

Fault-tolerant Preparation of Distant Logical Bell Pair

with application in the *Magic Square Game*

by

Andy Zeyi Liu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization (Quantum Information)

Waterloo, Ontario, Canada, 2024

© Andy Zeyi Liu 2024

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.

Abstract

Quantum entanglement facilitates nonlocal correlations that defy classical physics, forming the basis for quantum technologies, including quantum computation and communication. Nonlocal games exemplify this power, where entangled players achieve outcomes unattainable by classical means. This thesis focuses on optimizing the preparation of high-fidelity logical Bell pairs in the context of the *fault-tolerant magic square game*, seeking to minimize Bell pair(ebit) consumption while maintaining a low logical error rate.

In this thesis we introduce a novel approach leveraging an interface circuit and entanglement purification protocol (EPP) to translate states between physical and logical qubits and purify noisy logical ebits. This method significantly reduces the number of initial ebits needed compared to conventional strategies. Our analytical and numerical analyses, particularly for the $[[7^k, 1, 3^k]]$ concatenated Steane code, demonstrate substantial (actually, exponential) ebit savings and higher noise threshold. Analytical lower bounds for local noise threshold of 4.70×10^{-4} and initial ebit infidelity threshold of 18.3% are obtained. Additionally, we construct an analogous interface for the surface code through lattice surgery, offering further improvements in fault tolerance and compatibility with current quantum hardware.

Our framework is adaptable to various quantum error-correcting codes (QECCs) and experimental platforms. We hope our protocol will not only enhance understanding of fault-tolerant nonlocal games, but also spark further exploration of interfacing between different QECCs, promoting the development of modular quantum architectures and advancing quantum internet.

Statement of Contributions

- Definitions and results in Chapter 2 are mostly referenced from Debbie Leung’s lecture series [60] and Daniel Gottesman’s textbook [36]. This chapter also includes my own examples and motivations for clarity.
- Chapter 3 is largely based on paper by Aliferis, Gottesman and Preskill [3] with definitions and constructions modified for our context.
- Chapter 4 is based on joint work with Debbie Leung and the results are in preparation for publication.
- Chapter 5 is based on my own exploration, which attempts to extend results in Chapter 4 to the surface code. Paula Belzig pointed me to the paper by Paweł Mazurek et al [66], which motivates constructions in this chapter.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Debbie Leung, whose inspirational approach to research has profoundly shaped my academic journey over the past two years. Her enthusiasm, curiosity, scientific rigor, and persistence in addressing fundamental questions have been truly motivating. I am grateful for the freedom she allowed me to explore my interests within our project and overall research, as well as her support whenever I felt lost. Her guidance, including valuable suggestions on courses, has significantly enhanced my research capabilities. I ended up taking twice as many courses, but it was a worthwhile experience. Moreover, her care and support during my difficult times in the first year were immensely helpful. It has been an absolute joy to be supervised by her, and I look forward to more future collaborations.

I also like to thank Nilanjana Datta, Sergii Strelchuk, Marco Tomamichel and Changyu Kim for introducing QIQC to me. The interdisciplinary nature of this subject has profoundly fulfilled my curiosity about the world. It was a pleasure to work with them on various topics in the field.

Furthermore, I am very grateful to Michele Mosca and William Slofstra for being the readers of this thesis. Thank you for taking the time to read my thesis and sharing insightful feedback.

For the content in this thesis, I would like to thank Rui Chao for his open-source stabilizer circuit simulator, the Quantum StackExchange community for their assistance with quantum error correction, and Daniel Gottesman for his excellent textbook.

I would like to extend my deepest gratitude to my friends. Special thanks to Kohdai Kuroiwa for being a role model and engaging in thought-provoking discussions; Paula Belzig for insightful follow-ups and key references on the surface code; Xiao Yang for sparking my interest in graph theory and optimization, which fosters my spontaneous learning of algebraic graph theory that later enhanced my understanding of the latest qLDPC codes. Thanks to Dongze Li for being a gym partner and sharing life's ups and downs; Yuming Zhao for sharing course notes and enriching math discussions. Amit and Sanchit, thank you for being midnight and weekend grind companions. Amolak, your humor and companionship as an office mate kept me from burning out. I would to thank Anfu Wang and friends back in Cambridge; Roberto Rubboli in Singapore for continually examining my mathematical rigour while we worked together; Zhaoyi Li, for occasional but inspiring discussions on quantum computing. Lastly, thank you to all other friends whose names are not mentioned but whose support has been equally significant.

Finally, my heartfelt gratitude goes to my parents and grandparents for their unconditional love and sacrifices throughout my growing up. Their encouragement and support have been a constant source of strength and motivation. I dedicate my achievements to them with deepest appreciation.

Last but not least I would like to thank my girlfriend Linjun Wu for her unwavering support over long distance, her understanding during times of tremendous pressure, and our engaging discussions about research. Beyond academia, she brings immense joy and balance to my life. Together, we have grown and supported each other, becoming better individuals. We look forward to the future with optimism and shared aspirations.

Table of Contents

Author's Declaration	ii
Abstract	iii
Statement of Contributions	iv
Acknowledgements	v
List of Figures	xi
List of Tables	xv
1 Introduction	1
1.1 Motivation	1
1.2 Main results and thesis outline	2
2 Quantum Error Correction	6
2.1 Notations	6
2.2 QECC Condition	7
2.3 Stabilizer Formalism	8
2.4 Quantum Codes from Classical Codes	10
2.4.1 Steane Code	14

2.4.2	Surface Code	14
2.5	Evolution of Stabilizers	17
2.6	Entanglement Purification Protocol	22
3	Fault-tolerance(FT)	24
3.1	Error Model	25
3.2	Fault-tolerant Gadgets	26
3.2.1	FT Criteria	27
3.2.2	FT Logical Gates	28
3.2.3	FT Measurement and Preparation	29
3.2.4	FT EC	32
3.3	Threshold Theorem	34
3.3.1	ExRec Construction	34
3.3.2	The Theorem	37
3.3.3	FT for Steane Code	38
3.3.4	FT for Surface Code	39
4	FT Magic Square Game with Concatenated Steane Code	42
4.1	Magic Square Game	44
4.2	Preparation of High-fidelity Physical EPRs	44
4.3	ExRec and Threshold	46
4.4	Direct Encoding	52
4.4.1	Schemes	53
4.4.2	Logical error rate of Scheme A	54
4.5	Interface + EPP	56
4.5.1	Interface	56
4.5.2	Schemes	57

4.5.3	Logical error rate of Interface	60
4.5.4	Logical error rate of Interface+EPP	64
4.5.5	Scheme A	70
4.5.6	Scheme B	74
4.6	Fault-tolerant Magic Square Game	76
4.6.1	Failure probability	76
4.6.2	Nonlocal resources	78
4.6.3	Space-time overheads	82
4.6.4	Numerical results	83
5	Extension to the Surface Code	87
5.1	Direct Encoding	87
5.2	Interface	89
5.2.1	Part 1: $ \psi\rangle \rightarrow \psi\rangle_{L'}$	89
5.2.2	Part 2 $ \psi\rangle_{L'} \rightarrow \psi\rangle_L$	90
5.3	Discussion	92
6	Conclusion	93
	References	95
	APPENDICES	103
A	Bounds on $\mathbb{P}(\text{exRec is bad})$	104
A.1	Bound on third-order terms	104
A.2	Conversion to second-order bounds	108
A.3	System of equations for the error bounds	108
A.4	Tightness of bounds	109

B	Direct Encoding Scheme B	111
B.1	Validity of Scheme B	111
C	Interface+EPP error analysis	113
C.1	Interface error analysis	113
C.2	Proofs for bounds on EPP failure probabilities	115
D	Miscellaneous	118
D.1	Distribution of logical error after CNOT-exRec	118
D.2	Bounds for Shor's parity measurement	118
D.3	Malignant pair matrix(MPM)	120

List of Figures

1.1	The procedure of the <i>Fault-tolerant Magic Square Game</i> . For the <i>Magic Square Game</i> , Alice and Bob are randomly assigned a row and a column index $a, b \in \{0, 1, 2\}$. They then each reply with three answers $[A_0^a, A_1^a, A_2^a]$ and $[B_0^b, B_1^b, B_2^b]$ with $A_i^a, B_j^b \in \{\pm 1\} \forall i, j$. The winning condition is that $\prod_{i=0}^2 A_i^a = +1$ and $\prod_{i=0}^2 B_i^b = -1$ and $A_b^a = B_a^b$. No communication is allowed during the game. However, if Alice and Bob share two EPR pairs prior to the game and perform corresponding measurements, they are able to win the game with probability 1 (in the error-free scenario). In the FT case, Alice and Bob are supplied with noisy physical ebits. In order to implement the same strategy, they use these to create 2 logical ebits and then perform logical measurements.	3
1.2	The number of raw ebits required to achieve magic square game failure probability Δ versus Δ . Due to the complexity of simulation when the concatenation level is high, analytical bounds are obtained instead. The green and blue bands above represent regions bounded by the upper and lower bounds for the two methods respectively.	4
2.1	The full procedure for a QECC cycle	7
2.2	Non-fault-tolerant syndrome measurement circuit for the operator $XZZXI$	10
2.3	Space packed by a code $\mathcal{C}$ with distance d . Each circle is a Hamming ball with center being a codeword $c \in \mathcal{C}$ and radius $\lfloor \frac{d-1}{2} \rfloor$ to ensure circles are disjoint so different codewords are distinguishable.	12

2.4	An example of a surface code and its stabilizers. This surface code patch consists of 61 qubits, represented by blue dots in the figure. The intersections of black lines are ‘vertices’ and a ‘plaquette’ is either a square face in the middle or the region bounded by 3 qubits on two sides. The green operator crossing a vertex is the star operator X_v and the orange operator bounding a plaquette is the plaquette operator Z_p . The top and bottom boundaries are called ”smooth” while the left and right boundaries are called ”rough boundaries”.	15
2.5	Possible types of operators on a surface code. (a) Homologically trivial cycles, i.e. boundary of plaquettes in the original and dual lattices. The dual lattice is coloured light blue. The orange closed path is an example of Z -type cycle and can be tiled by Z_p -stabilizers. The green closed path is an example of X -type cycle and can be tiled by X_v -stabilizers. (b) Open path. These are detectable errors. The green path is an example of Z -type errors. (c) Homologically non-trivial cycles. The purple path represents an example of $\bar{Z}$ and the red represents an example of $\bar{X}$.	16
2.6	$ \bar{0}\rangle$ encoding circuit for the Steane code.	20
2.7	A simple EPP. The upper half and lower half are owned by Alice and Bob respectively. Upon measuring the second qubit in the Z -basis, if the outcomes agree it marks a purification.	23
3.1	Single cat-state measurement of $XZZXI$ for $[[5, 1, 3]]$ code	29
3.2	Fault-tolerant $ \bar{0}\rangle$ encoding circuit for the $[[7, 1, 3]]$ code.	31
3.3	(a) Steane EC. U and V are corrections corresponding to measured syndromes. (b) Knill EC using logical teleportation. The applied gate Q depends again on the syndrome measurements. (c) Shor EC, $S_1, \dots, S_{n-k}$ denote the stabilizers, and ‘cat’ is the cat state. The whole set of measurements is to be repeated many times until we get $t + 1$ the same syndrome measurements. Note that in (a)(b)(c) wires with a bar denote blocks of qubits, X, Z -measurements, and CNOT are all transversal (d) An example of Flag EC. This is to measure the syndrome of $XZZXI$ for the $[[5, 1, 3]]$ code. A flag qubit is introduced to detect errors of higher weight caused by the 2nd/3rd/4th gate.	32

3.4	Examples of different exRec. Upper-left is the $ \bar{0}\rangle$ -exRec. Upper-right is the $\bar{X}$ -measurement-exRec. Lower-left is a $\bar{H}$ -exRec. Other one-qubit gates are constructed analogously. Lower-right is an example of a two-qubit exRec, the $\overline{\text{CNOT}}$ -exRec. The part in the dashline, without the preceding ECs, is a <i>Rec</i>	35
3.5	Example for rough lattice merging. The pink qubits are initialized in $ 0\rangle$	39
3.6	Example of lattice-splitting	40
3.7	Measurement-based CNOT. a and b are the measurement outcome of $M_{\bar{Z}\bar{Z}}$ and $M_{\bar{X}\bar{X}}$ respectively.	41
4.1	EXPEDIENT purification circuit (one side). Alice and Bob share 5 ebits initially. They then locally perform the circuit and measurements above. Upon comparing the 4 measurement results, if all of them agree, they keep the first ebit; otherwise discard.	45
4.2	(a) Direct Encoding Scheme <i>A</i> . $ \bar{+}\rangle$ and $ \bar{0}\rangle$ are prepared by Alice and Bob respectively. Transversal CNOTs are then applied to create $\bar{\text{ebit}}$. (b) Direct Encoding Scheme <i>B</i> . Alice and Bob start with high-fidelity ebits, Alice measures the X -type stabilizers and Bob measures the Z -type stabilizers. The measurements are repeated for FT and the outcomes are used for correction to ensure the resultant joint logical state is $\bar{\text{ebit}}$	53
4.3	The circuit for performing a CNOT with one ebit and local operations[97]. The first and last qubits are the independent qubits we hope to perform CNOT on with the first being control and the last being target. In the middle two sides share an ebit.	54
4.4	Interface $\text{Enc}_{0 \rightarrow 1}$	57
4.5	A broad overview of the Interface+EPP method. On the left-hand side, the curved lines represent ebits. The triangles are the interfaces. This is followed by $\bar{\text{EPP}}$ and classical communication.	58
4.6	(a) Scheme <i>A</i> . Encoding is done sequentially, level by level, each followed by a logical EPP with 3 other similarly prepared $\bar{\text{ebit}}$. (b) Scheme <i>B</i> . State is encoded to the k -th level in a single interface, followed by rounds of $\bar{\text{EPP}}^{(k)}$ to reduce the logical error rate.	59
4.7	Network for $\bar{X}$ -error. The source S and sink T are introduced for completeness. The numbers above the edges indicate the capacities. The pink edges all have capacity 1.	65

4.8	An example solution of the feasibility problem. This indicates an $\overline{X}$ -error on the first $\overline{\text{ebit}}$ given $\overline{\text{EPP}}$ accepts.	67
4.9		67
4.10	An Interface+EPP procedure broken down into its individual components. The orange parts denote $\overline{\text{ebits}}$ prepared via physical ebits and interfaces. Other parts correspond to exRecs within $\overline{\text{EPP}}$. For example, A denotes the first CNOT-exRec on Alice's side.	68
4.11	Two components A and B with an overlapping set of ECs	73
4.12	Circuit for FT magic square game	77
4.13	(a) Logical error rates of $\overline{\text{ebit}}^{(1)}$ prepared with <i>Direct Encoding</i> and <i>Interface+EPP</i> against physical error rate p . Both methods yield similar logical error rates, thereby justifying a subsequent comparison of resource overheads. (b) Overall success probability against ϵ for <i>Interface+EPP</i>	84
4.14	The number of raw ebits required against failure probability of the magic square game Δ . Log-scale is used for both axes.	85
5.1	(a) Example of a transversal implementation of an $\overline{\text{ebit}}$. (b) Example of a distant surface code $\overline{\text{ebit}}$ with lattice surgery. Alice and Bob (at a distance) each prepare $ \overline{0}\rangle$. Alice holds her code patch and all ancilla qubits for lattice surgery and stabilizer measurements. The pink dots are the ancilla qubits for lattice surgery and the purple dots are ancilla qubits for measurements. To perform the X_v measurements, Bob will need to connect to pink qubits via long-distance CNOTs, which are implemented via gate teleportation with a high-fidelity ebit.	88
5.2		90
5.3	Measurement-based teleportation	91
5.4	Lattice-surgery based teleportation	91
A.1	Comparison of bounds on logical error rate of CNOT-exRec	110

List of Tables

2.1	Stabilizer generators and logical operators of the Steane code	14
3.1	Pseudo-thresholds for 4 EC gadgets	38
4.1	Measurement basis for Alice and Bob when they are assigned different indices. Alice holds the row index while Bob holds the column index.	44
4.2	<u>Bounds</u> for the failure probability of the magic square game under different ebit preparation methods at concatenation level k	78
4.3	Results for l'_A, l'_B as in Lemma 4.5.3 for given k and m	80
4.4	Results for l''_A, l''_B as in Lemma 4.5.3 for given k and m	81

Chapter 1

Introduction

1.1 Motivation

Quantum Entanglement, one of the most profound phenomena in quantum mechanics, allows particles to exhibit correlations that cannot be explained by classical physics, even when separated by large distances. This nonlocal behavior has significant implications for various quantum technologies, such as quantum computation and communication. The entangled particles act as a single system, showcasing nonlocal effects that challenge classical notions of locality and providing a foundational resource for tasks that are otherwise impossible or inefficient with classical systems.

Nonlocal games are one of the scenarios where quantum entanglement demonstrates its power. A nonlocal game involves interaction between a referee and spatially isolated players, where the referee assigns questions to each player based on a publicly known distribution. Players respond to their questions, aiming to satisfy specific criteria and win the game. While players cannot communicate during the game, they can use quantum entanglement prior to the game to increase their chances of winning, thus highlighting the power of quantum resources in achieving correlations that surpass classical limits. Nonlocal games not only offer insights into the fundamental nature of quantum mechanics but also have practical applications in quantum cryptography and device-independent quantum key distribution [19].

Experimental demonstrations of nonlocal games, such as loophole-free Bell tests, have validated these theoretical predictions under various conditions [6, 35, 76, 42]. In recent years, special types of nonlocal games like quantum pseudotelepathy [15] have been explored, where players can achieve success probability 1 with quantum strategies. They

have also been verified to show quantum advantage [94]. For the Magic Square Game, in the presence of hardware errors, the average winning probability across all assignments of indices is 93.8%, surpassing the classical value but still suboptimal. This prompts the question: Can we achieve a winning probability of 1 when each gate in the circuit has a constant error rate? Additionally, in both communication and nonlocality, what is the minimal amount of resources necessary to achieve this? One key resource is quantum entanglement, or more specifically *EPR pairs*(*ebits*), that enables quantum advantage in nonlocal games. And this quantity will be a central focus of this work.

One promising solution to the first question is *Fault-tolerant Quantum Computation* (*FTQC*), which permits arbitrarily low logical error rates in the presence of physical noise, provided that the error rates of all operations remain below a threshold value [54, 2]. An established method for error suppression is code concatenation [52, 73] and it was later proved that by concatenating the $[[7, 1, 3]]$ Steane code [1], *FTQC* is possible with a theoretically proven lower bound on the threshold [3]. Recent advancements have revitalized interest in this approach, showcasing the feasibility of time-efficient and constant-space-overhead *FTQC*[96]. Although these code concatenation constructions work well under quantum computation settings, complexities arise when we hope to apply them to quantum communication or nonlocal game scenarios, which are underexplored areas. It was shown that in the noisy setting, capacities of quantum channels can be asymptotically approached with a fault-tolerant construction based on Steane code concatenation[26, 9]. However, an exact threshold value is unknown and the question of resources is not addressed. Another popular family of codes for *FTQC* is topological codes, or more specifically, surface codes. By encoding logical qubits in 2D arrays of physical qubits and using local measurements, they effectively manage both bit-flip and phase-flip errors. This code is highly favored due to its high error threshold and compatibility with existing quantum hardware technologies [32, 57, 13, 81].

1.2 Main results and thesis outline

In this work, we restrict to the context of *Fault-tolerant Magic Square Game* and provide a partial solution to the following question

Given some fixed noise strength in local devices, if Alice and Bob hope to play the magic square game with a success probability arbitrarily close to 1, what is the minimal number of ebits required?

One essential component to establish the results on fault-tolerant quantum communication is an interface circuit that translates between physical qubits and logical qubits.

A similar idea was experimentally tested with 9-qubit Shor's code[64]. This concept can potentially save ebits as it moves entangling operations from the logical space to the physical space. In the $[[7^k, 1, 3^k]]$ case, it circumvents the need for transversal entangling gates. However the interface itself is not fault-tolerant, so we invoke the idea of entanglement purification [10] to filter out bad ebits. Combining these ideas, we use a modified interface and propose an alternative scheme for preparing a logical ebit and subsequently playing the magic square game.

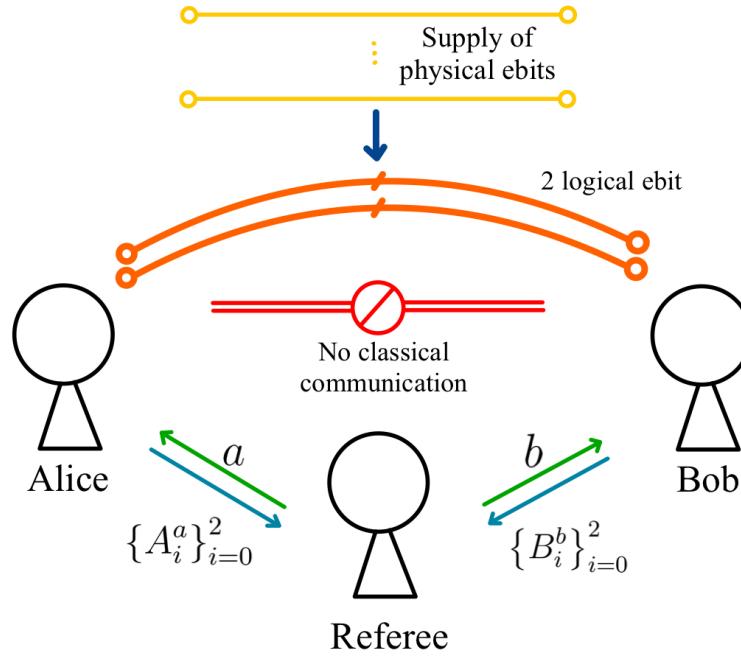


Figure 1.1: The procedure of the *Fault-tolerant Magic Square Game*. For the *Magic Square Game*, Alice and Bob are randomly assigned a row and a column index $a, b \in \{0, 1, 2\}$. They then each reply with three answers $[A_0^a, A_1^a, A_2^a]$ and $[B_0^b, B_1^b, B_2^b]$ with $A_i^a, B_j^b \in \{\pm 1\} \forall i, j$. The winning condition is that $\prod_{i=0}^2 A_i^a = +1$ and $\prod_{i=0}^2 B_i^b = -1$ and $A_b^a = B_a^b$. No communication is allowed during the game. However, if Alice and Bob share two EPR pairs prior to the game and perform corresponding measurements, they are able to win the game with probability 1 (in the error-free scenario). In the FT case, Alice and Bob are supplied with noisy physical ebits. In order to implement the same strategy, they use these to create 2 logical ebits and then perform logical measurements.

