}{3}\right) A_{0,1}^s + \mathcal{O}(n^2 p^2) \right] \,, \tag{A.29}$$

$$\mathsf{Pr}(s) = A_{0,1}^s (1-p)^{n-w} \left(\frac{p}{3}\right) + \mathcal{O}(n^2 p^2) \,, \tag{A.30}$$

$$\hat{\chi}_{i,i} = \delta_{i,0}(1-p)^n + 3n(1-p)^{n-1} \left(\frac{p}{3}\right) \delta_{i,0} + \mathcal{O}(n^2 p^2) \,. \tag{A.31}$$

It is now straightforward to see that eq. A.24 follows from the above set of equations.

In summary, this section establishes that the approximation used by the heuristic to compute $\widetilde{p}_u(\mathcal{C}^\star_{\ell,1})$, is accurate to $\mathcal{O}(n^{\ell+1} p^{2+\lfloor (d+1)/2 \rfloor})$ for the i.i.d depolarizing physical error model with error rate p. To get a sense for this approximation quality, we can plug in relevant numbers for an i.i.d Pauli error model and level-2 concatenated Steane code: $p = 10^{-3}, n = 7, \ell = 2, d = 3$. Numerical simulations of quantum error correction yield an estimate of the logical infidelity given by 4.2×10^{-9}. The analytical bound suggests that the logical estimator derived from the our heuristic method agrees with the logical infidelity up to $\mathcal{O}(10^{-11})$. However, the scaling suggests that the heuristic may not be not accurate for large codes in the high noise regime. Nonetheless we have strong numerical evidence to support that the logical estimator predicts the functional form of logical infidelity.

A.3 Predictability results for concatenated codes under coherent errors

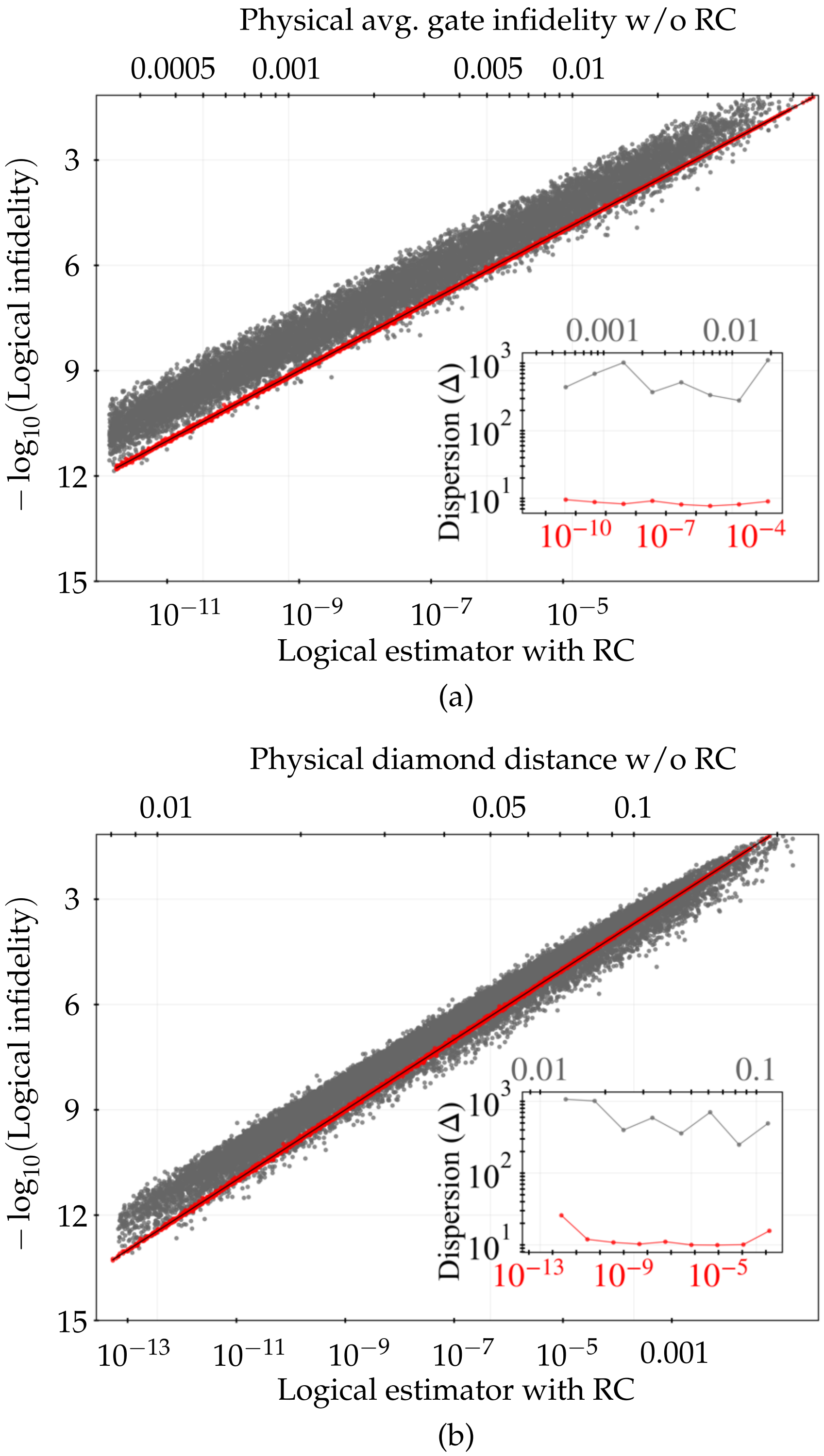
Numerical results presented in chapter 3 highlight the predictive power of the tools developed in this work with respect to the standard error-metrics, under random CPTP maps. Although CPTP maps encompass a wide range of physical noise processes, our method of generating random CPTP maps does not draw attention to an important class of noise processes – coherent errors – a special case of CPTP maps under which the evolution of a qubit is described by a unitary matrix. They occur due to imperfect control quantum devices and calibration errors [MB14; MAB20]. Various methods such as dynamical decoupling [YWL10; PSL13], designing pulses using optimal control theory [KRK+05] and machine learning approaches [NBS+19] are used to mitigate these errors. However, each of these methods have their shortcomings and unitary errors continue to form a major part of the total error budget [GD17; HDF19; BEK+18]. The methods presented in chapter 3 will be particularly advantageous in these cases.