We’ve compared our scheme with *Direct Encoding*, that is, Alice and Bob prepare $|\overline{+}\rangle$ and $|\overline{0}\rangle$ respectively and they use transversal CNOT to create a logical ebit. We used numerics-assisted methods for $k \geq 1$ and performed exact Monte-Carlo simulation for $k = 1$. Let Δ denote the failure probability of the magic square game. Given that $0 < \Delta \ll 1$, physical error rate $\epsilon = 2.25 \times 10^{-4}$ and initial infidelity of ebits being 10%, our proposal shows substantial savings on the initial ebits consumed. A more explicit comparison can be seen in Figure 1.2.

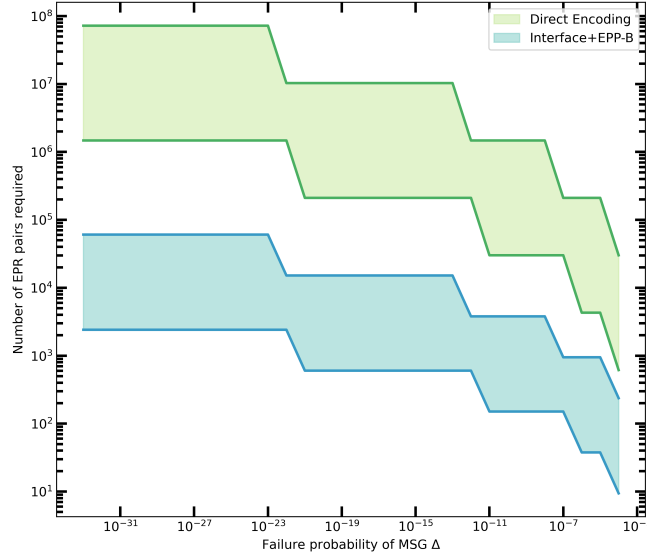


Figure 1.2: The number of raw ebits required to achieve magic square game failure probability Δ versus Δ . Due to the complexity of simulation when the concatenation level is high, analytical bounds are obtained instead. The green and blue bands above represent regions bounded by the upper and lower bounds for the two methods respectively.

In fact, there is another advantage of our scheme, the above ϵ is actually a derived lower bound on the threshold for *Direct Encoding*. Notably, *Interface+EPP* has the potential to tolerate higher local noise levels. Our analysis establishes a lower bound 4.70×10^{-4} . Additionally, this method is capable of handling significantly noisier initial ebits, with a lower bound on the infidelity threshold being 18.3%. In the later part of this thesis, we also construct an interface for the surface code through lattice-surgery [44].

The rest of this thesis is organized as follows. In Chapter 2 we review the basics of

quantum error correction, emphasizing the underlying intuition. We elaborate on a less discussed yet important topic: the evolution of stabilizers. Additionally, we provide an overview of the entanglement purification protocol (EPP) towards the end of this chapter. In Chapter 3 we review important notions in FTQC that will be used throughout this thesis. We further give a detailed comparison of different fault-tolerant error correction gadgets, including their pseudo-thresholds and qubit overheads. The specific fault-tolerant constructions in the Steane code and surface code are discussed at the end. The *Magic Square Game* is introduced in Chapter 4.1. In Chapter 4 we elaborate and compare the different methods of preparing a logical ebit for concatenated Steane code. We begin by discussing the preparation of high-fidelity ebits from noisy ebits. In Section 4.3, we derive a lower bound on the threshold. Using these results, Section 4.4 presents bounds on the logical error rate of a logical ebit prepared via *Direct Encoding*. Section 4.5 provides an overview of *Interface+EPP* methods, and describes our modified interface. We then obtain bounds on the logical error rate of the *Interface+EPP* schemes in Section 4.5.4. In Section 4.6, we combine the results and obtain our main result as above. Finally, we present a full numerical simulation comparing the methods for encoding level-1. Lastly, we summarize our results and outline future directions in Sec.6.

Chapter 2

Quantum Error Correction

2.1 Notations

Throughout this thesis, we shall assume familiarity with some standard notions and their notations in quantum information and computation. More detailed descriptions can be found in [68]. The Hilbert space for a quantum system, a complete inner product space, is denoted as $\mathcal{H}$. We denote $\mathcal{L}(V, W)$ as the set of linear operators with domain V and range W and $\mathcal{L}(V)$ if the domain and range are the same space. A density operator ρ is a positive operator such that $\text{Tr}\rho = 1$. The set of density operators acting on $\mathcal{H}$ is denoted $D(\mathcal{H})$. For the coding theory part, we use single-bracket $[n, k, d]$ to denote a classical code and double-bracket $[[n, k, d]]$ to denote a quantum code. $[P, Q] = PQ - QP$ is the commutator and $\{P, Q\} = PQ + QP$ is the anticommutator. Other notations will be clearly defined when they come into use. In the rest of this thesis, unless specified we will assume the QECC in use has parameters $[[n, k, d]]$ and $t = \lfloor \frac{d-1}{2} \rfloor$.

We will start by giving a very brief overview of *Quantum Error Correction*(QECC) with a focus on intuition, neglecting proofs of results and topics irrelevant to this work, ensuring this thesis is self-contained. Readers can refer to Gottesman's PhD thesis [37] or his textbook [36] for more details. Concepts and results covered in this section include the quantum error correction criteria, and the stabilizer formalism, notably, we will elaborate on the procedure of encoding a state in the logical space. Further, we discuss CSS codes and their properties, and details of the two error-correcting codes considered in this thesis, namely the Steane code and the surface code.

2.2 QECC Condition

Quantum errors can be generally modeled by quantum channels. Given a state $\rho \in D(\mathcal{H}_n)$, the Kraus representation of an error channel $\mathcal{E} : D(\mathcal{H}_n) \rightarrow D(\mathcal{H}_m)$ can be written as $\mathcal{E}(\rho) = \sum_s E_s \rho E_s^\dagger$, where $E_s \in \mathcal{L}(H)$ and $\sum_s E_s^\dagger E_s = I$. Some common error models include depolarizing noise, amplitude damping noise, and erasure noise. Before stating the criteria, I will formulate the error correction procedure more precisely. Part of the following is referenced from [60]. A full cycle of QECC can be described as in Fig 2.1, where $U : \mathcal{H}_k \rightarrow \mathcal{H}_n$ denotes the encoder, which is an isometry¹ mapping the physical qubits into the logical space. $E \in \mathcal{E}$ denotes the set of correctable errors. $D : \mathcal{H}_m \rightarrow \mathcal{H}_k$ denotes the decoder that corrects the corrupted state, which includes syndrome measurement and recovery operations. A valid QECC will output a state that is proportional to the original state.

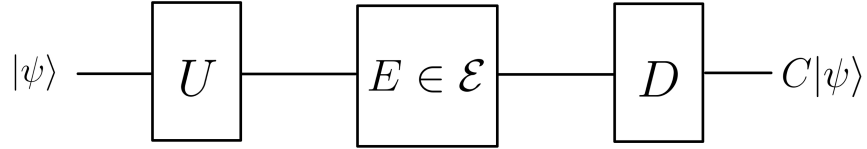


Figure 2.1: The full procedure for a QECC cycle

$\mathcal{H}_k$ is the *logical space* and $\mathcal{H}_n$ is the *physical space*. In this case, we say a QECC encodes k logical qubits into n physical qubits. At this point, we can state the quantum error correction condition [53]

Definition 2.2.1 (QECC criterion). Suppose we have the error channel $\mathcal{E}(\cdot) = \sum_s E_s(\cdot)E_s^\dagger$ and an isometry operator U . Let $\{|\psi_i\rangle\}$ be a basis of $\mathbf{Im}(U)$. Then $(U, \mathcal{E})$ is a *QECC* iff $\forall i, j, s, t$

$$\langle \psi_i | E_s^\dagger E_t | \psi_j \rangle = c_{s,t} \delta_{ij}$$

In particular, $c_{s,t}$ are entries of a Hermitian matrix.

In the above, $\mathbf{Im}(U)$ is in fact what we call the *codespace*. Intuitively, a set of errors is correctable by a QECC if the basis states, after the action of the error operators, remain

¹Recall that an operator U is an isometry if the distance between two points in the domain is preserved after the action of U . On a Hilbert space, this corresponds to a unitary operator.

orthogonal and therefore are perfectly distinguishable. Otherwise, in the degenerate case, different E_i 's maps have the same effect on the codeword, this is also reversible via the same recovery operation. In this work, we focus on the depolarizing noise. It has the form that

$$\mathcal{E}_p(\rho) = (1-p)\rho + \frac{p}{3}X\rho X^\dagger + \frac{p}{3}Y\rho Y^\dagger + \frac{p}{3}Z\rho Z^\dagger = \left(1 - \frac{4p}{3}\right)\rho + \frac{4p}{3}\frac{I}{2}$$

We say an error E has weight- t if it affects t qubits, denoted $\text{wt}(E)$. Given this we can define another essential parameter of a QECC, *distance*.

Definition 2.2.2 (Distance). Let $\mathcal{C} \subseteq \mathcal{H}_n$ be the codespace of some QECC. The distance of the QECC is defined as

$$d = \min\{\text{wt}(E) : \langle \psi | E | \phi \rangle \neq c \langle \psi | \phi \rangle, \forall |\psi\rangle, |\phi\rangle \in \mathcal{C}\}$$

We denote the QECC to have parameters $[[n, k, d]]$ and the rate of the code is $R = \frac{k}{n}$.

2.3 Stabilizer Formalism

While the QECC condition helps to verify, given error $\mathcal{E}$, whether a code is a valid QECC, it's hard to derive explicit codes from it. Therefore we will now review a more convenient framework of QECC, namely the stabilizer formalism, developed by Gottesman [37]. The group theory nature gives a natural decomposition of the Hilbert space, which we elaborate on below. Before explaining the formulation, we will first introduce the Pauli group.

Definition 2.3.1 (Pauli group). The Pauli group on n qubits consists of tensor product of $\{I, X, Y, Z\}$ with an overall phase of $\pm 1, \pm i$, where

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

There are a couple of properties worth mentioning. (1) $P^2 = \pm I, \forall P \in \mathcal{P}_n$. (2) $\forall P, Q \in \mathcal{P}_n$, either $[P, Q] = 0$ or $\{P, Q\} = 0$. Due to these properties of $\mathcal{P}_n$, the stabilizer group defined below divides the codespace.

Definition 2.3.2 (Stabilizer group). Given a QECC $T \subseteq \mathcal{H}_{2^n}$, the stabilizer group S is an abelian subgroup of $\mathcal{P}_n$ defined as

$$S(T) = \{Q \in \mathcal{P}_n | Q|\psi\rangle = |\psi\rangle \forall |\psi\rangle \in T\}$$

From the definition we see that $-I \notin S(T)$. Now we can define a stabilizer code,

Definition 2.3.3 (Stabilizer code). Given a stabilizer group $S \subset \mathcal{P}_n$, a stabilizer code $\mathcal{C}$ is the joint +1-eigenspace of S ,

$$\mathcal{C}(S) = \{|\psi\rangle \text{ s.t. } Q|\psi\rangle = |\psi\rangle \ \forall Q \in S\}$$

If there are n physical qubits and the group S has r generators, then there are $k = n - r$ logical qubits. Moreover we define the normalizer of S .

Definition 2.3.4 (Normalizer). The normalizer of the stabilizer S is

$$N(S) = \{N \in \mathcal{P}_n | NQ = QN \ \forall Q \in S\}$$

From the above definitions, we can infer a more concise expression for the distance of a stabilizer code.

Definition 2.3.5 (Distance of a stabilizer code). For a stabilizer code S , $N(S) \setminus S$ is the set of undetectable errors and the distance d is

$$d = \min\{\text{wt}E | E \in N(S) \setminus S\}$$

Elements in S have trivial actions on the code and elements $N(S) \setminus S$ are actually *logical operators*. We further note that the quotient group $N(S)/S \cong \mathcal{P}_k$. The disjoint cosets of S in $N(S)$ therefore represent distinct undetectable errors (logical operations) up to stabilizer transformation. We now have the ingredients for constructing a stabilizer code. Given blocklength n , we may first choose a stabilizer group $S \in \mathcal{P}_n$ with r generators $\{Q_1, \dots, Q_r\}$. There are $k = n - r$ logical qubits. Thus we find $\{\bar{X}_i, \bar{Z}_i\}_{i=1}^k$ such that each logical operator commutes with all Q_i as well as logical operators of the same type, i.e. $[\bar{X}_i, \bar{X}_j] = [\bar{Z}_i, \bar{Z}_j] = 0, \ \forall i, j \in [k]$. Furthermore $[\bar{X}_i, \bar{Z}_j] = 0$ for $i \neq j$ and $\{\bar{X}_i, \bar{Z}_i\} = 0$. Now given n physical qubits, how do we encode them in the logical space? Actually, we only need to apply the following operator to project onto the codespace

$$P_S = \frac{1}{2^r} \prod_{i=1}^r (I + Q_i) = \frac{1}{2^r} \sum_{Q \in S} Q$$

As discussed above, each stabilizer divides the codespace into two parts, with each half corresponding to either the +1 or -1 eigenspace. The operator $(I + Q_i)$ ensures that we always project onto the +1 eigenspace. Thus, the first equality is easily derived. The

second equality follows from expanding the first expression. If we want to obtain $|\bar{0}\rangle^{\otimes k}$, which is the +1 eigenstate of each $\bar{Z}_i$, we do similar as above and have

$$|\bar{0}\rangle^{\otimes k} \propto \prod_{i=1}^k (I + \bar{Z}_i) \prod_{i=1}^r (I + Q_i) |00 \dots 0\rangle$$

At the end, we want to discuss a subroutine in decoding a stabilizer code. In general, efficient decoding is hard. For most families of codes, it is NP-hard. However, for stabilizer codes we can perform syndrome measurements to obtain some information. We first need to define the error syndrome.

Definition 2.3.6 (Error syndrome). Let $S = \langle Q_1, \dots, Q_{n-k} \rangle$, the syndrome $q \in \{0, 1\}^{n-k}$ of a Pauli error E is defined as

$$q_i = \begin{cases} 0, & \text{if } [Q_i, E] = 0 \\ 1, & \text{if } \{Q_i, E\} = 0 \end{cases}$$

for a state $|\psi\rangle$ s.t. $Q|\psi\rangle = q|\psi\rangle \forall Q \in S$, the syndrome has the i -th bit being 0 if $Q_i|\psi\rangle = |\psi\rangle$ and 1 if $Q_i|\psi\rangle = -|\psi\rangle$.

We now show how to measure the parity of an operator (which combines to give the error syndrome). If we have some 5-qubit code and we hope to measure the stabilizer $XZZXI$ we may perform the circuit

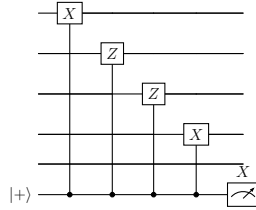


Figure 2.2: Non-fault-tolerant syndrome measurement circuit for the operator $XZZXI$

2.4 Quantum Codes from Classical Codes

Having introduced the stabilizer formalism, we take an interlude and discuss a specific method for constructing a stabilizer code, the CSS construction [20, 1]. Classical coding

theory is a well-established field [87], with numerous results that underpin much of our modern computational and communication technologies. Thus people naturally wonder if we can make use of these results in QECC. However, a significant challenge in quantum computing is the need to correct for two types of errors, X and Z . Fortunately, the formalism developed by Calderbank, Shor, and Steane allows us to construct quantum codes by combining classical linear codes. We will start by discussing linear codes and then the CSS construction.

Definition 2.4.1 (Linear code). A linear code $\mathcal{C}$ of dimension k and blocklength n over a finite field $\mathbb{F}$ is a k -dim subspace of $\mathbb{F}^n$.

In the 0/1 case, we have $\mathbb{F} = \mathbb{Z}_2$. As we are not concerned with qudit error correction in this thesis, we will keep this convention from now on. Due to the linear subspace requirement, we have $\forall x, y \in \mathcal{C} \Rightarrow x + y \in \mathcal{C}$. This also allows us to use linear algebraic tools. $\mathcal{C}$ can be represented by a generator matrix, defined as below:

Definition 2.4.2 (Generator matrix). Let $\mathcal{C} \subseteq \mathbb{Z}_2^n$ be a linear code of dimension k . A matrix $G \in \mathbb{Z}_2^{n \times k}$ is a generator matrix for $\mathcal{C}$ if $\mathcal{C} = \{G \cdot x : x \in \mathbb{Z}_2^k\}$.

In fact, we often consider $\mathcal{C}$ from a ‘dual’ perspective, the dual code of $\mathcal{C}$ is defined as $C^\perp = \{v \in \mathbb{Z}_2^n : \langle v, c \rangle = 0 \forall c \in \mathcal{C}\}$. We say an $[n, k]$ code $\mathcal{C}$ is weakly self-dual if $C^\perp \subseteq \mathcal{C}$ and self-dual if $C^\perp = \mathcal{C}$. We note that $\dim \mathcal{C} + \dim \mathcal{C}^\perp = n$. If we consider the generator matrix of a dual code $C^\perp$, denoted by H , this matrix possesses several notable properties. These properties are significant enough to warrant a separate definition.

Definition 2.4.3 (Parity-check). Let $\mathcal{C} \subseteq \mathbb{Z}_2^n$ be a linear code of dimension k . A matrix $H \in \mathbb{Z}_2^{(n-k) \times n}$ is a parity-check matrix if $\mathcal{C} = \ker H = \{x \in \mathbb{Z}_2^n : H \cdot x = 0\}$.

Since H^T is the generator matrix of $C^\perp$, we can infer that $HG = 0$. And now G^T is the parity-check of $C^\perp$. The name ‘parity-check’ comes from the fact that, given $x \in \mathcal{C}$ a codeword, if an error $e \in \mathbb{Z}_2^n$ occurs, we now have a string $x + e$ and

$$H(x + e) = 0 + He = He$$

which is a bit string we call the ‘error syndrome’. Classically, the distance of $\mathcal{C}$ is the minimum weight of any nonzero $c \in \mathcal{C}$. It’s also the smallest number d such that H has d linearly dependent columns. A linear code with distance d can correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors and detect up to $d - 1$ errors. This is best illustrated by Fig 2.3.

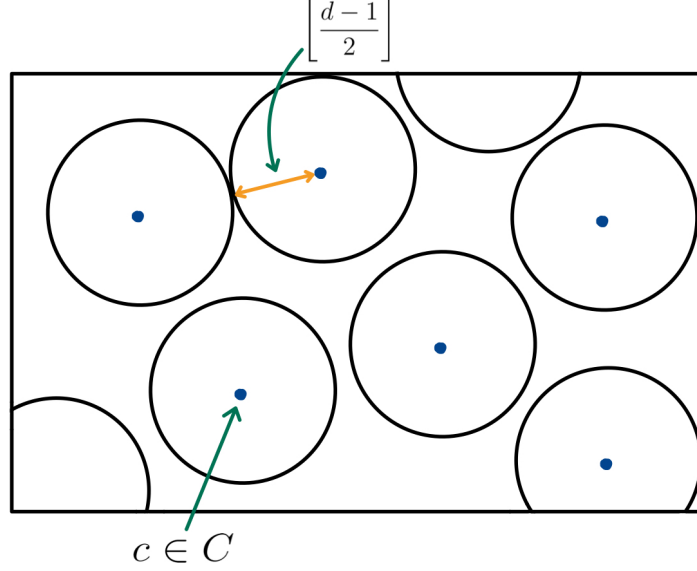


Figure 2.3: Space packed by a code $\mathcal{C}$ with distance d . Each circle is a Hamming ball with center being a codeword $c \in \mathcal{C}$ and radius $\lfloor \frac{d-1}{2} \rfloor$ to ensure circles are disjoint so different codewords are distinguishable.

We will give an example of classical linear code, the binary $[2^r - 1, 2^r - r - 1, 3]$ Hamming code [41], commonly used in ECC RAM. This is a class of codes with parity-check matrices having as columns all r -bit strings. When $r = 3$, we have the $[7, 4, 3]$ code with parity-check matrix

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

and generator matrix

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

It's easy to check that $HG = 0$ and the syndrome of error on the i -th bit is the i -th column

of H . We will see later how we obtain the Steane code from this code. Now we define the CSS code

Definition 2.4.4 (CSS code). Given a pair $(\mathcal{C}_X, \mathcal{C}_Z)$ of binary linear codes satisfying $\mathcal{C}_Z^\perp \subseteq \mathcal{C}_X$. The codes have parity-check matrices H_X and H_Z respectively. Let S be the CSS code with stabilizer generators

$$\begin{pmatrix} 0 & H_Z \\ H_X & 0 \end{pmatrix}$$

then S is an $[[n, k, d]]$ quantum code with $k = \dim(\mathcal{C}_X/\mathcal{C}_Z^\perp) = \dim(\mathcal{C}_Z/\mathcal{C}_X^\perp) = \dim(\mathcal{C}_X) + \dim(\mathcal{C}_Z) - n$ and $d = \min\{d_X, d_Z\}$.

We note that $\mathcal{C}_Z^\perp \subseteq \mathcal{C}_X$ immediately implies $H_X H_Z^T = 0$. This explicit equation also inspires a later generalization for code construction by connecting to homology [33, 69]. Essentially, a CSS code can be seen as a chain complex of size three. If we call X_0, X_1, X_2 the sets of rows of H_X , columns of H_X and rows of H_Z and the boundary operators being $\partial_1 = H_X$ and $\partial_2 = H_Z^T$, we then have

$$\mathbb{F}_2^{X_2} \xrightarrow{\partial_2} \mathbb{F}_2^{X_1} \xrightarrow{\partial_1} \mathbb{F}_2^{X_0}$$

such that $\partial_1 \partial_2 = 0$. As a matter of fact, the CSS construction remains the base for some of the latest QECCs, including several popular qLDPC codes² [88, 61]. Since this is not the focus of this thesis, we defer the details to interested readers. Finally, as an aside, we introduce an important concept,

Definition 2.4.5 (Code concatenation). If we have two QECC $\mathcal{C} = [[n_1, k_1, d_1]]$ and $\mathcal{C}' = [[n_2, k_2, d_2]]$, we may combine these two codes by first encodes in the *inner code* $\mathcal{C}$ and then encodes each of the physical qubits of $\mathcal{C}$ in an outer code $\mathcal{C}'$. This yields a $[[n_1 n_2, k_1, d \geq d_1 d_2]]$ concatenated quantum code.

We may extend this to higher levels. In Chapter 4 the concatenated Steane code $[[7^k, 1, 3^k]]$, where k is the level of concatenation, will be the premise of our analysis.

We will now present two examples of CSS code that are used in this thesis.

²Roughly speaking, a code is LDPC if its parity-check matrix is sparse.

2.4.1 Steane Code

The Steane code is the smallest qubit CSS code capable of correcting a single-qubit Pauli error [1]. In recent years, due to its simplicity and effectiveness, as well as its nice fault-tolerance properties, there has been an increasing amount of experimental success in demonstrating the Steane code in trapped-ion [78, 79, 71, 28] and neutral-atom systems [13, 12]. The Steane code is constructed by setting $H_X = H_Z =$ parity-check of the [7,4,3]-Hamming code. From Definition 2.4.4 we see that the code parameters are $[[7, 1, 3]]$ and the stabilizers and logical operators are

$$\begin{array}{rcl}
S_1 & = & I \quad I \quad I \quad X \quad X \quad X \quad X \\
S_2 & = & I \quad X \quad X \quad I \quad I \quad X \quad X \\
S_3 & = & X \quad I \quad X \quad I \quad X \quad I \quad X \\
S_4 & = & I \quad I \quad I \quad Z \quad Z \quad Z \quad Z \\
S_5 & = & I \quad Z \quad Z \quad I \quad I \quad Z \quad Z \\
S_6 & = & Z \quad I \quad Z \quad I \quad Z \quad I \quad Z \\
\overline{X} & = & X \quad X \quad X \quad X \quad X \quad X \quad X \\
\overline{Z} & = & Z \quad Z \quad Z \quad Z \quad Z \quad Z \quad Z
\end{array}$$

Table 2.1: Stabilizer generators and logical operators of the Steane code

The decoding of the Steane code is particularly simple. Each error syndrome corresponds to a unique X/Z error. Thus a Look-Up-Table is sufficient for decoding.

2.4.2 Surface Code

In some experimental platforms such as superconducting qubits, it is desirable to have a planar code with nearest-neighbour interactions. This makes the planar surface code, which is the 2D version of the Kitaev toric code [48], a strong candidate for such systems [32, 57, 13]. Besides, the surface code saturates the BPT bound [17]. This means under some locality constraints, it is optimal among all 2D codes. The surface code is defined on a $L \times (L - 1)$ -square lattice. An example of the surface code is shown in Figure 2.5. There are two types of stabilizers - the plaquette operators Z_p and the star operators X_v defined as

$$X_v = \prod_{l \in \text{star}} X_l, \quad Z_p = \prod_{l \in \partial(p)} Z_l$$

where ‘star’ is associated with a vertex v , it is the collection of qubits on edges incident to v ; and ‘ $\partial(p)$ ’ denotes the qubits on the boundary of the plaquette p . The error syndrome for each stabilizer can be measured by coupling the corresponding 4 qubits to an ancilla qubit and use circuit 2.2. It is evident that the stabilizers commute because X_v and Z_p overlap on at most two qubits. Exemplar stabilizers are shown in Figure 2.4.

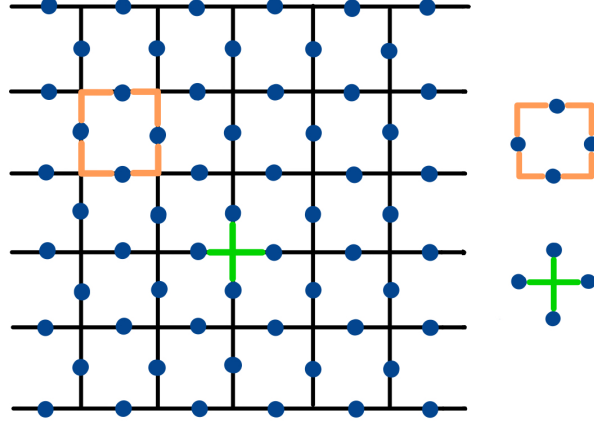


Figure 2.4: An example of a surface code and its stabilizers. This surface code patch consists of 61 qubits, represented by blue dots in the figure. The intersections of black lines are ‘vertices’ and a ‘plaquette’ is either a square face in the middle or the region bounded by 3 qubits on two sides. The green operator crossing a vertex is the star operator X_v and the orange operator bounding a plaquette is the plaquette operator Z_p . The top and bottom boundaries are called “smooth” while the left and right boundaries are called “rough boundaries”.

Next we discuss the possible operators that might act on a surface code. We may broadly split into three categories:

1. When the edge qubits of the operator form loops contractible to a torus, also known as *homologically trivial cycles*. Such a cycle on the square lattice can be tiled by plaquettes, and thus it is a product of Z_p -operators in the stabilizer group. Similarly, a product of X_v acting on edges of a homologically trivial cycle on the dual lattice is also in the stabilizer and, thus does not cause an error. An example can be found in

Figure 2.5a, where a Z_p -type cycle (in orange) and a X_v -type cycle (in green), and their respective tiling by Z_p and X_v are shown. It's worth noting that the X_v -type cycle is drawn on the dual lattice (in light blue) and it contains a hole.

2. Another possible operator is walks with distinct ends, or open paths in graph-theoretic terms. In this case, they are detectable errors, to be corrected by later decoding. An example of Z -error is shown in Figure 2.5b. We can see that the path AB anticommutes with X_A and X_B .
3. Finally we have the case that the operator forms loops that are not contractible, or *homologically non-trivial cycle*. In this case, the operator is in fact a logical operator. As demonstrated in Figure 2.5c, the purple path is a $\bar{Z}$ -operator and the operator in red is a $\bar{X}$ -operator. Any excursion into the code patch is equivalent up to applications of stabilizers. In other words, $\bar{Z}(\bar{X})$ -operators are paths that connect rough(smooth) boundaries. It is then obvious that any logical operator must have weight at least L .

From the above constructions, we can infer that the surface code has parameters $[[L^2 + (L - 1)^2, 1, L]]$.

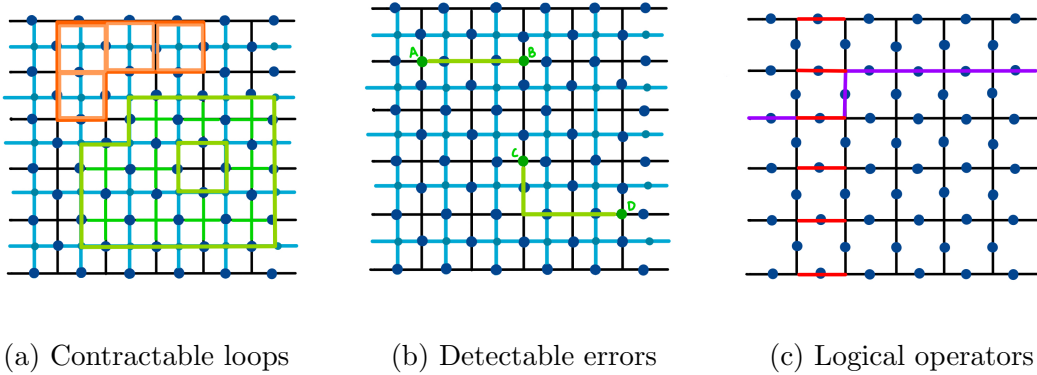


Figure 2.5: Possible types of operators on a surface code. (a) Homologically trivial cycles, i.e. boundary of plaquettes in the original and dual lattices. The dual lattice is coloured light blue. The orange closed path is an example of Z -type cycle and can be tiled by Z_p -stabilizers. The green closed path is an example of X -type cycle and can be tiled by X_v -stabilizers. (b) Open path. These are detectable errors. The green path is an example of Z -type errors. (c) Homologically non-trivial cycles. The purple path represents an example of $\bar{Z}$ and the red represents an example of $\bar{X}$.

Now we discuss the decoding problem. From a condensed matter perspective of the surface code, excitations come in pairs and thus path endpoints come in pairs. Therefore, having detected a number of erroneous qubits on the code patch, the recovery procedure consists of pairing up these qubits using the least amount of resources. By forming the syndrome graph and assigning weight to each edge (typically in surface code scenario we set $\text{wt}(e) = \log \frac{1-p}{p}$ where p is the physical error rate), we can see the problem as a *Minimum-Weight-Perfect-Matching(MWPM)* problem in graph theory³. Given the weighted syndrome graph being $G = (V, E)$, this is formulated as the following integer programming

$$\begin{aligned} \min \quad & \sum_{e \in E} \text{wt}(e) x_e \\ \text{s.t.} \quad & \sum_{e \in S(v)} x_e = 1 \quad \forall v \in V \\ & x_e \in \{0, 1\} \quad \forall e \in E \end{aligned}$$

where $S(v) = \{e : \exists u \in V \text{ s.t. } e = (u, v) \in E\}$ is the set of edges incident to vertex v . This problem can be solved efficiently in polynomial time using the *Blossom algorithm* [31]. Figure 2.5b shows an example. A, B, C, D are the excitations and the green lines are matching after performing *MWPM*.

2.5 Evolution of Stabilizers

In this section, we discuss a less often mentioned yet very important topic: the evolution of stabilizers under Clifford unitaries and Pauli measurement. We will first define the Clifford group.

Definition 2.5.1 (Clifford group). The Clifford group C_n on n qubits is

$$C_n = \{U \in \mathcal{U}(2^n) : U P U^\dagger \in \mathcal{P}_n \quad \forall P \in \mathcal{P}_n\}$$

Put simply, the Clifford group is the set of operators that map Paulis to Paulis under conjugation. One set of generators of the Clifford group C_n is $\{e^{i\theta} I, H_i, R_{i,\pi/4}, \mathbf{CNOT}_{ij}\}$ for $i, j \in [n]$ and $\theta \in [0, 2\pi)$. The Clifford group seems to have desirable properties. The

³Given a graph $G = (V, E)$, a perfect matching in G is a set of edges $M \subseteq E$ such that every vertex in V is adjacent to exactly one edge in M .

transformation of single- and two- qubit Paulis under these generators are

$$\begin{aligned}
HXH &= Z, \quad HYH = -Y, \quad HZH = X \\
R_{\pi/4}XR_{\pi/4}^\dagger &= Y, \quad R_{\pi/4}YR_{\pi/4}^\dagger = -X, \quad R_{\pi/4}ZR_{\pi/4}^\dagger = Z \\
\mathbf{CNOT}(X \otimes I)\mathbf{CNOT}^\dagger &= X \otimes X, \quad \mathbf{CNOT}(I \otimes X)\mathbf{CNOT}^\dagger = I \otimes X \\
\mathbf{CNOT}(Z \otimes I)\mathbf{CNOT}^\dagger &= Z \otimes I, \quad \mathbf{CNOT}(I \otimes Z)\mathbf{CNOT}^\dagger = Z \otimes Z.
\end{aligned}$$

As conjugation is a group homomorphism, i.e. $U(P_1P_2)U^\dagger = UP_1U^\dagger UP_2U^\dagger$, compositions of operators can be obtained. However, the Gottesman-Knill theorem states that stabilizer circuits, which consist of Clifford gates, can be simulated on a probabilistic classical computer in polynomial time. Therefore to harness the power of quantum computation, we would need a gate outside of the Clifford group to complete a universal gate set. Usually, this gate is taken to be the CCZ -gate or T -gate. It is implemented in the circuit via ‘magic state injection’, essentially gate-teleportation [97].

We will give an example of how these transformations can be applied in practice. At the end of Section 2.3 we reviewed theoretically the process of encoding n physical qubits in the state $|0\rangle^{\otimes n}$ into the logical space, though an explicit circuit is not easily deduced. In this example, we derive an encoding circuit for $|\bar{0}\rangle$ of the Steane code by reverse-engineering the evolution of stabilizers under Clifford gates. Since we hope to obtain $|\bar{0}\rangle$, we have the stabilizers $\{S_1, \dots, S_6, \bar{Z}\}$ from Table 2.1. The idea is to find Clifford gates to simplify them to

$$\begin{array}{cccccc}
X & I & \dots & \dots & \dots & I \\
I & X & I & \dots & \dots & \dots \\
& \dots & \dots & \dots & & \\
\dots & I & X & I & \dots & \dots \\
\dots & \dots & I & Z & I & \dots \\
& \dots & \dots & \dots & & \\
\dots & \dots & \dots & I & Z & I \\
I & \dots & \dots & \dots & I & Z.
\end{array}$$

In other words, reduce to a form where each row and column has weight 1, and run backward from a tensor product of $|0\rangle$ ’s and $|+\rangle$ ’s (as $X|+\rangle = |+\rangle$ and $Z|0\rangle = |0\rangle$). We

now show the procedure:

$$\begin{array}{cccc}
IIIXXXX & IIIXXXX & IIIXIXX & IIIXIXX \\
IXXIIXX & IXXIIXI & IXXIIXI & IXXIIXI \\
XIXIXIX & XIXIXII & XIXIXII & XIIIXII \\
IIIZZZZ \xrightarrow{\text{CNOT}_{37}} IIIZZZZ & \xrightarrow{\text{CNOT}_{45}} IIIZIZZ & \xrightarrow{\text{CNOT}_{13}} ZIZIZZZ & \\
IZZIIZZ & IZIIIZZ & IZIIIZZ & IZIIIZZ \\
ZIZIZIZ & ZIIIZIZ & ZIIZZIZ & ZIIZZIZ \\
ZZZIIII & ZZIZZZZ & ZZIIZZZ & ZZIIZZZ
\end{array}$$

$$\begin{array}{ccc}
IIIXIXI & IIIXIXI & IIIXIXI \\
IXXIIXI & IXXIIII & IXXIIII \\
XIIIXII & XIIIXII & XIIIIII \\
\xrightarrow{\text{CNOT}_{47}} ZIZZZZZ & \xrightarrow{\text{CNOT}_{26}} ZZZZZZZ & \xrightarrow{\text{CNOT}_{15}} IZZZZZZ \\
IZIZIZZ & IIIZIZZ & IIIZIZZ \\
ZIIIZIZ & ZIIIZIZ & IIIIZIZ \\
ZZIZZZZ & ZIIZZZZ & IIIZZZZ
\end{array}$$

$$\begin{array}{ccc}
IIIXIXI & IIIXIII & XIIIIII \\
IXXIIII & IXXIIII & IXXIIII \\
XIIIIII & XIIIIII & IIZIIII \\
\xrightarrow{\text{CNOT}_{23}} IIZZZZZ & \xrightarrow{\text{CNOT}_{46}} IIZIZZZ & \rightarrow IIIXIII \\
IIIIZZZ & IIIIIZZ & IIIIZII \\
IIIIIZI & IIIIZIZ & IIIIIZI \\
IIIZZZZ & IIIIIZZ & IIIIIIZ
\end{array}$$

where the last step is obtained by multiplying stabilizers in the previous step and rearranging them. Therefore we derive a non-fault-tolerant⁴ encoding circuit by taking these gates in reverse order as in the figure below.

⁴We will explain what this means in a later section.

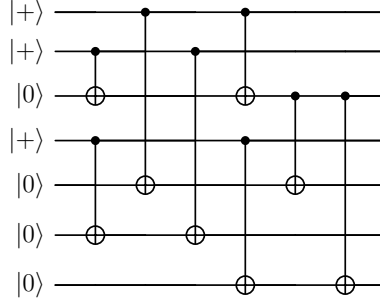


Figure 2.6: $|\bar{0}\rangle$ encoding circuit for the Steane code.

The encoding circuit is not unique. The one we present here uses relatively fewer CNOTs, thus limiting the occurrence of faults. We observe that any CSS state can be prepared using only $|+\rangle$, $|0\rangle$, and CNOT. More generally we have the following result

Theorem 2.5.2. *Any n -qubit stabilizer state can be prepared from $|0\rangle^{\otimes n}$ using $\mathcal{O}(n^2)$ one- and two-qubit Clifford gates.*

An example of the encoding of a non-CSS code can be found in [36]. In fact, the stabilizer formalism is useful beyond QECC. We can use it to infer the quantum state after a quantum circuit. Before illustrating this with an example, we will explore another critical element of a quantum circuit: measurement. Specifically, we aim to understand how the stabilizers are affected by measuring various operators $P \in \mathcal{P}_n$. Assuming the stabilizer group in the context is $S = \langle Q_1, \dots, Q_{n-k} \rangle$, we divide $P \in \mathcal{P}_n$ into three cases.

1. $\pm P \in S$. The state is unchanged since it is an eigenstate of P . Measurement outcome is +1 if $P \in S$ and -1 if $-P \in S$.
2. $P \in N(S) \setminus S$. If the original encoded state is $|\psi\rangle$, then the post-measurement state is $\frac{1}{\sqrt{2}}(I + bP)|\psi\rangle$ where $b = \pm 1$. Thus this state is a +1 eigenstate of bP , which indicates that bP is now also a stabilizer. Now we consider how the generators of $N(S)/S = \langle \bar{X}_1, \dots, \bar{X}_k, \bar{Z}_1, \dots, \bar{Z}_k \rangle$ change. Since $P \in N(S) \setminus S$, by definition of $\bar{X}_i, \bar{Z}_j$, at least one of these generators anticommutes with P . Without loss of generality we assume $\{\bar{X}_1, P\} = 0$. For other logical operators, if any of them anticommutes with P , we multiply $\bar{X}_1$ to it. Through this procedure we obtain $\bar{X}'_2, \dots, \bar{X}'_k, \bar{Z}'_2, \dots, \bar{Z}'_k$ that commute with all the new stabilizers. In summary, we have the following

Stabilizers before	Stabilizers after
Q_1	Q_1
$\dots$	$\dots$
Q_{n-k}	Q_{n-k}
	bP
Logical operators before	Logical operators after
$\overline{X}_1, \overline{Z}_1$	$\overline{X}'_2, \overline{Z}'_2$
$\dots$	$\dots$
$\overline{X}_k, \overline{Z}_k$	$\overline{X}'_k, \overline{Z}'_k$

3. $P \notin N(S)$. Because of this, $\exists Q_i \in S$ s.t. $\{Q_i, P\} = 0$. $\forall |\psi\rangle \in T(S)$, the probability of obtaining parity +1 is

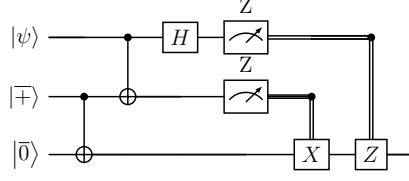
$$\begin{aligned}
\mathbb{P}(\text{parity} + 1) &= \text{Tr} \left(\frac{1}{2}(I + P)|\psi\rangle\langle\psi| \right) \\
&= \frac{1}{2} + \frac{1}{2}\langle\psi|P|\psi\rangle \\
&= \frac{1}{2} = \mathbb{P}(\text{parity} - 1)
\end{aligned}$$

The third equality holds because $\langle\psi|P|\psi\rangle = \langle\psi|PQ_i|\psi\rangle = -\langle\psi|Q_iP|\psi\rangle = -\langle\psi|P|\psi\rangle$. Similarly for $b = \pm 1$, the measurement outcome, bP is now promoted to the stabilizer. Other stabilizer generators and logical operators transform as

- $\forall j \neq i$, if $[Q_j, P] = 0$, then $Q'_j = Q_j$; if $\{Q_j, P\} = 0$, then $Q'_j = Q_j Q_i$
- $\forall j = 1, \dots, k$, if $[\overline{X}_j, P] = 0$, then $\overline{X}'_j = \overline{X}_j$; if $\{\overline{X}_j, P\} = 0$, then $\overline{X}'_j = \overline{X}_j Q_i$.

In summary, we first find Q_i that anticommutes with P and for any Q_j or $\overline{X}_j, \overline{Z}_j$ that anticommutes with P , we multiply Q_i to it, otherwise they remain unchanged. Finally, we replace Q_i by bP .

We now use the stabilizer formalism to verify quantum teleportation.



We see how the operators change

$$\begin{array}{ccccccc}
S : IXX & & IXX & & IXX & & IXX & & b_2 IZI \\
IZZ & \xrightarrow{\text{CNOT}_{12}} & ZZZ & \xrightarrow{H_1} & XZZ & \xrightarrow{\text{measure } Q_i=XXZ}^{ZII} & b_1 ZII & \xrightarrow{\text{measure } Q_i=IXX}^{ZII} & b_1 ZII \\
N(S)/S : XII & & XXI & & ZXI & & ZXI & & ZIX \\
ZII & & ZII & & XII & & IZZ & & IZZ
\end{array}$$

We then multiply the logical operators by the stabilizers above, obtaining $b_1 IIX$ and $b_2 IIZ$. Therefore by applying operator $Z^{b_1} X^{b_2}$ we recover the state on the third qubit, verifying teleportation as desired.

To summarize, in this section, we discussed in detail how stabilizers transform under Clifford gates and measurements of operators. The principles developed in this section will be used to justify some of the procedures used in this thesis.

2.6 Entanglement Purification Protocol

Entanglement purification protocol (EPP) aims at generating high-fidelity entangled quantum states from a larger set of low-fidelity ones, through local operations and classical communication (LOCC)⁵. This process is essential for practical applications in quantum communication and quantum computing, particularly quantum repeaters [30, 7]. EPP can be broadly classified into one-way and two-way communication protocols. One-way protocols, where classical communication is unidirectional (e.g. only Alice sends information to Bob), are equivalent to quantum error correcting codes. Two-way protocols involve bidirectional communication and iterative processes to enhance the entanglement of the pairs progressively. The two-way EPP also has a connection to error-detecting code. Due to its strong relation with QECC, we include EPP in this section. However, as we do not use one-way protocols in this thesis we will omit the discussion. More details can be found

⁵We assume classical communication is free and with no error.

in the seminal papers by Bennett et al [11]. In the following, we will introduce the very basic two-way EPP to illustrate the idea.

Two-way Protocols

The first EPP was proposed in 1995 [10]. It starts with Alice and Bob sharing imperfect ebits (an *ebit* is an EPR pair $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$). Throughout this thesis, we shall assume the initial noisy EPR pairs are in the Werner state form, so in the Bell basis they can be written as,

$$\rho = F|\Phi^+\rangle\langle\Phi^+| + \left(\frac{1-F}{3}\right) (|\Psi^+\rangle\langle\Psi^+| + |\Psi^-\rangle\langle\Psi^-| + |\Phi^-\rangle\langle\Phi^-|)$$

It was shown in [10, 11] that any two-qubit state can be converted into this form via the appropriate ‘twirl’ operation. Due to the fact $(U \otimes I)|\Phi\rangle = (I \otimes U^T)|\Phi\rangle$, an error on the ebit is one of XI, YI, ZI . The simplest example of EPP is shown in Figure 2.7,

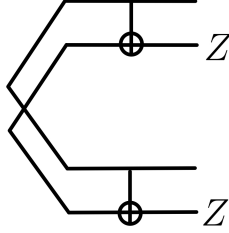


Figure 2.7: A simple EPP. The upper half and lower half are owned by Alice and Bob respectively. Upon measuring the second qubit in the Z -basis, if the outcomes agree it marks a purification.