In this section we highlight the predictive power of our tool, over standard error metrics, under different coherent noise processes. We choose a simple class of coherent errors modeled by an unknown unitary U_i on each physical qubit i, of the form $U = e^{-i\frac{\pi}{2}\delta\hat{n}\cdot\vec{\sigma}}$, where δ is the angle of rotation about an axis $\hat{n}$ on the Bloch sphere. With a slight loss of generality, we will consider $n-$qubit unitary errors of the form $\otimes_{i=1}^{n} U_i$. We control the noise strength by rotation angles δ_i drawn from a normal distribution of mean and variance equal to μ_δ where $10^{-3} \leq \mu_\delta \leq 10^{-1}$.

Figure A.1 shows that logical error rates vary over several orders of magnitudes across coherent errors with noise strength as measured by standard error-metrics such as infidelity and the diamond distance. In contrast, our tools provide an accurate prediction using the logical estimator. Moreover, we observe a drastic gain in in predictability using our tools for this case of unitary errors, when compared to CPTP maps in Fig. 3.2 of chapter 3.

A.4 Predicting the performance of surface codes

In this section we outline an extension of the techniques to predict the performance of concatenated codes using the logical estimator to surface codes. In summary, we make a crucial ansatz of a concatenated structure for surface codes. This assumption is motivated by a renormalization group based decoding algorithm developed in [DCP10]



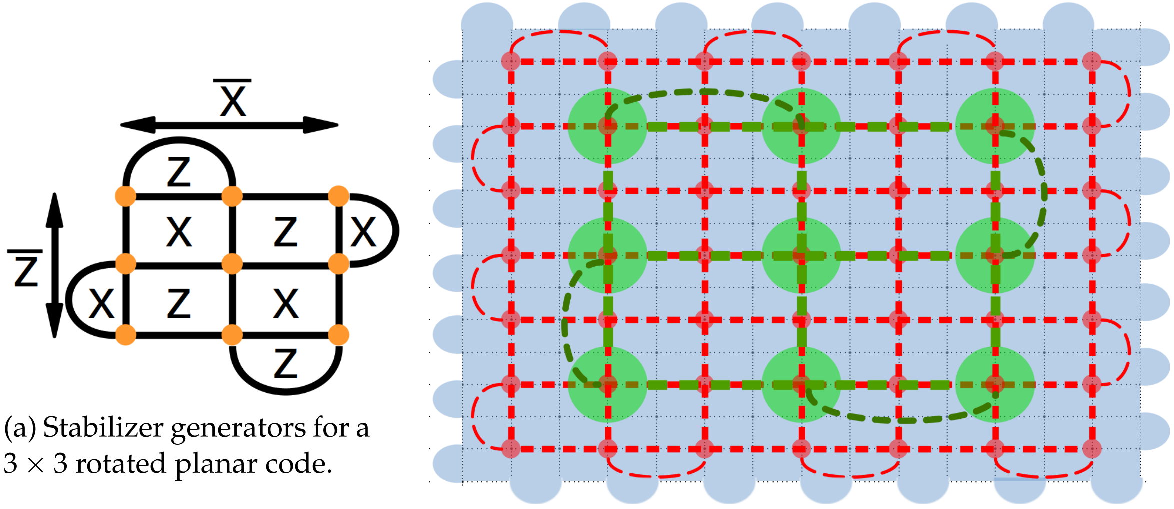
(a)

(b)

Figure A.1: Figures (a) and (b) compare the predictive powers of our tool (red) with the standard error metrics: infidelity and the diamond distance, respectively, under an ensemble of 16000 random unitary channels. These are similar to the ones in Fig. 3.2. The dispersion in the scatter corresponding to a metric (Δ in the insets) is indicative of its predictive power. The gains in predictability offered by our tool is drastic for the above case of unitary errors when compared to CPTP maps.

whose threshold is comparable to the optimal decoder. Hence, in order to define the logical estimator for surface codes, we must first specify a concatenated code structure for it.

For simplicity, we will illustrate the definition of the logical estimator using the square lattice rotated planar code in [Tuc20]. Let us consider the rotated planar code on a $3^\ell \times 3^\ell$ lattice for some integer $\ell > 0$, denoted by $\mathfrak{S}_{3^\ell \times 3^\ell}$. The $3^\ell \times 3^\ell$ square lattice has a self-similar structure in the bulk (ignoring boundaries) where a choice for the unit cell is the 3×3 lattice that specifies the smallest non-trivial code $\mathfrak{S}_{3\times 3}$ shown in Fig. A.2a. Based on this observation, we will construct a concatenated code shown in Fig. A.2b that will serve as a proxy for the surface code to compute the logical estimator. The surface code on the 3×3 unit cell, $\mathfrak{S}_{3\times 3}$ forms the smallest code-block of the concatenated code, and there are ℓ levels in total. Following the notation introduced in the background section, the resulting concatenated code is: $\mathfrak{S}_{3\times 3} \times \ldots \times \ldots \mathfrak{S}_{3\times 3}$.



(a) Stabilizer generators for a 3×3 rotated planar code.

(b) Rotated planar code as a concatenated code.

Figure A.2: Figure A.2a shows the stabilizer generators for the 3×3 rotated planar code denoted by $\mathfrak{S}_{3\times 3}$. The Fig. A.2b depicts an enforced concatenated structure on a rotated planar code. The blue, red and the green lattice depict levels $0, 1$ and 2 of the concatenated code corresponding to the underlying rotated planar code. For concatenated levels $\ell > 1$, the generators in Fig. A.2a are replaced by the corresponding logicals at level $\ell - 1$.

It is important to iterate that the concatenated code $\mathfrak{S}_{3\times 3} \times \ldots \times \ldots \mathfrak{S}_{3\times 3}$ and the surface code $\mathfrak{S}_{3^\ell \times 3^\ell}$ have fundamentally different encoding structures. Despite this difference, we use the former concatenated code to define the logical estimator for the latter surface code. Not to our surprise, we find that the logical estimator for the concatenated code $\mathfrak{S}_{3\times 3} \times \ldots \times \ldots \mathfrak{S}_{3\times 3}$ is significantly different from the average logical fidelity of the corresponding rotated planar surface code $\mathfrak{S}_{3^\ell \times 3^\ell}$. In contrast, we find that our heuristic for computing logical estimator for the surface codes plays a crucial role in selecting an optimal code. Recall that in the code selection section, we discussed how the logical estimator is crucial for selecting an optimal concatenated code for an underlying error model. In what follows, we have a similar illustration comparing logical estimators computed for two different surface codes along with their logical infidelities estimated through numerical simulations [Tuc20]. The underlying error model is identical to the error model in Fig. 3.5 – the twirl of a convex sum of rotations with a bias η between X and Z errors. We now consider two surface codes, one, $\mathfrak{S}_{9\times 12}$ – with the ability to correct more X than Z errors, and another, $\mathfrak{S}_{16\times 9}$ – which corrects more Z than X errors. The logical estimator verifies the expectation that the $\mathfrak{S}_{16\times 9}$ performs better as the bias for the Z errors increases relative X errors. Our results are summarized in Fig. A.3.