It can be alternatively thought of as an error-detecting circuit. Assuming we start with two perfect ebits, an XI error occurs in the first ebit, then on Alice’s side, it will be propagated by the CNOT to the second pair by results in Section 2.5 and the Z -basis measurement, the second ebit will now be in the state $|\Psi^+\rangle$. Thus if both parties measure in the Z -basis, the outcomes will disagree, leading to rejection. Following the paradigm, alternative EPP schemes are explored, a more detailed discussion can be found in [55]. In the presence of circuit-level noise, the results ebits will contain errors after EPP. For the specific EPP circuits we use in this work, we will provide detailed descriptions in subsequent sections as they become pertinent to our analysis.

Chapter 3

Fault-tolerance(FT)

In Chapter 2.1 we introduced how to protect qubits from errors using QECC. However, achieving reliable quantum computation requires further considerations. Firstly, in real devices, errors can occur on every gate, not just the data qubits. For example, if we measure the parity of stabilizer $IIIXXXX$ for the Steane code using the method in Figure 2.2 and a IX -error after the first CNOT occurs (so an X -error on the ancilla qubit), this will result in a weight-3-error. This exceeds the Steane code's correction capacity, compromising subsequent computations. Hence, the error correction process must not introduce more errors than it can handle. In fact, the central theme of *fault-tolerance* is to control the amount of errors propagating in the circuit. Secondly, performing quantum operations on the logical qubits in a fault-tolerant manner is essential. This includes executing logical gates, measurements, and state preparations reliably. After these, one naturally wonders, is reliable computation guaranteed if we put the components together?

In this chapter, we provide an overview of previous results that attempt to address some of these questions. We begin by introducing noise models in context, followed by an explanation of what it means to be fault-tolerant (FT) and a discussion of FT gadgets. These concepts combine to form the threshold theorem. Towards the end of the chapter, we explore further aspects of FT in the context of the Steane code and the surface code. More specifically, we compare different fault-tolerant error correction gadgets for the Steane code in terms of pseudo-thresholds and qubit overheads. This will thus justify our choice of Steane EC for later context.

Although the concept of a threshold was first proposed by D. Aharonov and M. Ben-Or [2], a more explicit and easily implementable protocol was not introduced until 2005 [3, 4]. In the subsequent chapters, where we implement the FT protocol, we will primar-

ily follow the later literature, making suitable adjustments to accommodate our specific context.

3.1 Error Model

The exact fault-tolerance criteria will depend on the error model of the simulation circuit. Note that for a quantum circuit, if we represent physical qubits with n -qubit encoded in a $[[n, k, d]]$ QECC and each operation with its corresponding logical operation. The new circuit is called the simulation circuit. Before introducing the error model, we first need to define the notion of a *location*.

A *location* is an indivisible component in a circuit that takes 1-timestep. Broadly speaking, locations can be categorized into the following types: preparation location, gate location, wait location, and measurement location. As assumed before, classical computation is assumed to be error-free. However, to derive our main results, it suffices to list the following locations:

1. preparation of $|0\rangle$
2. preparation of $|+\rangle$
3. measurement of X
4. measurement of Z
5. (local) CNOT gate
6. non-local resource
7. identity gate

The location of type i is denoted Loc_i . Note that in Loc_5 and Loc_6 we have the notion of ‘local’. I will explain what I mean here. Let us recall EPP in Section 2.6, the initial ebits shared between Alice and Bob are considered ‘non-local’ while any operation Alice(Bob) does on her(his) side is ‘local’. Thus Loc_6 can be a nonlocal CNOT (CNOT across Alice and Bob) or an ebit, which will be specified depending on the context. It’s worth distinguishing *error* from *fault*. An *error* occurs when one physical qubit is corrupted while a *fault* occurs when a location goes bad. For example, in the case of Pauli noise, when a CNOT gate is faulty, and XX follows the perfect CNOT. In this case, we say 2 errors occur but there is only one fault. Now we can define the error model.

Definition 3.1.1 (Independent Pauli Noise). In an independent Pauli noise model, each location in the circuit is assumed to fail independently. A faulty location Loc_i can be seen as a perfect Loc_i followed by Pauli noise with strength ϵ_i , except the case of measurement, where is seen as an error followed by perfect measurement.

There are other more realistic error models such as coherent noise, which is characterized by unitaries; qubit leakage, and loss. However, due to the convenience of the stabilizer formalism, independent Pauli noise is tractable with simulation and thus is widely used for analysis of fault-tolerance. Moreover, using randomized compiling techniques, we are able to convert other types of noise into independent Pauli noise [90, 92], validating the experimental relevance of analysis under this model. In our later analysis, we will treat Loc_5 error rate ϵ as the baseline and all others locations being proportional to ϵ with $\epsilon_i = \sigma_i \epsilon, \forall i \in \{1, 2, 3, 4, 6, 7\}$. By default, in this thesis we follow conventions in [51] and have $\sigma_i = \frac{4}{15}$ for $i \in \{1, 2, 3, 4\}$ and $\sigma_7 = \frac{4}{5}$. ϵ_6 is the error rate of the nonlocal resource, in the case of an ebit, it is the infidelity. The analytical and numerical methods in this thesis will be applicable to alternative configurations of σ_i 's, accommodating implementation across various platforms and devices.

The independent Pauli noise falls in the broad category of *circuit-level noise*. When dealing with surface code or qLDPC codes, people sometimes use simplifying noise models to ease numerical and analytical analysis. We list them here for later reference.

- *Code-capacity noise*. Errors only occur on physical qubits of the code and measurements are assumed to be perfect.
- *Phenomenological noise*. On top of code-capacity noise, each syndrome measurement fails with a certain probability. However, error propagation between data qubits and ancilla qubits is neglected.

Results obtained from different noise models should not be compared directly.

3.2 Fault-tolerant Gadgets

Now we set to perform a simulation for the quantum circuit. We first define what we mean by a *gadget*.

Definition 3.2.1 (Gadget). A gadget for a quantum operation is a circuit that executes the corresponding operation on the logical state when no fault occurs. An error-correction

(EC) gadget functions by measuring the stabilizer generators to extract the syndrome and subsequently applying Pauli corrections. When operating without faults, it can correct up to $t = \lfloor \frac{d-1}{2} \rfloor$ errors for a distance d QECC.

3.2.1 FT Criteria

Given different types of locations and their corresponding gadgets in the logical space, we may define what we mean by fault-tolerance.

Definition 3.2.2 (Fault-tolerance criteria [3]). Suppose we encode qubits in a QECC with distance d . Let r denote the number of input errors into a gadget, s denote the number of faults in the gadget ($r = 0$ for preparation gadget), and $t = \lfloor \frac{d-1}{2} \rfloor$. Then a gadget is fault-tolerant if

1. Preparation gadget
When $s \leq t$, a preparation gadget with s faults produces a logical state with at most t errors.
2. Measurement gadget
When $r + s \leq t$, the outcome of a measurement gadget agrees with an ideal measurement. In particular, for a non-destructive measurement, we also need the number of errors on the output data block to be at most t .
3. Gate gadget
When $r + s \leq t$, an output state from a gate gadget has at most t errors in each output block.
4. Error-correction(EC) gadget
When $r + s \leq t$, the output state deviates from the codespace by at most t errors. In particular, if $s = 0$, EC gadget takes any input in the codespace with $r \leq t$ to an output with no errors.

Now we can define a *fault-tolerant scheme*.

Definition 3.2.3 (Fault-tolerant scheme). A fault-tolerant scheme consists of a QECC and simulation gadgets satisfying the above definitions. For the gate gadget, we need FT gadgets for a universal set of gates.

In the following sections, we elaborate on how to obtain the actual gadgets.

3.2.2 FT Logical Gates

Transversal Gates

One of the earlier fault-tolerant logical gate constructions is the idea of *transversal gates*. This is saying, given a QECC, if we hope to simulate gate G on the logical level, we perform $G_L = \bigotimes_{i=1}^n G_i$, where G_i is a single-qubit physical gate acting on the i -th physical qubit. For a two-qubit gate, we have two data blocks of n -qubits each and a transversal gate applies two-qubit gates from the i -th qubit in the first block to the i -th qubit in the second $\forall i \in [n]$. Intuitively, damage caused by one fault in a transversal gate is limited. It will incur at most one error on each data block and won't propagate errors within the same block. A more rigorous argument can justify that it satisfies Criteria 3 in Def 3.2.2. This idea looks simple, however, we still need to check it does what we want on the logical level. In fact, we (unfortunately) have the no-go Eastin-Knill theorem

Theorem 3.2.4 (Eastin-Knill). *No quantum error correcting code can have a continuous symmetry that acts transversely on physical qubits.*

In other words, no universal gate set can be implemented transversely. But in some cases, we do have more convenient gates

Theorem 3.2.5 (Transversal CNOT for CSS Code). *For any CSS code transversal CNOT is the logical CNOT.*

Proof. The two copies of physical qubits are denoted A and B respectively. We denote the X -type and Z -type stabilizers by S_X and S_Z respectively. For $[[n, k, d]]$ code, $\overline{\text{CNOT}}^{\otimes k} = \text{CNOT}^{\otimes n}$. There are two steps we need to verify it is the logical CNOT.

- The operator preserves the stabilizer group.
From results in Section 2.5, we have $S_X^A \otimes I^B \rightarrow S_X^A \otimes S_X^B$, $I^A \otimes S_X^B \rightarrow I^A \otimes S_X^B$, $S_Z^A \otimes I^B \rightarrow S_Z^A \otimes I^B$ and $I^A \otimes S_Z^B \rightarrow S_Z^A \otimes S_Z^B$. Since each block is encoded in the $[[n, k, d]]$ code, the operators after transformation stabilize the states.
- The operator acts as $\overline{\text{CNOT}}^{\otimes k}$.
Let $\overline{X}_j$ and $\overline{Z}_j$ be the j -th logical operators. We have $\overline{X}_j^A \otimes I^B \rightarrow \overline{X}_j^A \otimes \overline{X}_j^B$, $I^A \otimes \overline{X}_j^B \rightarrow I^A \otimes \overline{X}_j^B$, $\overline{Z}_j^A \otimes I^B \rightarrow \overline{Z}_j^A \otimes I^B$ and $I^A \otimes \overline{Z}_j^B \rightarrow \overline{Z}_j^A \otimes \overline{Z}_j^B$. Therefore it performs what we hope for a logical CNOT.

□

Although the above proof is very straightforward, it gives a general recipe for verifying an operator is the correct logical operator. We note that the Steane code admits transversal implementation for $\{H, R_{\pi/4}, \mathbf{CNOT}\}$, the Clifford group. This is also another reason why in platforms with all-to-all connectivity the Steane code is preferable. To achieve FT universal gate set, usually, FT gate-teleportation or magic state distillation is applied [49]. As we do not use this procedure in the later work, we will omit the discussion. People also came up with the code-switching idea. Basically, since the Steane code has a transversal Clifford group and the 15-qubit quantum Reed-Muller code admits transversal T -gate, we hope to use them together: so we use the Steane code to implement Clifford gates and switch to Reed-Muller code when we want to implement T [43, 5].

3.2.3 FT Measurement and Preparation

In this subsection, we will first discuss how to measure the parity of operators fault-tolerantly. Recall in Figure 2.2 we presented a measurement circuit. To see this is non-fault-tolerant, imagine for $t \geq 1$ we have a IX -error after the first CNOT and $r = 0$, although this satisfies $r + s \leq t$, it incurs a $IZZXI$ -error (so 3 errors) in the data block after error propagation. This fails the second criterion. To tackle this, we try to use the nice properties of transversal gates. A solution for FT measurement is to prepare an ancillary "cat state" $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes m} + |1\rangle^{\otimes m})$, m is the weight of the operator being measured. A fault-tolerant version of $XZZXI$ -stabilizer measurement of the $[[5, 1, 3]]$ code is shown in Figure 3.1.

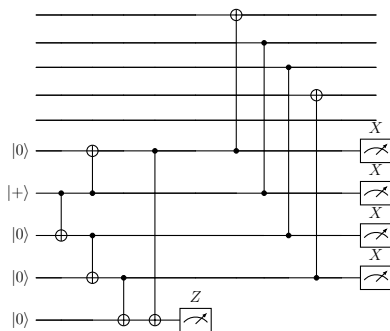


Figure 3.1: Single cat-state measurement of $XZZXI$ for $[[5, 1, 3]]$ code

Since $XZZXI$ has weight-4, we use a 4-qubit cat state. To make the preparation of cat state fault-tolerant, we have a verifying qubit. We postselect and proceed with the cat state only if the ancilla measurement returns 0. From the transversal $\overline{X}$ -measurement result, we can infer the parity. So when the outcome x has even weight, we say the eigenvalue is $+1$, otherwise it's -1 . This is valid by noting the fact that

$$\begin{aligned} H^{\otimes n}(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}) &\propto \sum_{\text{wt}(x)=\text{even}} |x\rangle \\ H^{\otimes n}(|0\rangle^{\otimes n} - |1\rangle^{\otimes n}) &\propto \sum_{\text{wt}(x)=\text{odd}} |x\rangle \end{aligned}$$

However, we observe that a weight-1 Z -error from the cat state can still cause parity change. So to ensure fault-tolerance we need to repeat syndrome measurement. As a rule of thumb, we repeat $2t + 1$ times and take the majority as the result (with EC gadget on the data block before and after each measurement, we will explain constructions for the EC gadget later). In the case of $[[5, 1, 3]]$ code, it suffices to repeat measurements at most 3 times. If the same parity is measured twice consecutively, then we accept the parity and there is no need to perform the third round.

This FT measurement also gives a way to do FT preparation. Say we hope to prepare $|\overline{0}\rangle^{\otimes k}$, the recipe is as follows,

- Initiate the qubits in $|0\rangle^{\otimes n}$ state.
- Measure the stabilizers fault-tolerantly to project the state into stabilizer eigenspace.
- Depending on the code, project onto the $+1$ eigenspace of $\overline{Z}_j \forall j \in [k]$.

In fact, FT state preparation for some codes such as qLDPC is still an underexplored area of research, not to mention efficient preparation. However, in some special cases, we have more efficient encoding. For example, recall that we provided an encoding circuit for the Steane code in Figure 2.6. Adding one ancilla qubit suffices to make it fault-tolerant.

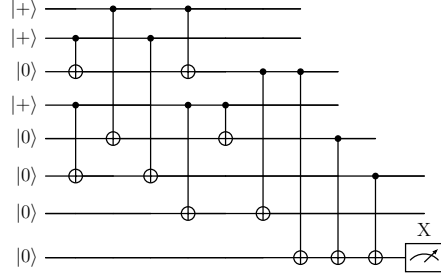


Figure 3.2: Fault-tolerant $|\bar{0}\rangle$ encoding circuit for the $[[7,1,3]]$ code.

Figure 3.2 presents the circuit for FT preparation of $|\bar{0}\rangle$ for the Steane code. Note that the ancilla qubit is used for catching bit-flip errors. Z error verification is not needed because all Z errors can be reduced to weight 0 or 1, thus obeying the FT criteria. $|\bar{+}\rangle$ can be analogously prepared, by reversing the CNOT gates and interchanging $|0\rangle$ and $|+\rangle$.

We note that it is impossible to encode an arbitrary state fault-tolerantly as there is no way to detect error on the initial state if it is unknown. This paves the way for the introduction of the ‘*Interface*’.

3.2.4 FT EC

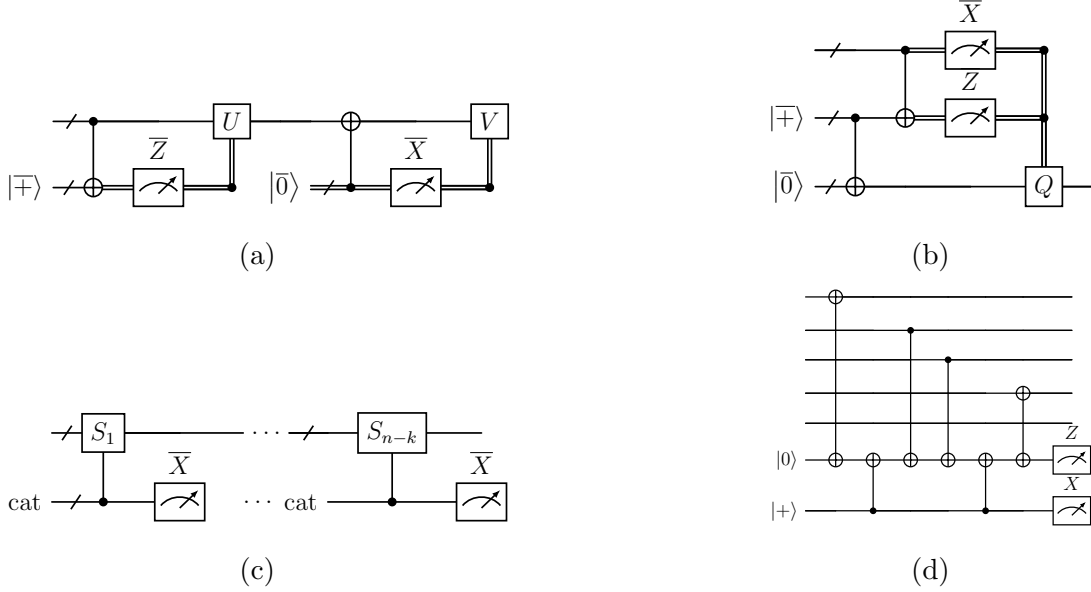


Figure 3.3: (a) Steane EC. U and V are corrections corresponding to measured syndromes. (b) Knill EC using logical teleportation. The applied gate Q depends again on the syndrome measurements. (c) Shor EC, $S_1, \dots, S_{n-k}$ denote the stabilizers, and ‘cat’ is the cat state. The whole set of measurements is to be repeated many times until we get $t + 1$ the same syndrome measurements. Note that in (a)(b)(c) wires with a bar denote blocks of qubits, X, Z -measurements, and CNOT are all transversal (d) An example of Flag EC. This is to measure the syndrome of $XZZXI$ for the $[[5, 1, 3]]$ code. A flag qubit is introduced to detect errors of higher weight caused by the 2nd/3rd/4th gate.

Here we list 4 types of FTEC gadgets.

1. Steane EC [1]. It is applicable only for CSS codes. As in Figure 3.3a, the two ancilla blocks are used to detect X - and Z -errors respectively. The outcomes of the transversal measurements then go through classical post-processing (using the parity-check matrix H_X, H_Z) to deduce the syndrome. From this, we infer the recovery operation U, V .
2. Knill EC [50]. In Figure 3.3b, we use logical qubits here and transversal $\bar{X}, \bar{Z}$ -measurements. The operator Q is used to complete logical teleportation as well as

error-correction on the first data block. To see why this construction is valid, consider a Pauli operator P acting on the data block and the same Pauli acting on the first ancilla block of the logical Bell pair. We focus on $P \in N(S)$ here. We may write

$$(-1)^{b_i} P = i^{c(P_X, P_Z)} (-1)^{b_1} P_X P_Z$$

$c(P_X, P_Z) = 0$ when $[P_X, P_Z] = 0$ and 1 otherwise. After the second transversal CNOT, we would have the following transformations

$$\begin{aligned} (-1)^{b_0} P \otimes I &\rightarrow i^{c(P_X, P_Z)} (-1)^{b_0} P_X P_Z \otimes P_X \\ (-1)^{b_1} I \otimes P &\rightarrow i^{c(P_X, P_Z)} (-1)^{b_1} P_Z \otimes P_X P_Z \end{aligned}$$

and thus multiply together we obtain

$$(-1)^{b_0+b_1} P \otimes P \rightarrow (-1)^{b_0+b_1+c(P_X, P_Z)} P_X \otimes P_Z$$

Therefore if there is any error on the data block, through the logical Bell-state measurement we can detect it and make corrections after teleportation.

3. Shor EC [82]. This is shown in Figure 3.3c and it is analogous to Shor's measurement, also using FT cat states. However, there are some subtle differences. In Shor EC, all stabilizer generators are measured sequentially without repetition to obtain the syndrome. This whole set of measurements is then repeated multiple times. Unlike Shor's measurement, there is no predetermined number of repetitions. Instead, the measurements are repeated until $t + 1$ consecutive syndrome measurements yield identical results.
4. Flag EC [22, 23]. An example of measuring the parity of $XZZXI$ for the $[[5, 1, 3]]$ code is shown in Figure 3.3d. The idea is to avoid faults that cause high-weight/correlated errors. We now describe the procedure for a distance-3 stabilizer code $S = \langle Q_1, \dots, Q_r \rangle$ using 1 flag qubit and the corresponding flagged circuit for measuring Q_i is denoted $C(Q_i)$. The following steps are performed
 - We perform the full set of flagged syndrome measurements and record the syndrome s at each round.
 - If the same syndrome occurs twice consecutively and the flags all give +1 results, we apply the standard syndrome decoding.
 - If in two consecutive rounds the syndromes differ, but the circuit does not flag. We then repeat the syndrome measurement using a set of non-flagged circuits

to obtain the syndrome and apply error correction.

- If during the flagged measurements, a circuit $C(Q_i)$ flags, we repeat the syndrome measurement using non-flagged circuits to obtain syndrome. In this case, the conventional decoding rule does not work. We have to choose the most likely error according to the flag qubits and syndrome (usually for a code we can form distinguishable sets consisting of the faults and their corresponding combinations of syndrome and flag [85]).

When analyzing for fault-tolerance, for distance-3 codes, we only need to check if one fault occurs in the whole EC procedure, the output state has no more than 1 error. Hence if one fault is present in the flagged circuit, the later unflagged circuit will be free of faults. In fact the flag paradigm has been extended to any stabilizer code [24].

As a brief comparison, Shor error correction (EC) seems less feasible due to the potentially large number of measurement repetitions required for larger distance codes. Steane EC is limited to CSS codes. Although Knill EC has a shallower circuit than Shor EC, it requires twice the number of ancilla qubits to perform logical teleportation. Flag EC, on the other hand, significantly conserves ancilla qubits, but its circuit depth may be greater than that of Steane EC. At the end of this chapter, after introducing ‘exRec’, we will provide a numerical comparison of the pseudo-thresholds (we will explain what this is) of these EC methods for the Steane code.

3.3 Threshold Theorem

3.3.1 ExRec Construction

Suppose we have gadgets that individually meet the specified criteria and we aim to use them to simulate an ideal circuit. We observe that unless the physical error rates are arbitrarily small, the errors will accumulate when the gadgets are combined as the ideal circuit gets larger, thereby violating the criteria. Hence to ensure fault-tolerance even when gadgets are put together, one solution is to perform error-correction following every logical operation. This leads to the following definition.