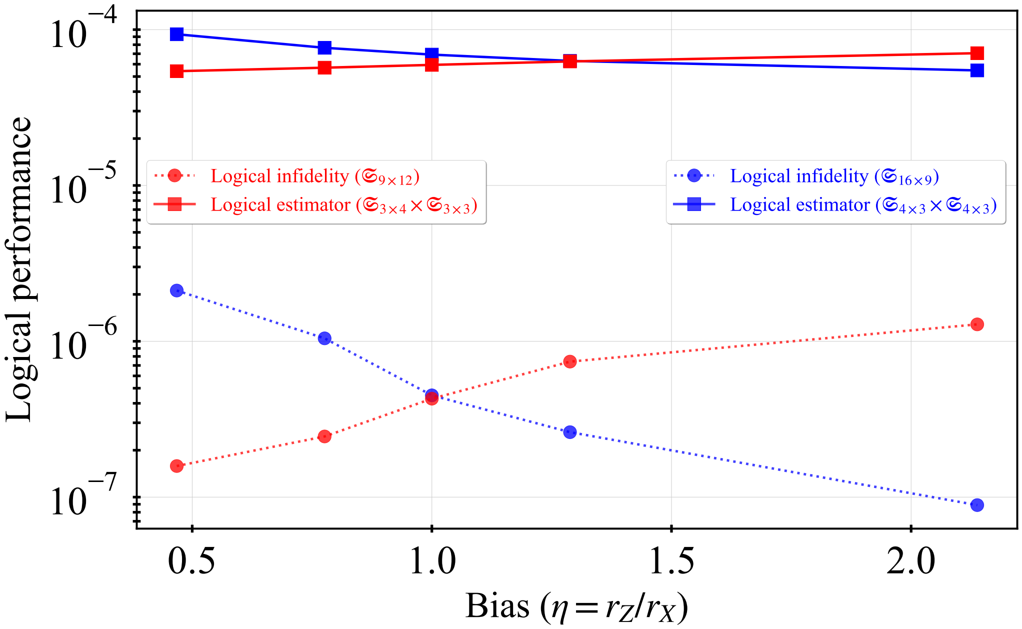


Figure A.3: Using the logical estimator to select optimal surface code. The above figure demonstrates the use of our tool in selecting an optimal surface under a biased Pauli error model. The choices of codes include rotated planar code of dimensions 9×12 and 16×9. While the solid lines depict the values of the logical estimator, the dashed lines correspond to the logical error rates estimated using numerical simulations. We observe that $\widetilde{p}_u$ helps select the optimal code for all noise rates.

It is important to note that these results are preliminary and a first step towards efficiently and accurately estimating the performance of surface codes. We believe that some of the ideas presented here using the logical estimator would guide the future research in this direction.

A.5 Numerical simulation details

The key steps involved in the simulation of an error correcting circuit include encoding, syndrome detection and application of recovery. In our simulations we assume each of these steps to be perfect and model the noise as an explicit step after encoding. Since we deal with coherent errors, we perform a full density matrix simulation. After application of the noise $\mathcal{E}$ to the encoded state $\overline{\rho}$, a syndrome s is sampled with probability $\mathrm{tr}(\Pi_s \mathcal{E}(\overline{\rho}))$ where Π_s is the syndrome projector. The state after syndrome detection is given by

$$\mathcal{E}(\overline{\rho}) \mapsto \overline{\rho}_s = \frac{\Pi_s \mathcal{E}(\overline{\rho}) \Pi_s}{\mathrm{tr}(\Pi_s \mathcal{E}(\overline{\rho}))}. \tag{A.32}$$

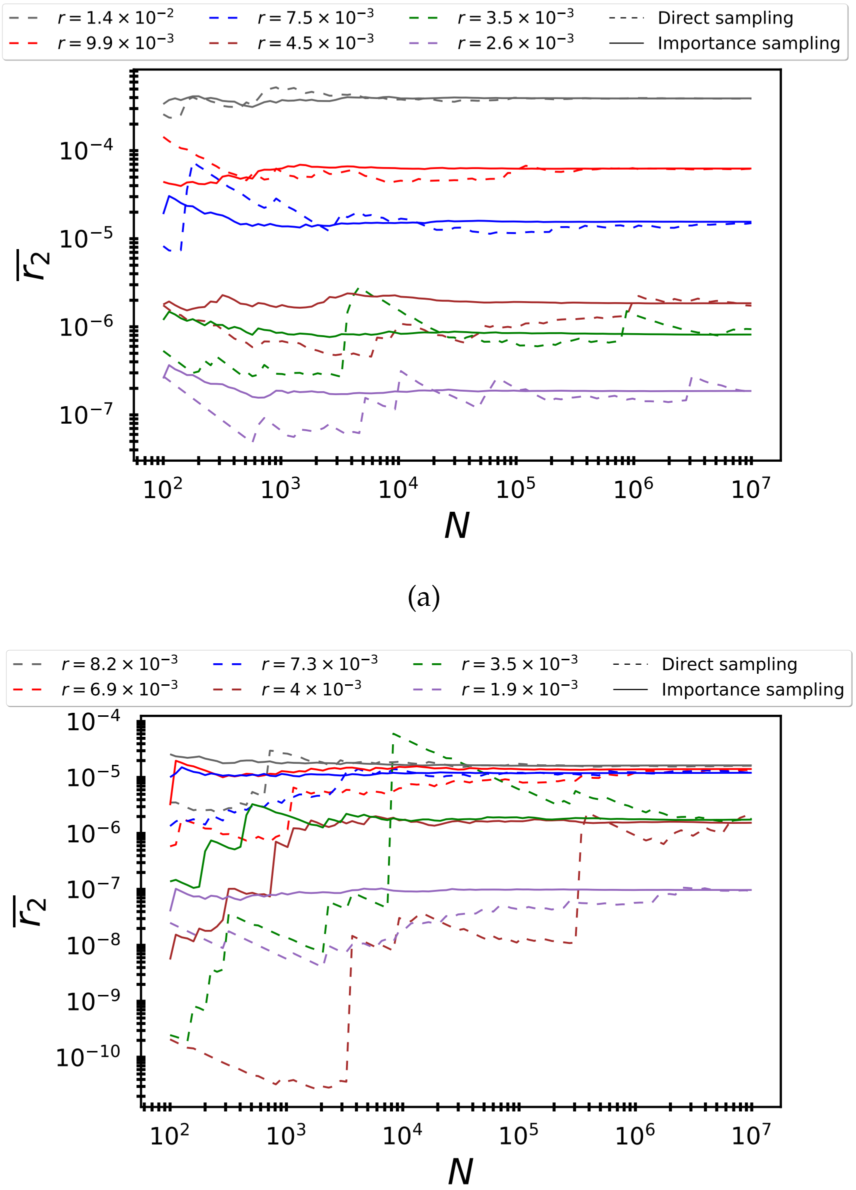
Followed by this, we apply a recovery based on minimum weight decoding and pass the resulting channel to the next level of the concatenated code. At the last level ℓ, we calculate the infidelity of the average logical channel. We report the mean of the infidelity over a large number of syndrome samples. More details about this procedure can be found in [IP18; Iye18]. Finally, we employ importance sampling for faster convergence detailed in section A.6.