Definition 3.3.1 (*Rec and exRec*). An *extended rectangle* (*exRec*) of a FT simulation circuit is defined as

1. Preparation-exRec

A preparation-exRec consists of a preparation gadget and the EC gadget after it.

2. Measurement-exRec

A measurement-exRec consists of a measurement gadget and the preceding EC gadget.

3. Gate-exRec

A gate-exRec consists of a gate gadget and the EC gadgets immediately before and after it. If omitting the preceding EC, we call it a *gate-rectangle*(*Rec*).

In Figure 3.4 we show the exRecs for various locations. $|\overline{+}\rangle$ -exRec and $\overline{Z}$ -mmt-exRecs will be identical to their counterparts in the figure except for changing $|\overline{0}\rangle$ to $|\overline{+}\rangle$ and $\overline{X}$ to $\overline{Z}$. For the CNOT-exRec we illustrate the difference between *Rec* and *exRec*, the part within the dashed line is a *Rec*.

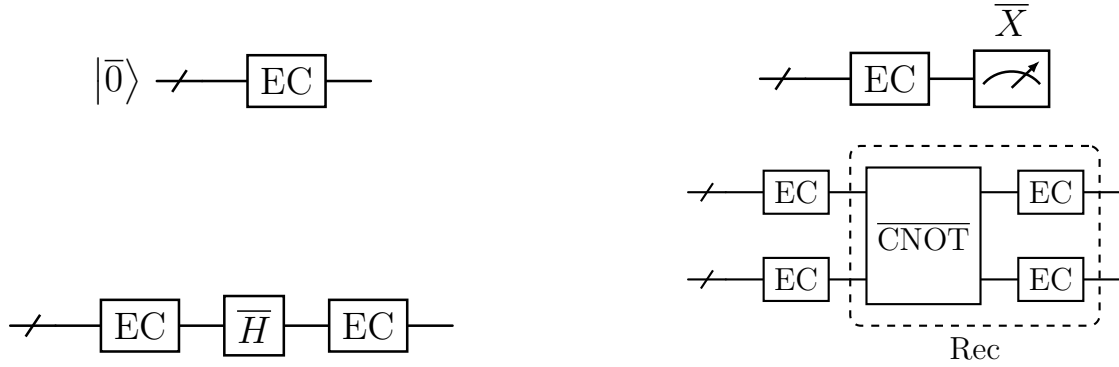


Figure 3.4: Examples of different exRec. Upper-left is the $|\overline{0}\rangle$ -exRec. Upper-right is the $\overline{X}$ -measurement-exRec. Lower-left is a $\overline{H}$ -exRec. Other one-qubit gates are constructed analogously. Lower-right is an example of a two-qubit exRec, the $\overline{\text{CNOT}}$ -exRec. The part in the dashline, without the preceding ECs, is a *Rec*.

To fault-tolerantly simulate a large quantum circuit, it is essential to suppress the error rate of each component. This ensures that, despite the increasing size of the quantum circuit and the presence of noise affecting each component, correct simulation can still be achieved. One way to achieve this is through code concatenation, as in Definition 2.4.5. Based on the exRec construction before, we can concatenate codes and have the following definitions. In this work, we will denote the level of concatenation by k .

Definition 3.3.2 (*Recursive simulation*). Let C_0 be the ideal circuit and let C_l be the level- l fault-tolerant simulation of C_0 . C_l is constructed in the following recursive way:

At level-1, each location is simulated by its corresponding Rec. At level- k , $k \geq 2$, each location is replaced by the $(k - 1)$ -Rec.

Now we still need to justify the above exRec construction combined with recursive simulation works and this will lead to the threshold theorem. To achieve this, we first discuss when an exRec is considered ‘good’. For individual exRec, in [3], the authors had a preliminary definition of ‘goodness/badness’ by counting all combinations of $(t + 1)$ -locations. However, in practice, not all combinations of locations within an exRec result in a logical error at the output. Therefore, we use a modified definition, as presented in the latter half of the same paper.

Definition 3.3.3 (*Malignant set*). A set of locations in an exRec is *benign* if the Rec contained in the exRec has no logical error at the output when any choice of errors occur in these locations. If the set of locations is not benign, they form a malignant set.

The reason why we address the correctness of the Rec will be evident in the following definition.

Definition 3.3.4 (*Badness of exRecs*). For $k = 1$ an exRec is *bad* if it contains it contains faults that form a malignant set; if it is not bad it is good. For $k > 1$, a k -exRec is bad if it contains independent $(k - 1)$ -exRecs at a malignant set of locations. Two bad exRecs are independent if they are non-overlapping or if they overlap and the earlier k -exRec is still bad when the shared k -EC is removed. If it is not bad it is good. For a simulation circuit consisting of exRecs, we call the whole circuit *bad* if the output contains a logical error.

The idea in this ‘independence’ definition is that if there are in total 3 faults in two consecutive 1-exRecs in which one fault is in the overlapping EC, then the overlapping pair of bad 1-exRec is really no worse than a single bad 1-exRec. In [3], it was shown that if all exRecs in the fault-tolerant circuit are good, then this would give a correct simulation of the ideal circuit, i.e. the probability distribution of the final measurement outcomes are the same, thereby validating the fault-tolerance. Based on these definitions we have the following nice result.

Theorem 3.3.5. *Let C be a circuit that begins with state preparation and ends with measurement. Let $\tilde{C}$ be the fault-tolerant simulation of C under independent Pauli noise. Suppose that for a particular fault path γ , the exRecs in γ form a benign set. Then, the output distribution of $\tilde{C}$, is identical to the output distribution of C with ideal gates.*

3.3.2 The Theorem

Before stating the theorem, it is necessary to define precisely what we mean by ‘threshold’. The following definition is adopted from [84].

Definition 3.3.6 (FT threshold). Consider any ideal quantum circuit C . For a series of FT schemes consisting of a family of QECC $[[n(L), k(L), d(L)]]$ parametrized by L and their corresponding FT gadget sets, let C_L denote the L th simulation circuit. For a noise model $\mathcal{N}$ of strength ϵ , the L -th simulation circuit with noise is denoted $C_{L,\mathcal{N}}$. The failure probability p_L of $C_{L,\mathcal{N}}$ is defined as

$$p_L = \sup_{\rho} T(C(\rho), C_{L,\mathcal{N}}(\rho))$$

where $T(\rho, \sigma) = \frac{1}{2} \text{Tr}|\rho - \sigma|$ is the trace distance. The *fault-tolerance threshold* is defined as $\epsilon_{\text{th}}(C)$ such that when $\epsilon < \epsilon_{\text{th}}(C)$,

$$\lim_{L \rightarrow \infty} p_L = 0$$

Based on the above definitions, we now state the theorem formally for independent Pauli noise as it is the main noise considered in this thesis

Theorem 3.3.7 (Threshold theorem for independent Pauli noise). *For any ideal quantum circuit C subject to independent Pauli noise ($\mathcal{P}$) of strength $\epsilon = \max\{\epsilon_i\}_i$, ϵ_i is the error rate for Loc_i , then there exists a family of FT schemes parametrized by L and a FT threshold ϵ_{th} such that when $\epsilon < \epsilon_{\text{th}}$, $\forall p > 0$, $\exists L$ such that $p_L < p$. In particular, if C has K locations, then the L -th simulation circuit has $O(K \text{polylog}(K/p))$ locations.*

The proof of the theorem is constructive based on recursive simulation of the $[[7, 1, 3]]$ -Steane code and exRecs¹. To obtain a FT threshold, due to Theorem 3.3.5, it suffices to ensure that the largest exRec is good, which is typically the CNOT-exRec. Since the $[[7, 1, 3]]$ code can correct one error, an exRec requires at least two faults to be bad. A loose upper bound on the probability $\mathbb{P}(\overline{\text{CNOT}}\text{-exRec bad})$ can be estimated by counting the total number of pairs of locations. Through calculation, we can observe that when $\epsilon < \epsilon_{\text{th}}$, this probability is suppressed double-exponentially. An explicit and improved threshold computation is detailed in Chapter 4. With the logical error rate for each exRec known, we analyze the entire circuit, using $p_L < p$ to determine the required L and thus the total number of locations.

¹Thus in this case the family of QECC used is $[[7^L, 1, 3^L]]$ concatenated Steane code. In the case of surface code, we have $[[L^2 + (L - 1)^2, 1, L]]$

3.3.3 FT for Steane Code

All aspects of fault-tolerant implementation for the Steane code have been discussed above. Here we use the above constructions to compare the different EC gadgets. We will first state the definition of *pseudo-threshold* used in this thesis, adapted from [84].

Definition 3.3.8 (Pseudo-threshold). Suppose we have a fault-tolerant simulation circuit subject to independent Pauli noise of strength $\{\epsilon_i\}_i$, where ϵ_i is the physical error rate of Loc_i . Let $\bar{\epsilon}_G$ denote the output logical error rate, which is a function of $\{\epsilon_i\}_i$. Then

- Let G be an exRec (not measurement or EC). The *pseudo-threshold* of G is

$$\hat{\epsilon}_i = \sup\{\epsilon_G | \bar{\epsilon}_G < \epsilon_G\}$$

- If G is a measurement-exRec, let $\bar{\epsilon}_G$ be the probability the outcome distribution of G differs from the ideal measurement distribution, ϵ_G^* is again the point below which $\bar{\epsilon}_G < \epsilon_G$.
- If G is an EC gadget, ϵ_G^* is defined as the point below which $\bar{\epsilon}_G < \max_i \epsilon_i$.

Now we compare the pseudo-thresholds for 4 different EC gadgets, as in Figure 3.3, for the Steane code. The numerical simulation in Table 3.1 is based on the noise model as described in Section 3.1.

Steane EC	Knill EC	Shor EC	Flag EC
1.7×10^{-3}	6.3×10^{-3}	1.5×10^{-3}	1.9×10^{-2}

Table 3.1: Pseudo-thresholds for 4 EC gadgets

Note that the pseudo-thresholds obtained here slightly differ from those reported in previous literature due to different state preparation procedures. We observe that Flag EC has a slightly better pseudo-threshold compared to Steane EC and Shor EC. As expected, Knill EC has a much higher pseudo-threshold value but requires significantly more ancilla qubits. Given these observations, we will use the Steane EC for the error correction gadget in Chapter 4 due to its reasonable pseudo-threshold and ease of analytical threshold analysis. In contrast, Flag EC requires multiple rounds of measurements based on flag values, making it difficult to count the number of malignant pairs. However, it is fair to conjecture that similar conclusions in later chapters would hold if we used Flag EC instead. This facilitates experimental realization in case of limited space overhead.

3.3.4 FT for Surface Code

For the surface code, fault-tolerant preparation and syndrome measurements can be carried out as previously described. Although single-shot logical state preparation with shallow circuits has been explored [16], we will not discuss it in detail here. For the logical gate, the surface code, being a CSS code, allows for the transverse implementation of a logical CNOT. However, performing a transversal $\overline{\text{CNOT}}$ on two code patches in the same plane undermines the surface code's advantage of using nearest-neighbor interactions for all checks. This motivates alternative schemes for performing logical operations. In this discussion, we focus on one promising scheme, *lattice surgery* [44, 63]. In particular, we will elaborate on how to perform $\overline{\text{CNOT}}$ with lattice surgery.

Essentially lattice surgery is a measurement-based scheme. There are in general two subroutines, lattice-merging and lattice-splitting. We will introduce these first.

Lattice-merging

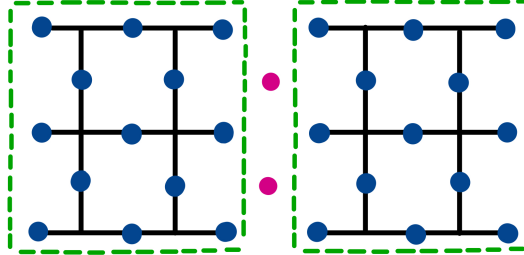


Figure 3.5: Example for rough lattice merging. The pink qubits are initialized in $|0\rangle$.

In Figure 3.5 we show the setup for a lattice merging on rough edges². Initially, two logical surface code patches, each of distance L , are present. An intermediate column of qubits is initialized in the state $|0\rangle$. Let us denote the $\overline{X}$ -operators of the two initial code patches by $\overline{X}^{(1)}$ and $\overline{X}^{(2)}$ respectively. The $\overline{X}$ of the combined patch is denoted $\overline{X}^{(3)}$. Next, we measure the X_v operators across the join, where the product of the outcomes corresponds to the measurement outcome of $\overline{X}^{(1)} \otimes \overline{X}^{(2)}$. This process effectively implements a measurement of $\overline{X}^{(1)} \otimes \overline{X}^{(2)}$. The procedure can be justified by performing stabilizer evolution as discussed

²Note that here the type of edges needs to align.

in Chapter 2. A minimal example can be found in Appendix A in [44]. Subsequently, the entire system is treated as a single patch, and L rounds of error correction are performed as outlined in Chapter 2 to ensure fault-tolerance (analogous to *Gate-EC* as in the exRec). By symmetry, we can prepare the intermediate qubits in $|+\rangle$ and merge the smooth edges to obtain measurement of $\bar{Z}^{(1)} \otimes \bar{Z}^{(2)}$.

Lattice-splitting

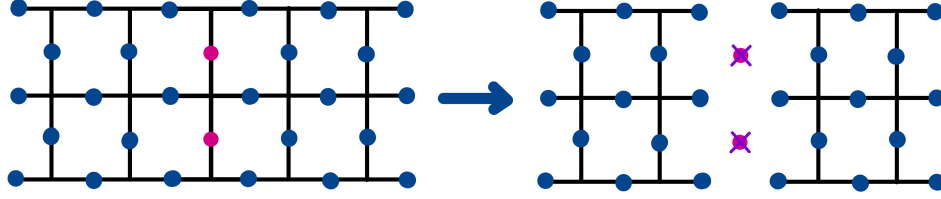


Figure 3.6: Example of lattice-splitting

In Figure 3.6 we illustrate an example of lattice-splitting for rough edges. In this case, we perform Z measurements on the intermediate column of qubits, followed by L rounds of error correction.

We now determine its action on the state. To keep consistency, we denote the X, Z logical operator of the original large patch by $\bar{X}^{(3)}, \bar{Z}^{(3)}$ and of the two separate patches by $\bar{X}^{(1)}, \bar{Z}^{(1)}$ and $\bar{X}^{(2)}, \bar{Z}^{(2)}$ respectively. After measurements, if all outcomes are $+1$, no corrections are needed. If any is -1 , matching pairs of Z -plaquette stabilizers in the -1 eigenstate appear on the boundaries. We apply a chain of physical X operations to the lower boundary. Using the top boundary would differ by a local $\bar{X}^{(i)}$. As long as the same boundary is chosen for both, the correction strategies, differing only by the stabilizer operation $\bar{X}^{(1)} \otimes \bar{X}^{(2)}$, are equivalent. Essentially, a rough split preserves the logical X operator but splits the logical Z , causing the effect

$$a |\bar{+}\rangle^{(3)} + b |\bar{-}\rangle^{(3)} \rightarrow a |\bar{+}\rangle^{(1)} |\bar{+}\rangle^{(2)} + b |\bar{-}\rangle^{(1)} |\bar{-}\rangle^{(2)}$$

For a smooth split, we exchange X and Z and arrive at the opposite effect.

CNOT

Now we can implement CNOT with a ‘measurement-only’ circuit as in Figure 3.7.

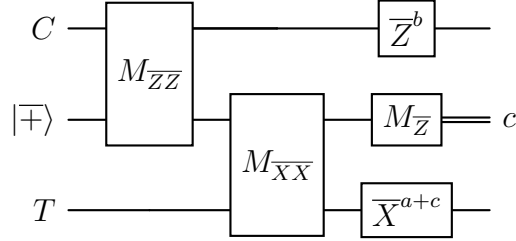


Figure 3.7: Measurement-based CNOT. a and b are the measurement outcome of $M_{\overline{Z}\overline{Z}}$ and $M_{\overline{X}\overline{X}}$ respectively.

To perform a CNOT operation between two code patches, each encodes a logical state, we follow the steps outlined in the circuit above:

1. Initiate an ancillary code patch in $|\overline{+}\rangle$. Let us denote this patch by P
2. Smooth-merge surfaces C and P . This performs $M_{\overline{Z}\overline{Z}}$ measurement with outcome a .
3. Smooth-split the merged surface.
4. Rough-merge surfaces P and T . This performs $M_{\overline{X}\overline{X}}$ measurement with outcome b .
5. Logical $M_{\overline{Z}}$ measurement on P , giving outcome c .
6. Perform corrections according to a, b, c .

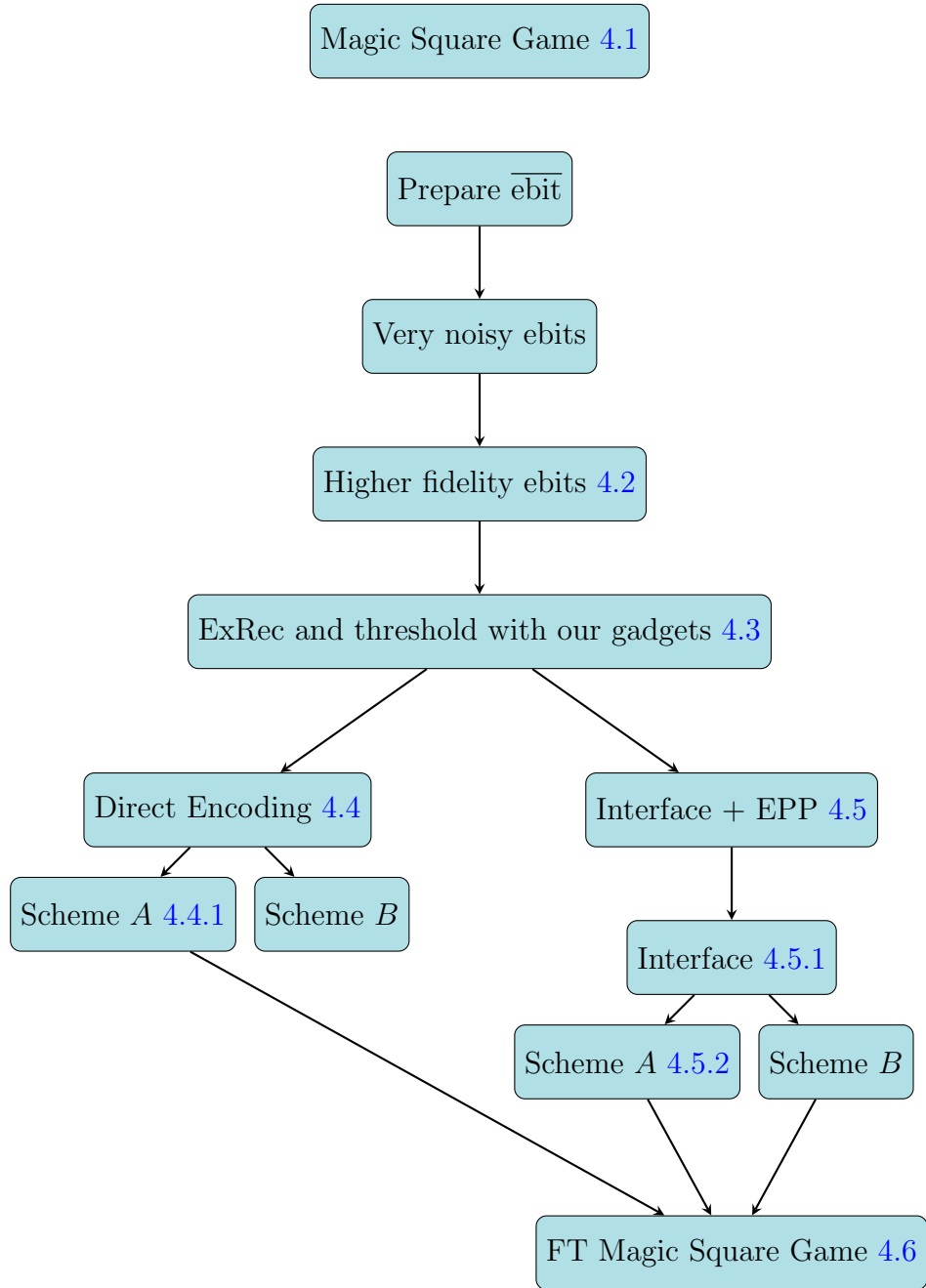
We note after each operation, L rounds of error correction are performed to ensure FT, in the spirit of the *EC-Gate-EC* in previous sections. With lattice surgery, it is viable to implement a logical universal gate set. The T gate can be realized via measurement-based gate-teleportation [39]. In fact, if we alternatively treat the Steane code as color code, lattice surgery can also be used to perform logical gates [58]. However, experimentally this is shown to have worse fidelity than transversal implementation [81]. Therefore for the Steane code, we will stick to transversal gates. For the surface code, we use lattice surgery. The advantage will be evident when we discuss the interface for surface code in Chapter 5.

Chapter 4

FT Magic Square Game with Concatenated Steane Code

In this chapter, we tackle the problem of *Fault-tolerant Magic Square Game*. We will start by introducing the *Magic Square Game* and the quantum strategy that guarantees success given no errors in the circuit. We will then discuss and compare two main methods for preparing $\overline{\text{ebit}}^{(k)}$ assuming that the two parties are provided with noisy ebits as their initial resources¹. The FT magic square game serves as an application resulting from these preparations. The first approach, termed *Direct Encoding*, involves simulating ideal gates through the exRec construction, following procedures outlined in [3]. Next, we introduce a scheme named *Interface+EPP*. This method commences with Alice and Bob sharing ebits, encoding into the logical space via an ‘interface’ followed by EPP, which necessitates classical communication to attain $\overline{\text{ebit}}^{(k)}$. We should note that the different strategies are only applied to prepare $\overline{\text{ebit}}^{(k)}$, before the actual game, thus there is no violation of the game setting. Moreover, we conduct an in-depth analysis and comparison of the logical error rates resulting from the application of these two methods. One key consideration is determining the expected number of noisy ebits needed to achieve a certain success rate when playing FT magic square game, which we will analyze at the end of this chapter. To facilitate navigation through this extensive chapter, we provide the following flowchart.

¹From now on, we denote a logical ebit by $\overline{\text{ebit}}$. An $\overline{\text{ebit}}$ encoded in $[[7^k, 1, 3^k]]$ concatenated Steane code is denoted $\overline{\text{ebit}}^{(k)}$



4.1 Magic Square Game

The Mermin-Peres Magic Square game is one of the simplest non-local games in which a referee randomly assigns a row and a column index $a, b \in \{0, 1, 2\}$ to two parties Alice and Bob. They then each replied with three answers $[A_0^a, A_1^a, A_2^a]$ and $[B_0^b, B_1^b, B_2^b]$. The winning condition is $\prod_{i=0}^2 A_i^a = +1$ for all a and $\prod_{i=0}^2 B_i^b = -1$ for all b and most importantly we require $A_b^a = B_a^b$, i.e. the overlapping element of their answers must be the same. The best classical strategy succeeds with probability $8/9$, but there is a perfect quantum strategy, assuming all operations are faultless. The strategy is as follows: A and B share the state $\frac{1}{2}(|00\rangle + |11\rangle)_{A_1 B_1}(|00\rangle + |11\rangle)_{A_2 B_2}$ before the game starts. They then measure in the set of basis given in Table 4.1 corresponding to the indices they are assigned. For instance, if the referee assigns Alice 1 and Bob 2, when the game starts Alice will measure her qubits in ZI, IX, ZX -basis sequentially and Bob will measure in XZ, ZX, YY -basis. Following this procedure, their measurement outcomes will satisfy the criteria.

	0	1	2
0	$I \otimes Z$	$Z \otimes I$	$Z \otimes Z$
1	$X \otimes I$	$I \otimes X$	$X \otimes X$
2	$X \otimes Z$	$Z \otimes X$	$Y \otimes Y$

Table 4.1: Measurement basis for Alice and Bob when they are assigned different indices. Alice holds the row index while Bob holds the column index.

4.2 Preparation of High-fidelity Physical EPRs

In practice, an ebit shared between two distant parties will have significantly higher infidelity compared to the local physical error rates, e.g. ebits generated with photons, and the attenuation will worsen in proportion to distance. In this thesis, for convenience, we will assume the infidelity of the initial EPR pair to be $3q \approx 10\%$ [46], where q is the probability of one of the other Bell states². For higher initial infidelity we can perform

²4 maximally entangled two-qubit Bell states are: $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, $|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$, $|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$, $|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$.

some rounds of EPP to reduce to the desired error rate. The physical error rate of local operations is assumed to be below the FT threshold³ (which is usually much lower than the initial infidelity) to ensure fault-tolerance. Due to this large discrepancy and by the threshold theorem in Chapter 3, it's necessary to first bring down the infidelity to a level comparable to local error rates before we prepare $\overline{\text{ebit}}^{(k)}$ so that we can treat them as ‘the same type of error’. To accomplish this, we will perform physical EPPs at the start. We refer to schemes explored by Krastanov et al [55]. For our purpose, as we hope to minimize non-local resources, we choose ‘EXPEDIENT’ that purifies 1 ebit from 5, as shown below (on one side)

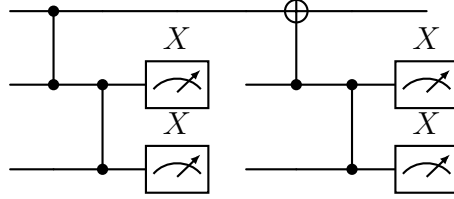


Figure 4.1: EXPEDIENT purification circuit (one side). Alice and Bob share 5 ebits initially. They then locally perform the circuit and measurements above. Upon comparing the 4 measurement results, if all of them agree, they keep the first ebit; otherwise discard.

Given the local physical error rate ϵ , we have that the infidelity after one round of purification being

$$I(q, \epsilon) = \frac{1}{2}\epsilon + \frac{63}{16}\epsilon^2 + \frac{49}{4}q\epsilon + 6q^2 \\ + \frac{879}{64}\epsilon^3 + \frac{107}{2}q\epsilon^2 + \frac{345}{4}q^2\epsilon + 48q^3$$

obtained from the software. If we perform another round of EPP, the infidelity will be $I(I(q, \epsilon)/3, \epsilon)$ etc. In the next section, we will compute a theoretical threshold value. With this information, we can determine the necessary number of rounds and the success probability of EPP, facilitating subsequent resource comparisons.

³For our specific case we will calculate the threshold later

4.3 ExRec and Threshold

In this chapter, we consider the concatenated $[7^k, 1, 3^k]$ Steane code. The exRec constructions introduced in Chapter 3 will be used. Throughout this chapter, the EC used will be the Steane EC. The FT preparation of $|\bar{0}\rangle$ is as in Figure 3.2. At $k = 1$, since $[7, 1, 3]$ is a CSS code, transversal CNOT is $\overline{\text{CNOT}}$. Besides, since the Steane code is of distance 3, it's capable of correcting one error. So at least two faults are needed to cause a logical error. We will thus confine Definition 3.3.3 to *malignant pairs* of locations. The error model used is detailed in Section 3.1 Chapter 3. In our simulations, a pair of locations in an exRec is identified as malignant when noise is introduced into these specific locations, while others remain fault-free, and this leads to a logical error for the Rec. The procedure of counting needs more prudent treatment, as detailed in [3]. The Steane code has the nice property that a faultless EC will take any input to the codespace and an EC with one fault will take any input to a state that deviates at most weight-one error from the codespace. So essentially we are testing the correctness of the Rec given the input to the Rec is an operator E_i of weight at most 1. This is sufficient for fault-tolerance since the whole circuit can be seen as a string of Recs followed by one another except the preparation exRec, which was justified to be fault-tolerant. Given previous definitions of malignant set and badness, we can treat the exRecs as independent when generalizing to a higher level of concatenation.

We shall proceed by analyzing the bounds for the logical error rate of the CNOT-exRec and other exRecs at $k = 1$ and then generalize to higher-level encoding according to their interdependence. It will be clear later why this is necessary instead of generalizing each exRec independently. For the CNOT-exRec, there are $\gamma_5 = 279$ locations. For the upper bound, if we start with second-order terms, we may enumerate the number of malignant pairs α . For a tighter upper bound and also a rigorous lower bound, not only must we consider the probability of malignant pairs, but we also need to take into account the type of faults (X/Z) causing the logical error. For example, if a Loc_1 and a Loc_5 form a malignant pair, then $IZ/ZI/ZZ$ error following Loc_5 will not cause a logical error. For the lower bound, we observe that if two locations of a malignant pair are faulty and every other location is fault-free, then this must result in a logical error.

The joint probability $\varepsilon_{5,\text{joint}}^{(k)}$ of failure for the CNOT k -exRec given acceptance of all ancillas (from the circuit in Appendix 2.1, every $|\bar{0}\rangle$ or $|\bar{+}\rangle$ state we use in EC gadget has

a verifying ancilla to ensure fault-tolerance) can be generally expressed as

$$\begin{aligned} & \sum_{j \leq i=1}^5 \alpha_5(i, j) \epsilon_i^{(k-1)} \epsilon_j^{(k-1)} (1 - \max\{\epsilon_i^{(k-1)}, \epsilon_j^{(k-1)}\})^{\gamma_5-2} \\ & \leq \varepsilon_{5, \text{joint}}^{(k)} \leq \sum_{j \leq i=1}^5 \alpha_5(i, j) \epsilon_i^{(k-1)} \epsilon_j^{(k-1)} + \beta(\epsilon_{\max}^{(k-1)})^3 \end{aligned}$$

where α_5 is the malignant pair matrix(MPM) with $\alpha_5(i, j)$ denote the number of pairs of locations of type i and j where faults can cause logical errors, given that all ancillas in the exRec are accepted. ϵ_6 , the error rate of CNOT gates across two data blocks is addressed separately. The rationale for this approach will become evident in the subsequent section. Note that each pair of locations is run 10^3 number of times and the average is taken.

$$\alpha_5 = \begin{pmatrix} 43 & & & & & & \\ 0 & 43 & & & & & \\ 0. & 176.0 & 168.0 & & & & \\ 176.0 & 0 & 0 & 168.0 & & & \\ 299.2 & 298.7 & 570.0 & 572.8 & 793.4 & & \\ 49.2 & 50.0 & 100.0 & 100.5 & 251.0 & 12.8 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

There are some details worth commenting,

1. Notice that in α_5 , $\alpha_5(1, 1) = \alpha_5(2, 2)$ and $\alpha_5(3, 3) = \alpha_5(4, 4)$. This is no coincidence. These two pairs of locations in this exRec are in some sense "symmetrical" to each other.
2. We observe that $\alpha_5(2, 1) = 0$, this is because if the pair consists of a Loc_1 and a Loc_2 . Loc_1 can only induce X -error and Loc_2 can only induce Z -error. The EC we use is capable of correcting one X and one Z error. Therefore they won't add up to a logical error. The same applies to other 0 entries.
3. For the CNOT-exRec we treat Loc_5 and Loc_6 separately. If later we use CNOT-exRecs in recursive simulation or local operations, we can simply set $\epsilon = \epsilon_6$.

If we further consider $\mathcal{O}(\epsilon^3)$ terms, if there are three faults in the circuit, this could potentially result in a logical error. To obtain a tighter upper bound, we break down the various types of locations. There are $n_5 = [32, 32, 32, 32, 144, 7, 0]$ locations of each type and we

hope to exclude the cases where a malignant pair is among the three locations since they are accounted for in the prior term. We have the following Lemma for an upper bound on the constant of $\mathcal{O}(\epsilon^3)$,

Lemma 4.3.1. *Given malignant pair matrix $\alpha \in \mathbb{R}^{n \times n}$, $\mathcal{O}(\epsilon^3)$ that potentially causes a logical error can be upper bounded by $F\epsilon^3$ where*

$$F = F(\mathbf{n}, \sigma, \alpha) = \sum_{s=1}^n f_{ss} + \sum_{\{s,t\} \in \binom{[n]}{2}} f_{st}$$

for $f_{ss} = \binom{n_s}{3} \sigma_s^3 - \frac{1}{3}(n_s - 2) \sigma_s \alpha_{ss}$ and $f_{st} = \binom{n_s}{2} \cdot n_t \cdot \sigma_s^2 \sigma_t + \binom{n_t}{2} \cdot n_s \cdot \sigma_s \sigma_t^2 - \frac{1}{3}(n_s - 1) \sigma_s \alpha_{st} - \frac{1}{3}(n_t - 1) \sigma_t \alpha_{st}$.

The proof of the lemma is left in Appendix A.1. This result will be used throughout the paper where similar scenarios occur. We will further mention the specification of one component, the EC gadget. For the EC gadget, for $k = 1$, there are 186.4 malignant pairs and for third-order terms, applying the theorem gives an upper bound of $F = 8847.5$.

Now by Bayes' rule, $\varepsilon_5^{(k)} = \frac{\varepsilon_{5,\text{joint}}^{(k)}}{\mathbb{P}(\text{ancillas accepted})}$. Since we have 8 ancilla qubits, if we let $\mathbb{P}_{|\bar{0}\rangle, \text{accept}}^{(k)}$ denote the probability of a level- k $|\bar{0}\rangle$ or $|\bar{+}\rangle$ being accepted, we have

$$\varepsilon_5^{(k)} = \left(\mathbb{P}_{|\bar{0}\rangle, \text{accept}}^{(k)} \right)^{-8} \varepsilon_{5,\text{joint}}^{(k)}$$

To establish an upper bound on $\varepsilon_5^{(k)}$, we need a lower bound on $\mathbb{P}_{|\bar{0}\rangle, \text{accept}}^{(k)}$. Through simulation, we enumerate on average 10.8 fault locations that will cause rejection of the data block. So

$$1 - C\varepsilon_5^{(k-1)} \leq \mathbb{P}_{|\bar{0}\rangle, \text{accept}}^{(k)} \leq 1$$

where $C = 10.8$. Putting all the above together, at $k = 1$, we will have the bounds for $\varepsilon_5^{(1)}$,

$$\begin{aligned}
& 1312.6\epsilon^2(1-\epsilon)^{\gamma_5-9}(1-\epsilon_6)^7 \\
& + 330.9\sigma_6\epsilon^2(1-\epsilon)^{\gamma_5-8}(1-\epsilon_6)^6 + 12.8\sigma_6^2\epsilon^2(1-\epsilon)^{\gamma_5-7}(1-\epsilon_6)^5 \\
& \leq \varepsilon_5^{(1)} \\
& \leq (1-10.8\epsilon)^{-8} \cdot \dots \\
& \dots \left(1312.6\epsilon^2 + (12.8\sigma_6 + 330.9)\sigma_6\epsilon^2 + 734691.4\epsilon^3 \right. \\
& \quad \left. + (57999.0\sigma_6 + 3079.0\sigma_6^2 + 13.6\sigma_6^3)\epsilon^3 \right)
\end{aligned}$$

Invoking Prop A.2.0.1 we will obtain $A_5^{(1)}$ and $B_5^{(1)}$ such that

$$A_5^{(1)}\epsilon^2 \leq \varepsilon_5^{(1)} \leq (1-C\epsilon)^{-8}B_5^{(1)}\epsilon^2$$

which holds for $\epsilon \leq 1/B_5^{(1)}$. In particular, for $\sigma_6 = 1$, e.g. in local CNOT-exRec, we have $A_5^{(1)} = 1431.4$, $B_5^{(1)} = 2038.9$. In this case, we can further simplify the upper bound since $\varepsilon_5^{(1)} \leq (1-C/B_5^{(1)})^{-8}B_5^{(1)}\epsilon^2 \leq D_5^{(1)}\epsilon^2$ where $D_5^{(1)} = 2127.4$. We denote the lower and upper bounds by $\mu_5^{(1)}$ and $\nu_5^{(1)}$ respectively. In Appendix A.4, we compare the above level-1 bounds with the actual logical error rates obtained from simulation and the bound acquired through the procedure in [3]. Note that when computing the numerics above, we take into account the factors σ_i as introduced in Chapter 3 Section 3.1 except for σ_6 which we keep as a variable. We will obtain level-1 bounds for other exRecs and the corresponding MPM will be left in Appendix D. Thus for $\varepsilon_{1,\text{joint}}^{(1)}$ we have,

$$\begin{aligned}
255.3\epsilon^2(1-\epsilon)^{\gamma_1-2} & \leq \varepsilon_{1,\text{joint}}^{(1)} \leq 255.3\epsilon^2 + 21381.0\epsilon^3 \\
244.6\epsilon^2 & \leq \varepsilon_{1,\text{joint}}^{(1)} \leq 321.8\epsilon^2
\end{aligned}$$

The bounds are obtained by applying Prop A.2.0.1. However, for the lower bound, since the CNOT-exRec is the largest exRec in the simulating circuit, $\epsilon \leq 1/D_5^{(1)}$, so the coefficient is obtained by $A - B/D_5^{(1)}$ instead. Since there are 3 ancilla verifications here, we obtain,

$$\begin{aligned}
244.6\epsilon^2 & \leq \varepsilon_1^{(1)} \leq (1-C\epsilon)^{-3} \cdot 321.8\epsilon^2 \\
A_1^{(1)}\epsilon^2 & \leq \varepsilon_1^{(1)} \leq D_1^{(1)}\epsilon^2
\end{aligned}$$

where $A_1^{(1)} = 244.6, D_1^{(1)} = 327.0$. Again we denote the lower and upper bounds by $\mu_1^{(1)}$ and $\nu_1^{(1)}$. For $\varepsilon_2^{(1)}$, note that $|\bar{+}\rangle$ -prep-exRec can be simply obtained from $|\bar{0}\rangle$ -prep-exRec by swapping $|+\rangle$ and $|0\rangle$, meas- X and meas- Z , therefore it will have the same logical error rate, except the distribution of X - and Z - logical error rates are now swapped. Similarly for the destructive measurements Loc_4 and Loc_5 , we have

$$\begin{aligned} 16.6\epsilon^2 &\leq \varepsilon_4^{(1)} \leq (1 - C\epsilon)^{-2} \cdot 78.8\epsilon^2 \\ 16.6\epsilon^2 &\leq \varepsilon_4^{(1)} \leq 79.6\epsilon^2 \end{aligned}$$

where the lower and upper bounds are denoted by $\mu_4^{(1)}$ and $\nu_4^{(1)}$ respectively.

Now we hope to generalize to higher level ks . Since we pursue tight upper and lower bounds here, we incorporate the error probabilities of different exRecs in the computation. This also adds extra complexity when $k > 1$ because, at a higher level, there is no guarantee that $\varepsilon_i^{(k)} = \sigma_i \varepsilon_5^{(k)}$ still holds. We provide a workaround for this issue. Let $A_t^{(k)}$ and $D_t^{(k)}$ denote the lower and upper bounds of second-order(in $\varepsilon_5^{(k)}$) coefficients for Loc_t on k^{th} -level encoding, $\sigma_{L,t}^{(k)}$ and $\sigma_{U,t}^{(k)}$ denote the lower and upper bound for $\sigma_i^{(k)}$ where $\varepsilon_i^{(k)} = \sigma_i^{(k)} \varepsilon_5^{(k)}$. From the bounds and expressions above, we can generalize to the following system of equations.

$$\begin{aligned} B_t^{(k+1)} &= G_2 \left(\sum_{1 \leq i, j \leq 5} \alpha_t(i, j) \sigma_{U,i}^{(k)} \sigma_{U,j}^{(k)}, \bar{F}(\mathbf{n}_t, \underline{\sigma}_L^{(k)}, \underline{\sigma}_U^{(k)}, \alpha_t) \right) \\ D_t^{(k+1)} &= B_t^{(k+1)} \left(1 - C\epsilon_0^{(k)} \right)^{-s} \\ \epsilon_0^{(k+1)} &= \min_{i \leq (k+1)} 1/D_5^{(i)} \\ A_t^{(k+1)} &= \left(\sum_{1 \leq i, j \leq 5} \alpha_t(i, j) \sigma_{L,i}^{(k)} \sigma_{L,j}^{(k)} \right) \left(1 - n_t \epsilon_0^{(k)} \right) \\ \sigma_{L,t}^{(k+1)} &= \frac{D_t^{(k)}}{A_5^{(k)}} \\ \sigma_{U,t}^{(k+1)} &= \frac{A_t^{(k)}}{D_5^{(k)}} \end{aligned}$$

$1 \leq t \leq 7, t \neq 6$ represents the fault location index, $t \neq 6$ since we hope to find the local threshold here; $\bar{F}$ is almost identical to F except that in the sum, every positive term is replaced by $\sigma_{U,t}^{(k)}$ and negative term is replaced by $\sigma_{L,t}^{(k)}$ to ensure $D_t^{(k+1)}$ is indeed an upper bound. G_2 is the upper bound function in Prop A.2.0.1; $s = 3$ when $t = 1, 2$, $s = 2$ when $t = 3, 4$, $s = 8$ when $t = 5$ and $s = 4$ when $t = 7$; α_t is the malignant pair matrix for Loc_t ; $\epsilon_0^{(k)}$ represents the threshold value and is a non-increasing function. The equation is established by observing that the Loc_5 -exRec is the largest exRecs in the simulation circuit. n_t is the number of $(k-1)$ -gadgets in a k - t -exRec.

The above set of equations effectively forms a discrete-variable dynamical system and fortunately, we have a good initial point $\sigma_{U,t}^{(0)} = \sigma_{L,t}^{(0)} = 4/15, \forall 1 \leq i \leq 4$, $\sigma_{U,5}^{(0)} = \sigma_{L,5}^{(0)} = 1$, $\epsilon_0^{(0)} = 1/D_5^{(1)}$ and $\sigma_{U,7}^{(0)} = \sigma_{L,7}^{(0)} = 4/5$. We then apply the fixed-point iteration method to find the non-trivial fixed-point ($\sigma_5 = 1$ will not change through iteration).

$$\begin{aligned}\sigma_{U,1}^* &= \sigma_{U,2}^* = 0.302, \\ \sigma_{U,3}^* &= \sigma_{U,4}^* = 0.0643, \\ \sigma_{U,7}^* &= 0.540, \\ \sigma_{L,1}^* &= \sigma_{L,2}^* = 9.77 \times 10^{-2}, \\ \sigma_{L,3}^* &= \sigma_{L,4}^* = 0, \\ \sigma_{L,7}^* &= 0.166.\end{aligned}$$

And the corresponding coefficients are

$$\begin{aligned}A_1^* &= 178.6, A_4^* = 0, A_5^* = 979.7, A_7^* = 302.8, \\ D_1^* &= 295.8, D_4^* = 63.0, D_5^* = 1827.1, D_7^* = 529.4, \\ \epsilon_0^* &= 1/2129.4.\end{aligned}$$

Thus $\epsilon_0 = 1/2129.4 \approx 4.70 \times 10^{-4}$ will be taken as the threshold value. The fixed point can be checked by plugging into the original equations. Through analyzing the Jacobian at the initial point, we found that the largest eigenvalue is less than 1, which indicates this is a stable fixed point. The numerical simulation of the behaviour of convergence is left in Appendix A.3. Having obtained $D_t^{(k)}$'s, we can establish the following recursive relation for $\varepsilon_5^{(k)}$

$$A_5^{(k)} \left(\varepsilon_5^{(k-1)} \right)^2 \leq \varepsilon_5^{(k)} \leq D_5^{(k)} \left(\varepsilon_5^{(k-1)} \right)^2.$$

And thus we have the following theorem,

Theorem 4.3.2. *Suppose that independent stochastic noise occurs with probability ϵ at each location in a noisy quantum circuit. Then the probability $\epsilon^{(k)}$ of the largest k -exRec being bad satisfies the following bounds*

$$\begin{aligned} & \frac{1}{A_5^{(k)}} \left(A_5^{(k)} \prod_{i=1}^{k-1} \left(A_5^{(i)} \right)^{2^{k-1-i}} \epsilon^{2^{k-1}} \right)^2 \\ & \leq \epsilon_5^{(k)} \leq \frac{1}{D_5^{(k)}} \left(D_5^{(k)} \prod_{i=1}^{k-1} \left(D_5^{(i)} \right)^{2^{k-1-i}} \epsilon^{2^{k-1}} \right)^2 \end{aligned}$$

for $\epsilon \leq \epsilon_0 \approx 4.70 \times 10^{-4}$. $\lim_{k \rightarrow \infty} A_5^{(k)} = A_5^* = 979.7$ and $\lim_{k \rightarrow \infty} D_5^{(k)} = D_5^* = 1827.1$. Both bounds approach 0 as $k \rightarrow \infty$.

For each k , we denote the lower and upper bound by $\mu_5^{(k)}$ and $\nu_5^{(k)}$ respectively. Similarly we can obtain $\mu_i^{(k)}$ and $\nu_i^{(k)}$ for other locations given the corresponding $A_i^{(k)}$ and $D_i^{(k)}$. Up to now, we have obtained bounds on the logical error rates of the exRecs and a threshold on fault-tolerant computation locally.

4.4 Direct Encoding

In this section, we introduce two alternative (commonly conceived) ways of preparing $\overline{\text{ebit}}^{(k)}$. We will outline the implementations of each method and subsequently provide an analysis of the logical error rate for $\overline{\text{ebit}}^{(k)}$ prepared using the first method.