So far, we discussed the simulation details for concatenated codes. Simulating surface codes require a slightly different machinery due to the difference in code structure. For deriving the logical error rates in the appendix section A.4, we used a software package called qecsim [Tuc20], which is also based on Monte Carlo simulation of error correcting circuits. The package also assumes perfect encoding, syndrome extraction and recovery application similar to the setting for concatenated codes. We used the minimum weight perfect matching (MWPM) decoder to obtain these results.

A.6 Importance sampling

A straightforward technique to estimate the logical error rate involves sampling syndrome outcomes according to the syndrome probability distribution for a quantum error correcting code and a physical noise process pair. However, there are serious drawbacks to this sampling method, due to the presence of rare syndromes – whose probability is typically less than the inverse number of syndrome samples. A detailed account of this can be found in [IP18] and in section 3.3 of [**I18**]. We briefly review the technique here for completeness.

In summary the average logical error rate is grossly underestimated unless an unreasonably large number of outcomes are sampled. We will resort to an importance



(a)

(b)

Figure A.4: The above figures highlight the rapid convergence rate of the importance sampler as compared to the direct sampler, under CPTP noise processes in Fig. A.4a and coherent errors in Fig. A.4b. Each trend line in the figures is associated to a physical noise rate. While different colors are used to identify different physical error rates, the solid and dashed lines are used to distinguish between the sampling techniques. Note that while the direct sampler takes a large number to syndrome samples to provide a reliable estimate of $r(\overline{\mathcal{E}}_\ell)$, the importance sampler achieves this task with far lesser syndrome samples. The speedup offered by importance sampling is quite drastic. The case for $r = 4 \times 10^{-3}$ in Fig. A.4b is a good example. The direct sampler shows signs of convergence around 10^7 syndrome samples, whereas the importance sampler converges with just 10^4 samples. Notice however that with only 10^4 samples, the direct sampler underestimates $r(\overline{\mathcal{E}}_\ell)$ by almost two orders of magnitude.

sampling technique proposed in [IP18], to improve our estimate of the average logical error rate. Previously, similar techniques have also been discussed for Pauli noise processes in [BV13; TLG+18]. Instead of choosing to sample the syndrome probability distribution, we sample an alternate distribution $Q(s)$, which we will simply refer to as the *importance distribution*. The corresponding sampling methods with $\mathsf{Pr}(s)$ and $Q(s)$ will be referred to as *direct sampling* and *importance sampling* respectively.

The expression for the average logical error rate estimated by the importance sampler takes a form:

$$r(\hat{\mathcal{E}}_\ell) = \sum_{\hat{s}} r(\mathcal{E}_\ell^{\hat{s}}) \frac{\mathsf{Pr}(s)}{Q(s)} \,, \tag{A.33}$$

where $\hat{s}$ is a random syndrome outcome drawn from the importance distribution $Q(s)$. The average estimated by importance sampling coincides with $r(\overline{\mathcal{E}}_\ell)$ which is estimated by the direct sampling technique. The crucial difference between the two sampling techniques is that the variance of the estimated average can be significantly lowered by an appropriate choice for the importance distribution $Q(s)$, which in our case, takes the form

$$Q(s) = \frac{P(s)^{1/k}}{Z} \,, \tag{A.34}$$

where Z is a normalization factor

$$Z = \sum_{s} P(s)^{1/k} \,, \tag{A.35}$$

and $k \in (0, 1]$ is chosen such that the total probability of non-trivial syndrome outcomes, $s \neq 00 \ldots 0$, is above a fixed threshold λ_0, i.e.,

$$\sum_{s \neq 00\ldots0} \frac{\mathsf{Pr}(s)^{1/k}}{Z} \geq \lambda_0 \,. \tag{A.36}$$

Figure A.4 shows that our heuristic for the importance distribution provides a rapid convergence to $r(\overline{\mathcal{E}}_\ell)$, when compared to the direct sampling method. Note that the noise processes in these figures are the same as those used to compare the predictive powers of physical error metrics in Fig. 3.2 of chapter 3 and Fig. A.1. Hence, the employment of importance sampling is key to an honest comparison of the predictive powers of the physical error metrics.

Appendix B

Impact of randomized compiling for general noise models

The purpose of the appendix is to aid and enhance the understanding of the content in chapter 4. It is organised as follows. Section B.1 presents a derivation of the logical error rate when each qubit of the Steane code unergoes a small rotation about $Z-$axis. In section B.2, we generalize these expressions to derive logical error rates for higher levels. Section B.3 presents numerical results for impact of randomized compiling on the logical performance under complex noise models.

B.1 Logical fidelity calculation for rotation about Z-axis

In this appendix section, we will derive the logical performance of Steane code under a unitary noise process described by a small over-rotation about the $Z-$axis, i.e. $\mathcal{E}(\rho) = R_Z(\omega)\rho R_Z(-\omega)$ where

$$R_Z(\omega) = \cos(\omega/2)\ I + i\sin(\omega/2)\ Z\ . \tag{B.1}$$

Recall that the Steane code is a $[[n,k]]$ quantum code with $n = 7, k = 1$, whose encoded states are fixed by the Stabilizer group $\mathcal{S}$ generated by $n-k$ generators:

$$\mathcal{S} = \langle ZZZZIII, ZZIIZZI, ZIZIZIZ, XXXXIII, XXIIXXI, XIXIXIX \rangle\ . \tag{B.2}$$

The effect of the unitary noise in Eq.(4.2) on each of the n qubits in the encoded state can be written as

$$\begin{aligned}\mathcal{E}^{\otimes n}(\bar{\rho}) &= R_Z^{\otimes n}(\omega)\ \bar{\rho}\ R_Z^{\otimes n}(-\omega) \\ &= \sum_{w\,\in\,\mathbb{Z}_2^{2n}} (-1)^{\sum_{j=n+1}^{2n} w_j} (\cos(\omega/2))^{2n-|w|} (i\sin(\omega/2))^{|w|} \left(\otimes_{j=1}^{n} Z^{w_j}\right) \bar{\rho} \left(\otimes_{j=n+1}^{2n} Z^{w_j}\right) .\end{aligned} \tag{B.3}$$

where $|w|$ is the Hamming weight of the binary sequence $w \in \mathbb{Z}_2^{2n}$.

To understand the effect of RC on performance, we need to estimate the total contribution to logical fidelity from terms in the noise process whose effect is rendered useless by RC. Since the noise model in Eq.(B.3) only applies $Z-$type errors, it suffices to consider the effect of correctable errors E and E' that are purely $Z-$type, besides the identity. In other words, $E, E' \in \langle Z_1, Z_2, \ldots, Z_n \rangle$. Table B.1a shows the contribution to the logical fidelity that is eliminated by RC. Each of the four rows in the table is associated with a $\chi-$matrix element of a particular form, labelled by γ_i for $1 \leq i \leq 4$.

Table B.1b provides all the ingredients necessary to compute the logical infidelity of the Steane code under the RC setting:

$$r(\overline{\mathcal{E}^T}_1) = 1 - (\kappa_1 + 7\kappa_2 + 7\kappa_3 + 7\kappa_4)\ , \tag{B.4}$$

$$= \frac{1}{512}(256 - 231\cos(\omega) - 49\cos(3\omega) + 21\cos(5\omega) + 3\cos(7\omega))\ . \tag{B.5}$$

Note that the coefficient appearing alongside each ϕ_i in Eq.(B.4) corresponds to its multiplicity, i.e., the number of combinations of errors E, E' that result in the same value of ϕ_i. In the absence of RC, the logical infidelity can be calculated using both tables B.1a and B.1b:

$$\begin{aligned} r(\overline{\mathcal{E}}_1) &= 1 - (\kappa_1 + 7\kappa_2 + 7\kappa_3 + 7\kappa_4 + 14\gamma_1 + 14\gamma_2 + 42\gamma_3 + 14\gamma_4)\ , \\ &= \frac{1}{64}(32 - 21\cos(\omega) - 14\cos(3\omega) + 3\cos(7\omega))\ . \end{aligned} \tag{B.6}$$

The above expressions describe the logical infidelities for level-1 concatenated Steane code in the RC and non-RC settings. The gain δ_1 can be calculated as the ratio of the above quantities. The appendix section B.2 discusses the recursion to compute the average logical channel for level$-\ell$ concatenated Steane code followed by the computation of the different metrics at level$-\ell$.

E	E'	Condition on E and E'	$\chi_{E,E'}$	
$I^{\otimes 7}$	S	$S \in \mathcal{S} \setminus \{\mathbb{I}\}$	$\cos^{10}(\omega/2)\sin^4(\omega/2)$	$= \gamma_1$
Z_i	$Z_i S$	$S \in \mathcal{S} \setminus \{\mathbb{I}\}\,, 1 \leq i \leq 7$	$\cos^8(\omega/2)\sin^4(\omega/2)$ $(3\sin^2(\omega/2) - 4\cos^2(\omega/2))$	$= \gamma_2$
S	S'	$S, S' \in \mathcal{S} \setminus \{\mathbb{I}\}\,, S \neq S'$	$\cos^6(\omega/2)\sin^8(\omega/2)$	$= \gamma_3$
$Z_i S$	$Z_i S'$	$S, S' \in \mathcal{S} \setminus \{\mathbb{I}\}\,, S \neq S'\,, 1 \leq i \leq 7$	$6\cos^8(\omega/2)\sin^6(\omega/2)$ $-12\cos^6(\omega/2)\sin^8(\omega/2)$ $+3\cos^4(\omega/2)\sin^{10}(\omega/2)$	$= \gamma_4$

(a)

E	E'	Condition on E and E'	$\chi_{E,E'}$	
$I^{\otimes 7}$	$I^{\otimes 7}$		$\cos^{14}(\omega/2)$	$= \kappa_1$
S	S	$S \in \mathcal{S} \setminus \{\mathbb{I}\}$	$\cos^6(\omega/2)\sin^8(\omega/2)$	$= \kappa_2$
Z_i	Z_i	$1 \leq i \leq 7$	$\cos^{12}(\omega/2)\sin^2(\omega/2)$	$= \kappa_3$
$Z_i S$	$Z_i S$	$S \in \mathcal{S} \setminus \{\mathbb{I}\}, 1 \leq i \leq 7$	$4\cos^8(\omega/2)\sin^6(\omega/2)$ $+3\cos^4(\omega/2)\sin^{10}(\omega/2)$	$= \kappa_4$

(b)

Table B.1: The above table describes the contribution to logical fidelity from different types of elements of the physical channel. While table B.1b describes the contribution to the logical infidelity from the diagonal (Pauli) terms, table B.1a specifies that from the off-diagonal terms in the physical channel. In each of the tables, the total contribution to logical infidelity is divided into four categories: (i) labelled $\gamma_1, \gamma_2, \gamma_3$ and γ_4 for the off-diagonal terms and (ii) $\kappa_1, \kappa_2, \kappa_3$ and κ_4 for the diagonal terms.

B.2 Logical channel for the concatenated Steane code

In this appendix section, we will describe the computation of the average logical channel for the level$-\ell$ concatenated Steane code under rotations about the $Z-$axis described in section 4.2. Ideally, we would like to take an exact average over conditional channels corresponding to all possible syndromes of the level$-\ell$ concatenated Steane code. However, the number of syndromes and hence the number of conditional channels grow exponentially with the number of physical qubits and the analysis becomes intractable beyond a few levels. Instead, in this section we compute an approximation wherein we recurse over the individual entries of the level-1 logical channel to arrive at the level$-\ell$ logical channel. We will achieve this in two broad steps:

1. Computation of level-1 logical channel.

2. Establish a recursion to compute level$-(\ell+1)$ from level$-\ell$ logical channel.

For a given noise process $\mathcal{E}$, we refer to its $\chi-$matrix as $\chi(\mathcal{E})$ and the corresponding logical $\chi-$matrix as $\chi(\overline{\mathcal{E}}_1)$. The following equation prescribes a way to calculate the entries of $\chi(\overline{\mathcal{E}})$ from $\chi(\mathcal{E})$ [IP18].

$$\chi(\overline{\mathcal{E}}_1)_{l,m} = \sum_{\substack{E,E' \in \mathcal{E}_C \\ s(E)=s(E')\,,\,\overline{E}=\overline{E}'}} \phi(E,l)\ \phi^{\star}(E',m)\ \chi_{E\overline{P}_l,\overline{P}_m E'}\ . \tag{B.7}$$

where $\mathcal{E}_C$ refers to the set of correctable errors, $\overline{P}_i$ denotes the logical version of Pauli P_i, and $R_{s(E)}\ |E\ \overline{P}_l| = \phi(E,l)\ S\ |\overline{P}_l|$, for $l \in \{0,1,2,3\}$, any Pauli error E and some stabilizer S. Here $|P|$ stands for the bare Pauli without any associated global phase. Note that, since the error model is a rotation about the $Z-$axis, we have $\mathcal{E}_C = \langle\{S_j Z_i : 1 \leq i \leq n\,,\, S_j \in \mathcal{S}_Z\}\rangle$. Here Z_i refers to a single qubit Z error on qubit i and $\mathcal{S}_Z = \langle ZZZZIII, ZZIIZZI, ZIZIZIZ\rangle$.