4.4.1 Schemes

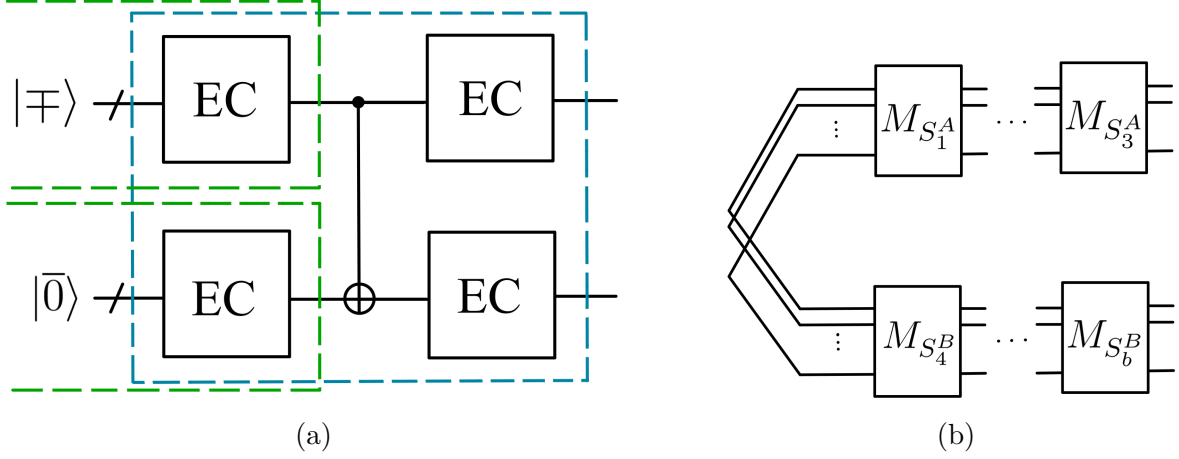


Figure 4.2: (a) Direct Encoding Scheme A. $|\overline{+}\rangle$ and $|\overline{0}\rangle$ are prepared by Alice and Bob respectively. Transversal CNOTs are then applied to create $\overline{\text{ebit}}$. (b) Direct Encoding Scheme B. Alice and Bob start with high-fidelity ebits, Alice measures the X-type stabilizers and Bob measures the Z-type stabilizers. The measurements are repeated for FT and the outcomes are used for correction to ensure the resultant joint logical state is $\overline{\text{ebit}}$.

Scheme A

In Scheme A, Alice locally prepares a $|\overline{+}\rangle$ and Bob prepares $|\overline{0}\rangle$. They then jointly perform a logical-CNOT, giving $\overline{\text{ebit}}$. The procedure is illustrated in Figure 4.2a. However, there is an extra layer of complication here. Based on our assumption that the only entangling resources shared by two parties are noisy ebits, i.e. Alice and Bob cannot directly apply CNOT between their qubits. To tackle this issue, we observe that by utilizing a single ebit and gate teleportation, we can implement a CNOT gate between two independent qubits. Such an efficient circuit is shown in Figure 4.3(also experimentally demonstrated in [25])

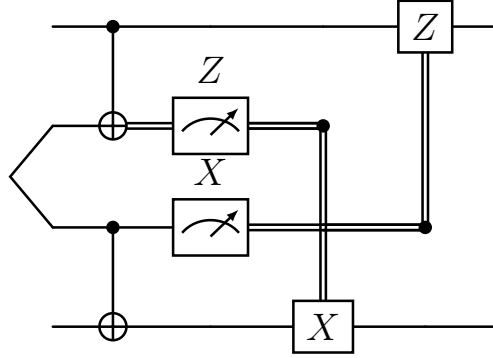


Figure 4.3: The circuit for performing a CNOT with one ebit and local operations[97]. The first and last qubits are the independent qubits we hope to perform CNOT on with the first being control and the last being target. In the middle two sides share an ebit.

Scheme B

The procedure for Scheme *B* is illustrated in Figure 4.2b. At $k = 1$, Alice and Bob initially share 7 high-fidelity physical ebits. Alice then measures the X -type stabilizers of the Steane code S_1, S_2, S_3 respectively. For fault tolerance, each measurement needs to be repeated three times (or twice if the first two consecutive measurements agree). Simultaneously, Bob measures the Z -type stabilizers S_4, S_5 , and S_6 . The reason why Scheme *B* produces $\overline{\text{ebit}}^{(1)}$ is not immediately apparent. However, using the techniques introduced in Chapter 2, Section 2.5, specifically the evolution of stabilizers, a simple justification can be obtained. This explanation is provided in Appendix B.1. For higher level concatenations, Alice and Bob will be sharing 7^k physical ebits. The following measurements are the same, except the gates are replaced with exRecs with recursive simulation.

Scheme *B* has a significantly more complex circuit compared to Scheme *A*, so we will focus our analysis solely on Scheme *A* and omit further consideration of Scheme *B*.

4.4.2 Logical error rate of Scheme *A*

The error rate of the nonlocal CNOT circuit in Figure 4.3 will be ϵ_6 , which depends on q , the infidelity of ebit and ϵ . To guarantee that the above construction works, we require $\epsilon_6 \leq \epsilon_0$. Concerning the fidelity of the EPR pair, if we start with an initial value of

$3q = 10\%$, we observe $I(q, 0) = 8.44 \times 10^{-3}$ and $I(I(q, 0)/3, 0) = 4.86 \times 10^{-5} \leq \epsilon_0$. Since $I(q, \epsilon) > I(q, 0) \geq \epsilon_0$, at least two iterations of initial EPP are required to sufficiently reduce the infidelity to a level comparable to the threshold. Subsequent iterations of EPP are not expected to significantly decrease the infidelity as it's lower bounded by $I(0, \epsilon)$. As we need $\epsilon_6(I(I(q, \epsilon)/3, \epsilon), \epsilon) \leq \epsilon_0$, from simulation we obtain an upper bound on ϵ being $\epsilon \leq \epsilon'_0 = 2.25 \times 10^{-4}$. Hence, to ensure the efficacy of *Direct Encoding*, it is imperative to reduce the threshold, and thus the physical error rate. For higher-level simulation, when $\epsilon = \epsilon'_0$, $\epsilon_6 = \epsilon_0$, we will run the system of equations with ϵ_6 included, such that $\sigma_6 = 2.09$ and $\epsilon_0^{(0)} = \epsilon'_0$. The threshold equation is modified as $\epsilon_0^{(k+1)} = \min_{i \leq (k+1)} \left\{ 1/D_5^{(i)}, 1/D_6^{(i)} \right\}$.

Now, to analyze the bounds for $\mathbb{P}(\overline{\text{ebit}}^{(k)} \text{ bad})$, we will employ the bounds on exRecs outlined in the preceding section. Instead of directly utilizing the bounds on $\varepsilon_1^{(k)}, \varepsilon_2^{(k)}$, and $\varepsilon_5^{(k)}$, we choose to analyze with level- $(k-1)$ gadgets as a detour. As we will see later, this is necessary to make a fair comparison with the *Interface+EPP* methods and also to obtain tighter bounds. The total number of locations in the circuit is $\gamma_{\text{EPR}} = 319$. It is noteworthy that, in addition to the standard stabilizers of the Steane code, the encoded EPRs are stabilized by $\{\overline{XX}, \overline{ZZ}\}$. For this reason we will obtain the MPM for $\overline{\text{ebit}}^{(k)}$ and bound the logical error rate with level- $(k-1)$ simulation. Definition 3.3.4 justifies the feasibility of the following calculations. In summary, we have, for level- k encoding,

$$\begin{aligned}
& \sum_{j \leq i=1}^6 \alpha_{\text{EPR}}(i, j) \mu_i^{(k-1)} \mu_j^{(k-1)} \left(1 - \nu_5^{(k-1)}\right)^{\gamma_{\text{EPR}}-9} \dots \\
& \dots \left(1 - \nu_6^{(k-1)}\right)^7 \leq \mathbb{P}(\overline{\text{ebit}}^{(k)} \text{ bad}) \\
& \leq \sum_{j \leq i=1}^6 \alpha_{\text{EPR}}(i, j) \nu_i^{(k-1)} \nu_j^{(k-1)} + \dots \\
& \dots \overline{F}(\mathbf{n}_{\text{EPR}}, \underline{\sigma}_L^{(k)}, \underline{\sigma}_U^{(k)}, \alpha_{\text{EPR}}) \left(\nu_5^{(k-1)}\right)^3
\end{aligned}$$

where α_{EPR} is as in Appendix D.3. For $k = 1$, if ϵ, ϵ_6 are as above, then

$$\begin{aligned}
& 2197.6\epsilon^2(1 - \epsilon)^{310}(1 - \epsilon_6)^7 \leq \mathbb{P}(\overline{\text{ebit}}^{(1)} \text{ bad}) \\
& \leq 2197.6\epsilon^2 + 1.355 \times 10^6 \epsilon^3
\end{aligned}$$

For $k \geq 2$, To put the bounds in a simpler form, we note that $\{\underline{\sigma}_L^{(k)}\}, \{\underline{\sigma}_U^{(k)}\}$ are bounded sequences, so for the lower and upper bounds we may take the infimum and supremum

respectively. The detailed data is left in Appendix A.3. For the upper bound, if we denote $\sigma_{U,i}^{\sup} = \sup_k \sigma_{U,i}^{(k)}$ and $\sigma_{L,i}^{\inf} = \inf_k \sigma_{L,i}^{(k)}$,

$$\begin{aligned}
& \mathbb{P}(\overline{\text{ebit}}^{(k)} \text{ bad}) \\
& \leq \sum_{i,j} \alpha_{\text{EPR}}(i,j) \sigma_{U,i}^{\sup} \sigma_{U,j}^{\sup} \left(\nu_5^{(k-1)} \right)^2 \\
& \quad + \overline{F}(\mathbf{n}_{\text{EPR}}, \underline{\sigma}_L^{\inf}, \underline{\sigma}_U^{\sup}, \alpha_{\text{EPR}}) \left(\nu_5^{(k-1)} \right)^3 \\
& \leq 1757.3 \left(\nu_5^{(k-1)} \right)^2 + 1.143 \times 10^6 \left(\nu_5^{(k-1)} \right)^3 \\
& \leq 1880.5 \left(\nu_5^{(k-1)} \right)^2 \\
& \leq 0.883 \epsilon_0 \left(\frac{\epsilon}{\epsilon_0} \right)^{2^k}
\end{aligned}$$

where the last two inequalities follow from the fact that $\epsilon \leq \epsilon'_0 < \epsilon_0/2$ and given this, when $k \geq 2$, $\nu_5^{(k)} \leq \epsilon_0(\epsilon/\epsilon_0)^{2^k}$ is a good upper bound. Similarly for the lower bound,

$$\begin{aligned}
& \mathbb{P}(\overline{\text{ebit}}^{(k)} \text{ bad}) \\
& \geq \sum_{i,j} \alpha_{\text{EPR}}(i,j) \sigma_{L,i}^{\inf} \sigma_{L,j}^{\inf} \left(\mu_5^{(k-1)} \right)^2 - 317 \nu_5^{(k-1)} - 7 \nu_6^{(k-1)} \\
& \geq 1087.2 \left(\mu_5^{(k-1)} \right)^2 \\
& \geq 1.02 \mu_0 \left(\frac{\epsilon}{\mu_0} \right)^{2^k}
\end{aligned}$$

where $\mu_0 = 1/1061.0 \approx 9.43 \times 10^{-4}$.

4.5 Interface + EPP

4.5.1 Interface

To start off this section, we formally introduce the *Interface*. To motivate the construction, we recall that in Chapter 3 Section 3.2.3 we discussed FT encoding of a known state, in that case, $|0\rangle$. Here as we hope to minimize the number of initial ebits, it would

unknown states. Therefore they perform EPP procedures to filter out the 'bad' pairs. The scheme used in our work is shown below.

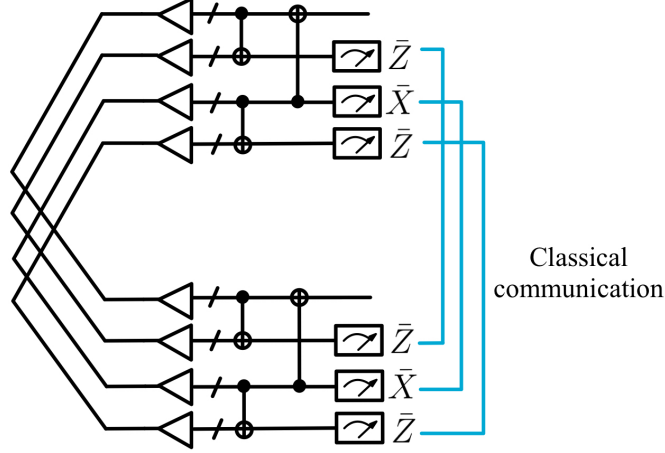


Figure 4.5: A broad overview of the Interface+EPP method. On the left-hand side, the curved lines represent ebits. The triangles are the interfaces. This is followed by $\overline{\text{EPP}}$ and classical communication.

Alice and Bob commence with 4 ebits, which they employ via an interface to encode them into logical qubits. Subsequently, they individually perform logical CNOT operations on the four logical qubits they possess. Finally, they conduct destructive measurements on the last three logical qubits, applying measurements in the $\overline{Z}$, $\overline{X}$, and $\overline{Z}$ bases, respectively. Upon obtaining classical outputs, they post-process (decode) the results and compare them through error-free classical communication.

In the remainder of the section, we establish that the failure probability of the EPP using the interface remains bounded, independent of the parameter k . Following this, we analyze the logical error rate of the $\overline{\text{ebit}}^{(k)}$ given acceptance of $\overline{\text{EPP}}$. Regarding the physical error rate, the local threshold value solely depends on the local CNOT-exRec, which follows the previously described construction, with the same threshold ϵ_0 applicable. We note that the nonlocal resources, in this case, are bare ebits (potentially after initial rounds of EPP), thus $\epsilon_6 = 3q$, the infidelity of the initial ebits. Unlike in Direct Encoding, there's no need to convert them to CNOT gates here. Besides, from the circuit construction, it's evident that ϵ_6 doesn't affect the local threshold.

However, as demonstrated later, the final logical error rate will have dependence on q , thus we will also establish a lower bound on the threshold value for q . The number of initial EPP as in Figure 4.1 needed will also depend on this threshold. To achieve exponential suppression, we explore two schemes and provide a detailed comparison in subsequent sections.

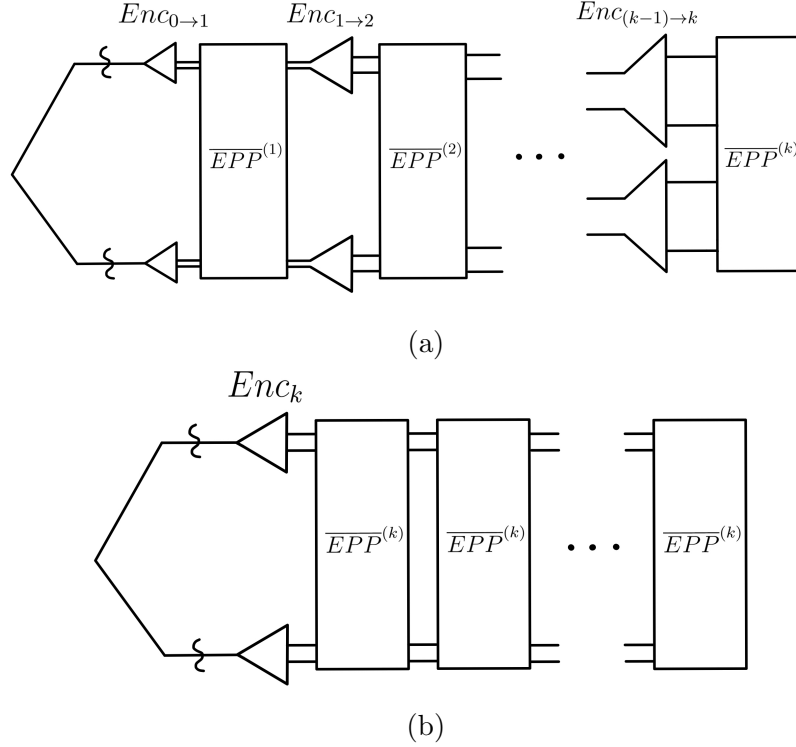


Figure 4.6: (a) Scheme A. Encoding is done sequentially, level by level, each followed by a logical EPP with 3 other similarly prepared ebit. (b) Scheme B. State is encoded to the k -th level in a single interface, followed by rounds of $\overline{EPP}^{(k)}$ to reduce the logical error rate.

In Scheme B, Alice and Bob share 4 physical EPR pairs, encode to $k = 1$, and perform $\overline{EPP}^{(1)}$ as above. They then encode to $k = 2$ and do EPP again with other 3 similarly prepared ebit⁽²⁾ etc. In Scheme A, for the ebits Alice and Bob share, they encode locally to level- k through the interface first. They then conduct $\overline{EPP}^{(k)}$ repeatedly with similarly prepared ebit^(k). Through these two procedures, the logical error rate will decrease as

$\mathcal{O}\left((\epsilon_6)^{2^l}\right)$ where l is the number of EPPs performed.

4.5.3 Logical error rate of Interface

In the noisy circuit scenario, the recovery operation P is followed by an identity gate on each qubit as the noise. We will investigate some properties of this interface. We will show a result similar to Lemma III.6 in [26] but with a tighter upper bound.

Lemma 4.5.1 (Error probability of Enc_l).

$$\begin{aligned} & \mathbb{P}(\text{Enc}_l \text{ bad}) \\ & \leq \exp(\beta_0 \cdot C_1 (\epsilon + \rho_1(\epsilon))) (\beta_1 \epsilon + \beta_2 \epsilon^2 + \beta_3 \rho_1(\epsilon) + \beta_4 \rho_1(\epsilon)^2) \end{aligned}$$

where $C_1 = 18.1$, $\underline{\beta} = (1.0043 \quad 2.8 \quad 628.5 \quad 2.43 \quad 521.3)$ and $\rho_1(\epsilon) = \sum_{k=1}^{\infty} \nu_5^{(k)}$.

Proof. To compute $\mathbb{P}(\text{Enc}_l \text{ bad})$, we note that in the interface there are also ancilla state verifications, let H denote the instances that $|\Omega\rangle^{(k)}$ accepted $\forall k \leq l-1$ given that $|\bar{0}\rangle^{(k)}$'s are accepted, what we really want is actually $\mathbb{P}(\text{Enc}_l \text{ bad} | H)$. To compute this conditional probability we will compute $\mathbb{P}(\text{Enc}_l \text{ bad} \wedge H)$ first.

An interface consists of the teleportation(43 locations) and the EC gadget(68 locations). We shall proceed with $\text{Enc}_{0 \rightarrow 1}$ first. By simulation, the teleportation circuit has on average $[0, 1, 1, 1, 2, 0, 0]$ locations(e.g. the second entry means 1 Loc_2 -fault location) that will cause $\text{Enc}_{0 \rightarrow 1}$ to have a bad encoded state. If we encode to level- k , since we have $\text{Enc}_k = \text{Enc}_{(k-1) \rightarrow k} \circ \dots \circ \text{Enc}_{1 \rightarrow 2} \circ \text{Enc}_{0 \rightarrow 1}$ and by applying the union bound, the sum of the first-order(in $\varepsilon_5^{(k)}$) terms will be upper bounded by

$$\begin{aligned} & \leq \sum_{k=0}^{l-1} \left(\varepsilon_2^{(k)} + \varepsilon_3^{(k)} + \varepsilon_4^{(k)} + 2\varepsilon_5^{(k)} \right) \\ & = \sum_{k=0}^{l-1} \left(\sigma_2^{(k)} + 2\sigma_3^{(k)} + 2 \right) \varepsilon_5^{(k)} \end{aligned}$$

From Appendix A.3 we know that $\sigma_i^{(k)}$ is bounded above and we numerically obtain the

values. Therefore we arrive at the upper bound,

$$2.8\epsilon + 2.43 \sum_{k=1}^{\infty} \nu_5^{(k)}$$

where we replaced the finite sum with an infinite sum to encompass all $k \in \mathbb{N}$. The convergence can be seen by noticing that $\nu_5^{(k)} \leq \epsilon_0 (\epsilon/\epsilon_0)^{2^k}$, $\forall k \geq 2$, the infinite sum is upper bounded by a convergent geometric series and thus it's also convergent when $\epsilon < \epsilon_0$ ⁴. For the sake of simplicity and to obtain a tight bound while maintaining the current threshold value, we resort to numerics. Let us denote this converging limit as $\rho_1(\epsilon)$, For example, if $\epsilon = \epsilon_0$, $\rho_1(\epsilon) \approx 1.27 \times 10^{-3}$. In summary, for first-order terms, we have the upper bound $2.8\epsilon + 2.43\rho_1(\epsilon)$.