It is easy to see that the average logical channel for the level$-\ell$ concatenated Steane code $\chi(\overline{\mathcal{E}}_\ell)$ takes the form [HDF19]:

$$\chi(\overline{\mathcal{E}}_\ell) = \begin{pmatrix} [\chi(\overline{\mathcal{E}}_\ell)]_{0,0} & 0 & 0 & [\chi(\overline{\mathcal{E}}_\ell)]_{0,3} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ ([\chi(\overline{\mathcal{E}}_\ell)]_{0,3})^* & 0 & 0 & 1-[\chi(\overline{\mathcal{E}}_\ell)]_{0,0} \end{pmatrix}, \tag{B.8}$$

where $([\chi(\overline{\mathcal{E}}_\ell)]_{0,3})^*$ denotes the complex conjugate of $[\chi(\overline{\mathcal{E}}_\ell)]_{0,3}$.

First, we compute the entries for the level-1 matrix $\chi(\overline{\mathcal{E}}_1)$. Using table B.1a, we have

$$\begin{aligned}[\chi(\overline{\mathcal{E}}_1)]_{0,0} &= \kappa_1 + 7\sum_{i=1}^{3}\phi_i + 28\,\chi_3 + 14\sum_{j=1}^{4}\chi_j\ , \\ &= \frac{1}{64}(21\cos(\omega) + 14\cos(3\omega) - 3\cos(7\omega) + 32)\ . \end{aligned} \tag{B.9}$$

Table B.2 provides all the ingredients necessary to compute $[\chi(\overline{\mathcal{E}}_1)]_{0,3}$. Taking into account the multiplicities of terms of each kind, we have

$$[\chi(\overline{\mathcal{E}}_1)]_{0,3} = \zeta_1 + 42\,\zeta_4 + 7\sum_{i=2}^{8}\zeta_i\ , \tag{B.10}$$

$$= -\frac{1}{8} i \sin^3(\omega)(9\cos(2\omega) + 3\cos(4\omega) + 2)\ . \tag{B.11}$$

E	$\overline{Z}E'$	Condition on E and E'	$\chi_{E,\overline{Z}E'}$	
$I^{\otimes 7}$	$\overline{Z}$		$i\sin^7(\omega/2)\cos^7(\omega/2)$	$=\zeta_1$
$I^{\otimes 7}$	$\overline{Z}S$	$S\in\mathcal{S}_Z\setminus\{\mathbb{I}\}$	$i\sin^3(\omega/2)\cos^{11}(\omega/2)$	$=\zeta_2$
S	$\overline{Z}$	$S\in\mathcal{S}_Z\setminus\{\mathbb{I}\}$	$i\sin^{11}(\omega/2)\cos^3(\omega/2)$	$=\zeta_3$
S	$\overline{Z}S'$	$S,S'\in\mathcal{S}_Z\setminus\{\mathbb{I}\}\,,S\neq S'$	$i\sin^7(\omega/2)\cos^7(\omega/2)$	$=\zeta_4$
Z_i	$\overline{Z}Z_i$	$1\leq i\leq 7$	$-i\sin^7(\omega/2)\cos^7(\omega/2)$	$=\zeta_5$
Z_i	$\overline{Z}Z_iS$	$S\in\mathcal{S}_Z\setminus\{\mathbb{I}\},1\leq i\leq 7$	$4i\sin^5(\omega/2)\cos^9(\omega/2)$ $-3i\sin^3(\omega/2)\cos^{11}(\omega/2)$	$=\zeta_6$
Z_iS	$\overline{Z}Z_i$	$S\in\mathcal{S}_Z\setminus\{\mathbb{I}\},1\leq i\leq 7$	$4i\sin^9(\omega/2)\cos^5(\omega/2)$ $-3i\sin^{11}(\omega/2)\cos^3(\omega/2)$	$=\zeta_7$
Z_iS	$\overline{Z}Z_iS'$	$S,S'\in\mathcal{S}_Z\setminus\{\mathbb{I}\}\,,S\neq S',1\leq i\leq 7$	$12i\sin^5(\omega/2)\cos^9(\omega/2)$ $-25i\sin^7(\omega/2)\cos^7(\omega/2)$ $+12i\sin^9(\omega/2)\cos^5(\omega/2)$	$=\zeta_8$

Table B.2: The above table describes the contribution to $\overline{\chi}_{0,3}(\mathcal{E})$ from different types of elements of the physical channel. Note that none of these contributions come from the diagonal part of $\chi(\mathcal{E})$.

In the second step, we establish a recursion to compute the individual entries of $\chi(\overline{\mathcal{E}}_\ell)$ from the entries of $\overline{\chi}_{\ell-1}(\mathcal{E})$ under hard-decoding algorithm. After massaging the expressions in equations B.9 and B.11, we observe that

$$[\chi(\overline{\mathcal{E}}_{\ell+1})]_{0,0} = f_{0,0}([\chi(\overline{\mathcal{E}}_\ell)]_{0,0}), \text{and} \tag{B.12}$$
$$[\chi(\overline{\mathcal{E}}_{\ell+1})]_{0,3} = f_{0,3}([\chi(\overline{\mathcal{E}}_\ell)]_{0,3}), \tag{B.13}$$

where

$$\begin{aligned} f_{0,0}(z) &= z^2(63-434z+1260z^2-1848z^3+1344z^4-384z^5), \text{and} \\ f_{0,3}(z) &= -2z^3(7+84z^2+192z^4). \end{aligned} \tag{B.14}$$

Combining the above two steps, we compute all the entries of $[\chi(\overline{\mathcal{E}}_{\ell+1})]$.

For small rotation angle ω, we observe from equations B.9 and B.14 that up to leading order

$$\begin{aligned} [\chi(\mathcal{E}_\ell)]_{0,0} &\approx 1-63^{2^\ell-1}\,(\omega/2)^{2^{\ell+1}}, \text{and} \\ [\chi(\mathcal{E}_\ell)]_{0,3} &\approx -i14^{\frac{3^\ell-1}{2}}\,(\omega/2)^{3^\ell}. \end{aligned} \tag{B.15}$$

Note that with increase in number of levels ℓ, for small angle ω, $[\chi(\overline{\mathcal{E}}_\ell)]_{0,0} \to 1$ and $[\chi(\overline{\mathcal{E}}_\ell)]_{0,3} \to 0$. This is expected because for small angles, the channel is close to the identity channel and the error correction procedures is able to correct all the errors. Also, note that the off diagonal entry approaches 0 faster than the diagonal entry approaches 1. This is a consequence of the process of error correction decohering the physical channel [BWG+18].

Now, we compute the logical $\chi-$matrix corresponding to the noise process under RC i.e. $\chi(\overline{\mathcal{E}^T}_\ell)$. The matrix in this case takes the form:

$$\chi(\overline{\mathcal{E}^T}_\ell) = \begin{pmatrix} [\chi(\overline{\mathcal{E}^T}_\ell)]_{0,0} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 - [\chi(\overline{\mathcal{E}^T}_\ell)]_{0,0} \end{pmatrix} . \tag{B.16}$$

Similar to the nonRC case, we first compute the entries for the level-1 matrix $\chi(\overline{\mathcal{E}^T}_1)$. Using the ingredients from table B.1b, we have

$$[\chi(\overline{\mathcal{E}^T}_1)]_{0,0} = \frac{1}{512}(256 + 231\cos(\omega) + 49\cos(3\omega) - 21\cos(5\omega) - 3\cos(7\omega)) .$$

The recursive relation to calculate the above quantity for higher levels is given by:

$$[\chi(\overline{\mathcal{E}^T}_{\ell+1})]_{0,0} = g_{0,0}([\chi(\overline{\mathcal{E}^T}_\ell)]_{0,0}) ,$$

where

$$g_{0,0}(z) = z^2(21 - 98z + 210z^2 - 252z^3 + 168z^4 - 48z^5) .$$

For small rotation angle ω, up to leading order

$$[\chi(\overline{\mathcal{E}^T}_\ell)]_{0,0} \approx 1 - 21^{2^\ell - 1}\,(\omega/2)^{2^{\ell+1}} .$$

The above expression indicates that $[\chi(\overline{\mathcal{E}^T}_\ell)]_{0,0} \to 1$ with increase in number of concatenation levels ℓ provided the angle of rotation is below the threshold.

Having arrived at an expression for the average logical channel for a level$-\ell$ concatenated code, we can now define the logical error rate using the infidelity and diamond distance metrics. The logical infidelity takes the simple closed form:

$$\overline{r}_\ell = 1 - [\chi(\overline{\mathcal{E}}_\ell)]_{0,0} .$$

B.3 Numerical results for complex noise models

In this appendix section, we will present numerical studies of the performance of concatenated Steane codes under two distinct models of general Markovian noise. The results are presented as scatter plots formatted as follows. Each point is associated to the performance of a physical noise process. While the $X-$ coordinate is used to denote the physical error rate, its $Y-$coordinate denotes the ratio between the performance in the non-RC setting and the RC setting, measured by δ_ℓ in Eq.(4.1). Note that RC can either improve or degrade the code's performance. We have used a dashed line at $\delta_\ell = 1$ to identify the break-even region where RC has no impact on the performance. Points that lie below the dashed line, coloured in red, identify physical channels where a degradation in performance is observed. On the other hand, points in green that lie above the dashed line identify physical channels where RC provides a performance gain. The points in grey, that lie close to the dashed line should be ignored since they correspond to cases where the relative difference between the logical error rates for the non-RC and RC cases is negligible: less than 10%.

The first complex error model is a unitary model where each qubit experiences a different random rotation about an arbitrary non-Pauli axis $\hat{n}$, specified by U of the form

$$U = e^{-i\frac{\pi}{2}\delta\hat{n}\cdot\vec{\sigma}}, \tag{B.17}$$

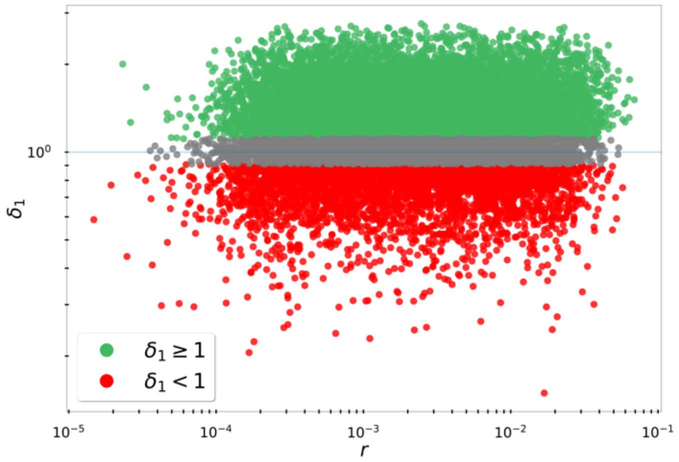
where δ is the angle of rotation. Hence, the $n-$qubit unitary errors in our model are of the form $\otimes_{i=1}^{n} U_i$, where U_i in prescribed by Eq.(B.17). We control the noise strength by setting the rotation angles δ_i drawn from the normal distribution: $\mathcal{N}(\mu_\delta, \mu_\delta)$, where $10^{-3} \leq \mu_\delta \leq 10^{-1}$. Fig. B.1 shows the performance gain metric under this error model. It demonstrates that there exist some instances where RC provides a performance gain of 10x, as well as others where RC causes a performance degradation of 10x.

The second error model is described by the i.i.d action of a random single qubit CPTP map, on each of the physical qubits of the code. The random CPTP map on a single qubit is derived from unitary dynamics U on a Hilbert space of three qubits [IP18]. The unitary matrix U is generated form a random Hermitian matrix H using $U = e^{-iHt}$, where $0 \leq t \leq 1$ provides a handle on the strength of noise described by the resulting CPTP map. We vary the noise strength by controlling t in the range $[0.001, 0.1]$. Figure B.2 shows RC's impact on the performance of concatenated Steane codes under physical CPTP maps. The absence of a clear trend showing a performance gain or degradation is evident for level-2 in Fig. B.2b. Even across physical CPTP maps with similar fidelity, while for one instance, RC induces a performance gain of up to three orders of magnitude, for another, it inflicts a loss in performance of similar magnitude.

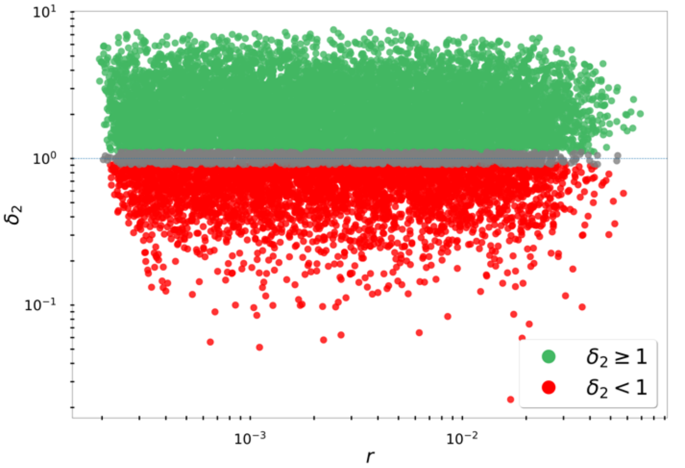
The case of level-1 performance under physical CPTP maps in Fig. B.2a is rather different from the level-2 case in Fig. B.2b. Over the large ensemble of 18000 physical CPTP maps, we observe that RC always leads to performance gains for the level-1 Steane code. These performance gains can be explained as follows. First of all, a CPTP map can be well approximated by its leading Kraus operator K, which is derived from the largest eigenvector of its Choi matrix [CDAE19]. Furthermore, in an i.i.d physical error model, K can be expressed as a tensor product. In terms of K, the leading contributions to infidelity come from chi-matrix entries $\chi_{i,j}$ expressed as:

$$\chi_{i,j} = \mathrm{tr}(KP_i)\mathrm{tr}(K^\dagger P_j) \ , \tag{B.18}$$

where P_i is a single qubit of one type (X, Y or Z), and $P_j = P_i S$ for some stabilizer S, is a three-qubit error of the same type as P_i. In the low noise regime, the off-diagonal entries of K are small, especially for incoherent CPTP maps, where K is close to a Positive semi-definite matrix [CDAE19]. Using the fact that the trace inner product between K and the Pauli matrix Z is a real number d given by $d = K_{1,1} - K_{2,2}$, we can conclude that $\chi_{i,j}$ in Eq.(B.18) for $Z-$type errors P_i and P_j of weights 1 and 3 respectively, is always positive. In other words, the $\chi_{i,j} \sim d^4$ for some $d \ll 1$. Removing such terms should degrade the performance of the code. On the contrary, removal of $\chi_{i,j}$ for uncorrectable errors P_i, P_j leads to performance gains. The largest of these chi-matrix entries can be identified with two $Z-$type Pauli errors P_i, P_j, each having weight two. This property can be associated with the fact that the Steane code is degenerate: there exists a logical operator whose weight is smaller than that of a stabilizer. Repeating a similar analysis as before, we find that the corresponding $\chi_{i,j}$ for these uncorrectable errors, also scale as d^4 for some $d \ll 1$. Their removal leads to performance gains. Note that there are more uncorrectable errors than correctable ones and the corresponding chi-matrix elements have comparable magnitudes. Hence, we find that RC is more likely to induce performance gains. Note that higher concatenation levels of the Steane code do not correspond to degenerate codes. Hence, we cannot guarantee a performance gain or degradation in those cases, as shown in Fig. B.2b.

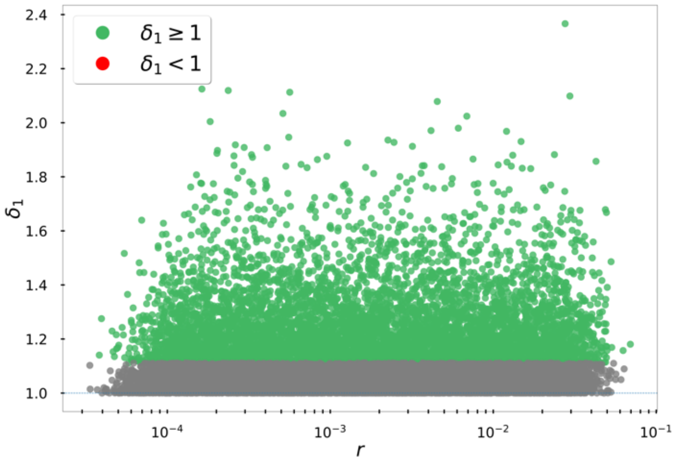


(a)

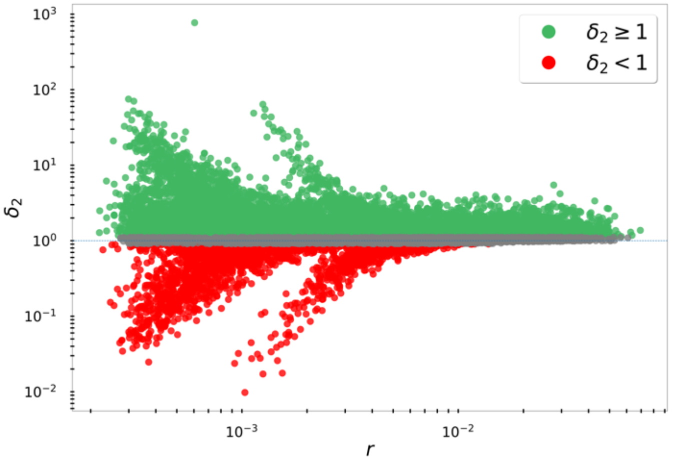


(b)

Figure B.1: The above figures highlight the strong dependence of the impact of RC on the details of the physical noise process, for concatenated Steane codes. The ensemble of noise processes considered here comprises of 16000 samples of unitary rotations about a fixed random axis. Red and green points are used to identify physical noise processes that lead to a performance gain and a performance loss, respectively, in the presence of RC. The magnitude of performance gains and losses are measured by the ratio of logical error rates in the non-RC and RC settings, i.e., δ_1 for level-1 concatenated Steane code in Fig. B.1a, and δ_2 for level-2 in Fig. B.1b.



(a)



(b)

Figure B.2: The above figures highlight the strong dependence of the impact of RC on the details of the physical noise process, for concatenated Steane codes. The ensemble of noise processes considered here comprises of 18000 random CPTP maps. Red and green points are used to identify physical noise processes that lead to a performance gain and a performance loss, respectively, in the presence of RC. The magnitude of performance gains and losses are measured by the ratio of logical error rates in the non-RC and RC settings, i.e., δ_1 for level-1 concatenated Steane code in Fig. B.2a, and δ_2 for level-2 in Fig. B.2b.