If two locations have faults, we again have the MPM α_{in} as in Appendix D. Similar to before, we have the following upper bound

$$\sum_{k=0}^{l-1} \sum_{i,j} \alpha_{in}(i,j) \nu_i^{(k)} \nu_j^{(k)} + \overline{F}(\mathbf{n}_{in}, \underline{\sigma}_L^{(k)}, \underline{\sigma}_U^{(k)}, \alpha_{in}) \left(\nu_5^{(k)} \right)^3$$

where $\mathbf{n}_{in} = [14, 12, 9, 11, 58, 0, 7]$. In $\overline{F}$ we may replace $\underline{\sigma}_L^{(k)}$ and $\underline{\sigma}_U^{(k)}$ by $\underline{\sigma}_U^{\sup}$ and $\underline{\sigma}_L^{\inf}$. In this case, $\overline{F}$ evaluates to a constant 36372.3 $\forall k \geq 1$ and 44437.8 when $k = 0$. For the first term, we can also simplify,

$$\begin{aligned} & \sum_{k=0}^{l-1} \sum_{i,j} \alpha_{in}(i,j) \nu_i^{(k)} \nu_j^{(k)} \\ & \leq 557.8\epsilon^2 + \sum_{i,j} \alpha_{in}(i,j) \sigma_{U,i}^{\max} \sigma_{U,j}^{\max} \sum_{k=1}^{\infty} \left(\nu_5^{(k)} \right)^2 \\ & = 557.8\epsilon^2 + 451.6 \sum_{k=1}^{\infty} \left(\nu_5^{(k)} \right)^2 \end{aligned}$$

Thus by summing up the results above and applying Prop A.2.0.1 we obtain an upper

⁴This infinite series is also known as lacunary series, which has no simply closed-form expression, an expression derived from Fourier transform can be found [45].

bound for the second-order terms

$$\leq 628.5\epsilon^2 + 451.6 \sum_{k=1}^{\infty} \left(\nu_5^{(k)} \right)^2 + 36372.3 \sum_{k=1}^{\infty} \left(\nu_5^{(k)} \right)^3$$

By the simple observation that $\sum_{k=1}^{\infty} a_i^k \leq (\sum_{k=1}^{\infty} a_i)^k$ for $a_i \geq 0 \ \forall i$, we obtain,

$$\begin{aligned} & \mathbb{P}(\text{Enc}_l \text{ bad} \wedge H) \\ & \leq 2.8\epsilon + 2.43\rho_1(\epsilon) + 628.5\epsilon^2 + 521.3\rho_1(\epsilon)^2 \end{aligned}$$

Now we compute a lower bound on $\mathbb{P}(H)$, as the ancillary states $|\Omega\rangle^{(k)}$ are independent.

$$\begin{aligned} \mathbb{P}(H) &= \sum_{k=0}^{l-1} \mathbb{P} \left(|\Omega^{(k)}\rangle \text{ accepted} \mid |\bar{0}^{(k)}\rangle \text{ accepted} \right) \\ &= \prod_{k=0}^{l-1} \frac{\mathbb{P} \left(|\Omega^{(k)}\rangle \text{ accepted} \wedge |\bar{0}^{(k)}\rangle \text{ accepted} \right)}{\mathbb{P} \left(|\bar{0}^{(k)}\rangle \text{ accepted} \right)} \\ &\geq \prod_{k=0}^{l-1} \mathbb{P} \left(|\Omega^{(k)}\rangle \text{ accepted} \wedge |\bar{0}^{(k)}\rangle \text{ accepted} \right) \\ &\geq \prod_{k=0}^{l-1} \left(1 - C_1 \nu_5^{(k)} \right) \end{aligned}$$

where $C_1 = 18.1$. We will now need to upper bound $\mathbb{P}(H)^{-1}$,

$$\begin{aligned}
& \prod_{k=0}^{l-1} \left(1 - C_1 \nu_5^{(k)}\right)^{-1} \\
& \leq \prod_{k=0}^{\infty} \left(1 - C_1 \nu_5^{(k)}\right)^{-1} \\
& = \exp \left(- \sum_{k=0}^{\infty} \log \left(1 - C_1 \nu_5^{(k)}\right) \right) \\
& \leq \exp \left(\sum_{k=0}^{\infty} \frac{C_1 \nu_5^{(k)}}{\sqrt{1 - C_1 \nu_5^{(k)}}} \right) \\
& \leq \exp \left(\sum_{k=0}^{\infty} 1.0043 \cdot C_1 \nu_5^{(k)} \right) \\
& = \exp (1.0043 \cdot C_1 (\epsilon + \rho_1(\epsilon)))
\end{aligned}$$

where the second inequality follows from the fact that $\forall x \in (0, 1)$, $\log(1 - x) \geq \frac{-x}{\sqrt{1-x}}$ and the third follows from that $\nu_5^{(k)} \leq \epsilon_0 \ \forall k \geq 1$. Combining the above results gives us the Lemma. $\square$

We will use $f_{in}(\epsilon)$ to denote this upper bound in later context. From the bound, we observe that it's independent of k . In particular, when $\epsilon = \epsilon_0$, $\mathbb{P}(\text{Enc}_l \text{ bad}) \approx 4.40 \times 10^{-3}$, the detailed behaviour will be illustrated in Appendix C.1. We note that, for $\epsilon \leq \epsilon_0$, we can actually upper bound $\rho_1(\epsilon)$ with a geometric series, which evaluates to a constant independent of ϵ , which in turn gives a linear upper bound. However, in the pursuit of tight bounds, we will refrain from further elaboration here. Additionally, it's worth noting that based on the aforementioned proof, the upper limit of the failure probability for $\text{Enc}_{l \rightarrow (l+1)}$ is bounded by $f_{in}^{(l+1)}(\epsilon) = 521.4 \nu_5^{(l)} (1 - C \nu_5^{(l)})$, which will prove to be useful in subsequent derivations.

For later analysis, the interface needs to be examined more closely, in particular, we will investigate the number of single-fault locations and malignant pairs that cause different logical errors. We will leave this more detailed analysis in Appendix C.1.

EPP rejection probability

We will first give bounds on the failure probability of EPP, part of the results in the proofs are helpful for later derivation. We will state the upper and lower bounds for the above building block below and the proofs will be left in Appendix C.2. Thus we have the following

Theorem 4.5.2 (Upper bound for $\mathbb{P}(\text{EPR rejected})$). *Let Enc_k be the k -th level interface circuit from Figure 4.4, then $\forall k \geq 1$,*

$$\mathbb{P}(\overline{\text{ebit}}^{(k)} \text{ rejected}) \leq 4\epsilon_6 + 12\epsilon + 8f_{in}(\epsilon) \quad (4.1)$$

and the lower bound,

Corollary 4.5.2.1 (Lower bound for $\mathbb{P}(\text{EPR rejected})$). *Let Enc_k be the k -th level interface circuit from Figure 4.4, then for $k \geq 2$,*

$$\mathbb{P}(\overline{\text{ebit}}^{(k)} \text{ rejected}) \geq 22.4\epsilon + 4\epsilon_6 - \sum_{i=0}^2 u_i \epsilon^i \epsilon_6^{2-i}$$

where $\underline{u} = (16 \quad 4404 \quad 6067)$ up to $\mathcal{O}(\epsilon^3)$.

We will denote the lower and upper bound by $f_1(\epsilon, \sigma_6)$ and $f_2(\epsilon, \sigma_6)$ respectively for convenience.

4.5.4 Logical error rate of Interface+EPP

Having obtained bounds for the failure probability of encoded EPP, we analyze the logical error rate given that the EPR pair is accepted via EPP. The error probability can be analyzed similarly as in the previous section, except now the individual components of the circuit are not all fault-tolerant, and the interface preparation will cause problems. In addition, the fact that EPP rejects some EPR pairs makes the analysis more complex. In particular, we can no longer upper bound $\mathcal{O}(\epsilon^3)$ by simply calculating all combinations, which will result in significant overcounting. Thus, for a more systematic analysis, we will convert the circuit into a network flow problem, which will be discussed in detail

for two schemes below. Note that we will obtain $\mathbb{P}(\text{EPR bad} \wedge \text{EPR accepted})$ first and $\mathbb{P}(\text{EPR bad} \wedge \text{EPR accepted})$ will follow by

$$\begin{aligned} & \mathbb{P}(\text{EPR bad} \wedge \text{EPP accepted}) \\ &= \frac{\mathbb{P}(\text{EPR bad} \wedge \text{EPR accepted})}{1 - \mathbb{P}(\text{EPR rejected})} \end{aligned}$$

Before we delve into the analysis, we will first translate the circuit into a network flow graph to help us compute $\mathbb{P}(\text{EPR bad} \wedge \text{EPR accepted})$. The flow represents the propagation of errors. For convenience, we will analyze the $\bar{X}$ and $\bar{Z}$ errors separately. The graphs will help us decide the combinations of failures of exRecs/interfaces that would lead to logical $\bar{X}$ -error despite passing the EPP test. The graph can be set up as follows: Let $N_{A,X} = (V, E)$ be the network above with $S, T \in V$ being the source and sink of $N_{A,X}$ respectively and f is a function on the edges of $N_{A,X}$, its value on $(x, y) \in E$ is denoted by $f(x, y)$. $c(x, y)$ denotes the capacity of the edge. The propagation of $\bar{X}$ -errors is then shown by the directed-graph

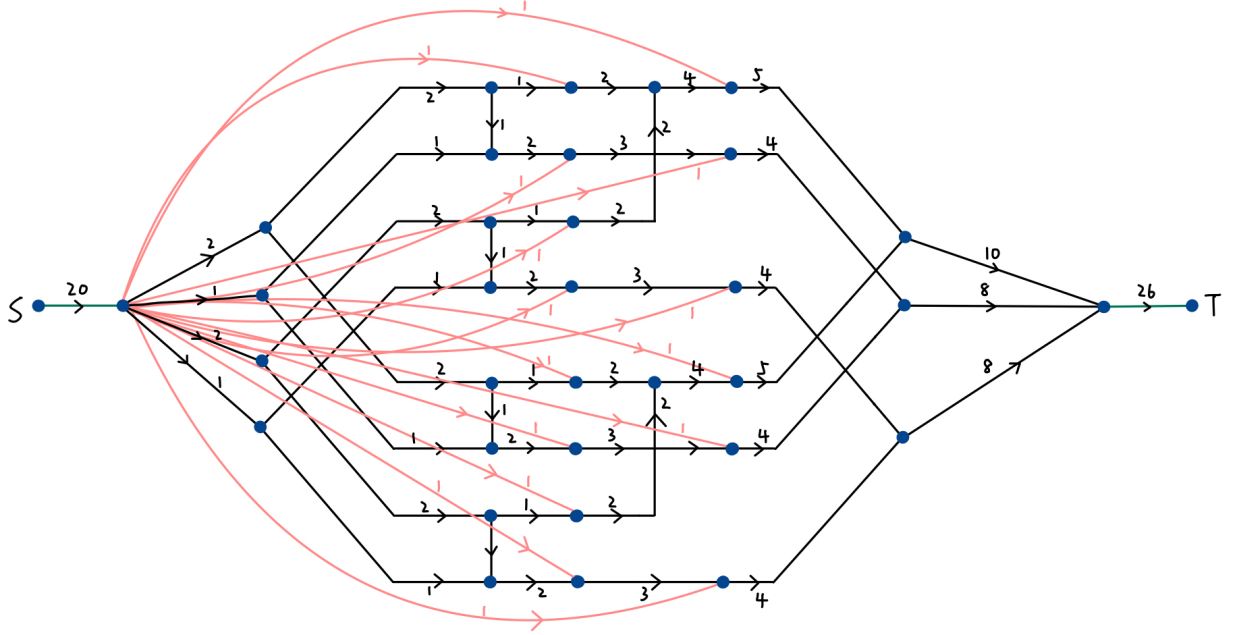


Figure 4.7: Network for $\bar{X}$ -error. The source S and sink T are introduced for completeness. The numbers above the edges indicate the capacities. The pink edges all have capacity 1.

There are a couple of details worth clarifying

- The central block is the standard *Interface+EPP* scheme. The third logical EPR pair has no directly-connected $\{D_i\}$ vertex because in the scheme we perform X measurement here and $\overline{X}$ -error will not cause an issue.
- The parity of flow in the circuit indicates the presence of $\overline{X}$ -error. If the flow is even, then the combined $\overline{X}$ -errors cancel out, equivalent to no error; otherwise there is $\overline{X}$ -error.
- Some edges with capacity 1 in the middle of the graph are directly connected with the source e.g. A_2 as the CNOT-exRec in the circuit will potentially introduce $\overline{X}$ -error. For instance, if $f(S, A_2) = 1$ and $f((S, A_6)) = 0$, this indicates the first CNOT-exRec introduces $\overline{X}$ -error. Besides, $SA_7, SA_{12}, SB_7, SB_{12}$ are introduced to indicate potential errors on mmt-exRecs.

Then the combinations of failures that contribute to $\mathbb{P}(\overline{\text{ebit bad}} | \overline{\text{EPP}} \text{ accepts})$ can be determined via the following feasibility problem

$$\begin{aligned}
& \min 0 \\
& \text{s.t. } f(x, y) \leq c(x, y) \quad \forall (x, y) \in E \\
& \sum_{x:(x,y) \in E, f(x,y) > 0} f(x, y) = \sum_{x:(y,x) \in E, f(y,x) > 0} f(y, x) \\
& \quad \forall y \in V \setminus \{S, T\} \\
& f(x, y) \cdot (f(x, y) - c(x, y)) = 0 \\
& \quad \forall (S, y) \in E \\
& f(D_1, T) = 2y_1 + 1 \\
& f(D_2, T) = 2y_2 \\
& f(D_3, T) = 2y_3 \\
& y_1, y_2, y_3 \in \mathbb{N} \\
& f(x, y) \in \mathbb{N} \quad \forall (x, y) \in E
\end{aligned}$$

The first two are standard flow constraints. The third one makes sure either the initial edge has $\overline{X}$ -error or error-free. $f(D_2, T)$ and $f(D_3, T)$ need to be even as for acceptance, the measurement results of two parties need to agree. $f(D_1, T)$ odd as for the desired scenario, we need the first $\overline{\text{ebit}}$ to have error. If $f(D_1, T)$ is even, this means the first logical EPR

pair either has no error, or has $\overline{XX}$ -error. However it is stabilized by $\overline{XX}$, this will not cause any problem. We give one example of the solution in Figure 4.8.

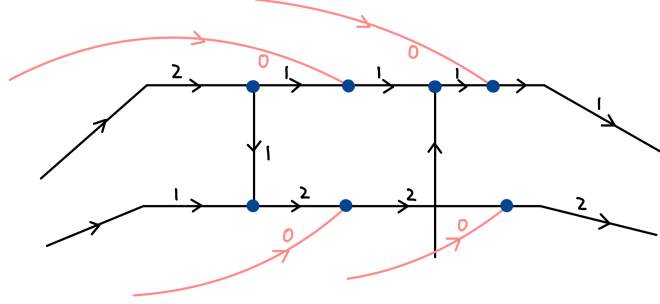


Figure 4.8: An example solution of the feasibility problem. This indicates an $\overline{X}$ -error on the first ebit given $\overline{\text{EPP}}$ accepts.

This shows that on Alice' side, her logical qubits of the first two EPR pairs both contain $\overline{X}$ -errors and they end up canceling out on the second data block, thus leading to acceptance while having errors on the first data block. Due to the third constraint, it would be easier to search for solutions exhaustively.

Analogously based on the propagation rule of $\overline{Z}$ -error, we can generate a network graph for the propagation of $\overline{Z}$ -error, the central block on Alice' side is shown in Figure 4.9. The capacities can be chosen suitably. The details will not be repeated.

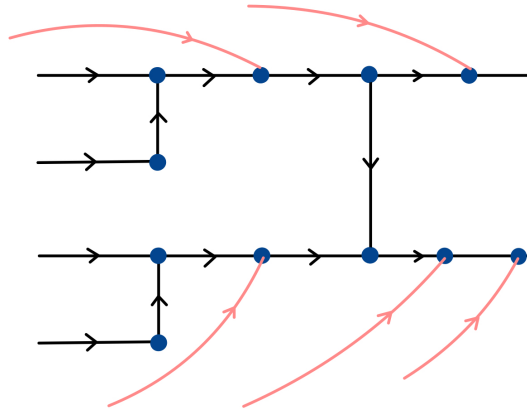


Figure 4.9

Any solution that is common to both types of errors indicates a potential $\bar{Y}$ -error. Now we have a way to identify cases that lead to failure, we will match them with error terms of different orders. In the following we will denote the 4 ebits part as $\mathcal{P}_1$ and the $\overline{\text{EPP}}$ part as $\mathcal{P}_2$. Again we will obtain bounds for $k = 1$ and then generalize to higher-level encoding. When there is one fault in the circuit, it will not cause a logical error in the EPP circuit according to fault-tolerance conditions. It can only cause logical error when it's present in the EPR preparation, thus one of SC_1, SC_2, SC_3, SC_4 is saturated while all other edges from the source are empty. There is no solution in both $\bar{X}$ and $\bar{Z}$ problems. Hence there are no first-order terms (which in turn also verifies our claim that the scheme is fault-tolerant).

For second-order terms, we will consider the distribution of the faults. For convenience of discussion, we will first label the $\overline{\text{ebit}}^{(k)}$'s and exRecs as follows,

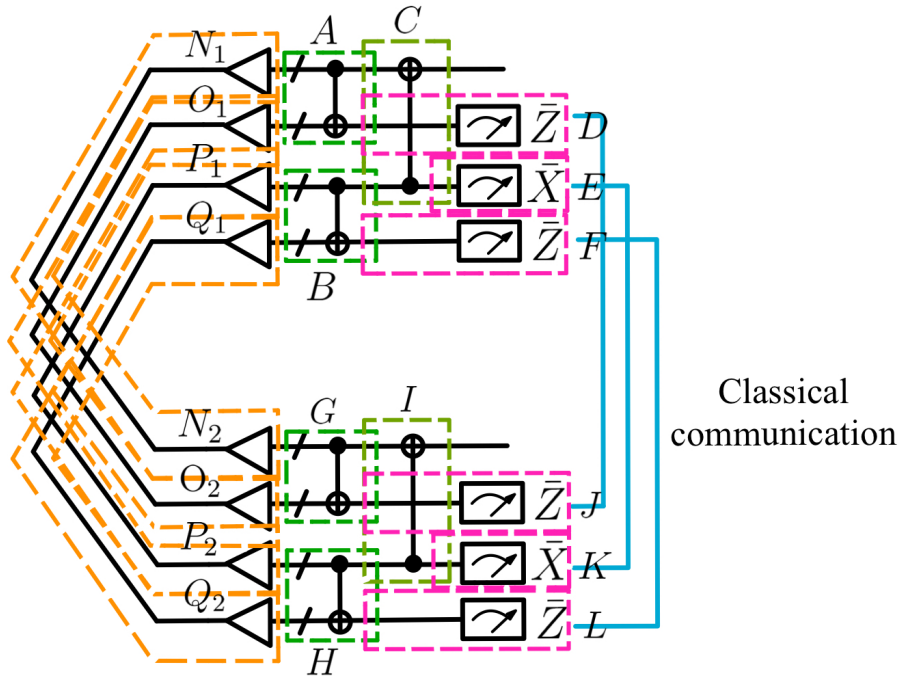


Figure 4.10: An Interface+EPP procedure broken down into its individual components. The orange parts denote ebits prepared via physical ebits and interfaces. Other parts correspond to exRecs within $\overline{\text{EPP}}$. For example, A denotes the first CNOT-exRec on Alice's side.

From the different cases and the fact that 2 faults can cause at most 2 logical errors,

we can post-process the feasible solutions and we'll present them in a more readable form in terms of the components that are bad. By combining cases that will result in both $\overline{X}$ and $\overline{Z}$ errors, we obtain the following cases

- (2I) If they are both in $\mathcal{P}_2$, then they would contribute to the joint probability only when they are in the same exRec and form a malignant pair. From the filtered feasible solutions, we have the cases,

- $\overline{XI}, \overline{XZ}$ on $A/B/G/H$
- $\overline{IX}, \overline{IZ}, \overline{IY}, \overline{XY}, \overline{XZ}, \overline{XX}$ on C/I

The errors on the second one are reversed because the CNOT-exRec in the circuit is reversed.

- (2II) If they are both in $\mathcal{P}_1$, they must each be in one ebit and cause canceling errors. They must both have the same error type (meaning both are $\overline{X}, \overline{Y}$ or $\overline{Z}$). Therefore,

- Both $\overline{X}$: $N_1O_1, N_1O_2, O_1N_2, P_1Q_1, P_1Q_2, P_1Q_1, N_2Q_2, P_2Q_2$
- Both $\overline{Z}$: $N_1P_1, N_1Q_1, N_1P_2, N_1Q_2, O_1P_1, O_1Q_1, O_1P_2, O_1Q_2$

- (2III) If one is in $\mathcal{P}_1$ and one in $\mathcal{P}_2$, this is equivalent to the order-one case because according to fault-tolerance properties, the fault in $\mathcal{P}_2$ will result in correct simulation.

Identifying cases that include errors in $\mathcal{P}_1$ is crucial. The reason will be made clear when we generalize to level- k . So for $k = 1$ we have

- (2I) To enumerate the number of such cases, we recall that the number of malignant pairs in a CNOT-exRec is 1656.3 and we then use the probability distribution of errors from Appendix D.1 and we count, in total 2587.3 malignant pairs.
- (2II) Since both errors are in $\mathcal{P}_1$, we refer to Table C.1 and count 29.1 pairs that do not contain Loc_6 , 11.6 pairs that contain one Loc_6 and 1.32 pairs that are both Loc_6 's. This differentiation is essential, which we will see when we generalize to level- k .

Now we consider third-order terms. There are again a number of cases and from combining feasible solutions and suitable upper bounds, we summarize the following cases:

- (3I) When the three faults are all in $\mathcal{P}_2$, this will only contribute to the desired probability if, they are all in C/I ; they are in two consecutive exRecs and there is one error in the overlapping EC. So we upper bound the combinations and count 589499.6 such cases for $k = 1$.

- (3II) One fault in $\mathcal{P}_1$ and two faults in $\mathcal{P}_2$ forming a malignant pair. We count an upper bound of 281160.3 cases not containing ϵ_6 and 40165.8 containing ϵ_6 .
- (3III) The faults are in $\mathcal{P}_1$, two faults in one $\overline{\text{ebit}}$ and one in another. We count 8591.3 triples excluding ϵ_6 , 2645.4 triples including one ϵ_6 .
- (3IV) The faults are all in $\mathcal{P}_1$, with each in one 'bad' interface respectively. From the feasible solutions, we see there are no combinations that are of the same error type, so cases resulting in $\overline{X}$ and $\overline{Z}$ errors with one overlapping part. This case will be included in the second-order terms.

Having analyzed the different cases, we can obtain the following approximate bounds for $\mathbb{P}(\overline{\text{ebit}}^{(1)} \text{ bad})$,

$$\begin{aligned}
& (2587.3\epsilon^2(1-\epsilon)^{\gamma_1-2}(1-\epsilon_6)^4 + 11.6\epsilon_6\epsilon(1-\epsilon)^{\gamma_1-1} \dots \\
& \dots (1-\epsilon_6)^3 + 1.32\epsilon_6^2(1-\epsilon)^{\gamma_1}(1-\epsilon_6)^2)(1-f_1(\epsilon))^{-1} \\
& \leq \mathbb{P}(\overline{\text{ebit}}^{(1)} \text{ bad}) \\
& \leq (1.32\epsilon_6^2 + 11.6\epsilon_6\epsilon + 2587.9\epsilon^2 + 879251.2\epsilon^3 \\
& \quad + 42811.2\epsilon_6\epsilon^2)(1-f_2(\epsilon))^{-1},
\end{aligned}$$

where $\gamma_1 = 1788$ is the total number of locations excluding Loc_6 at $k = 1$. To ensure a fair comparison with *Direct Encoding* later, we will confine $\epsilon \leq \epsilon'_0$ and consequently employ statistics of σ 's based on this constraint. We will denote the upper bound by $\kappa_2(1, \sigma_6)\epsilon^2$, where $\kappa_2(1, \epsilon_6)$ is obtained from the fact that $\epsilon \leq \epsilon'_0$ here. For the lower bound we will denote as $\sum_{i=0}^2 \kappa_{1,i}(1, \sigma_6)\epsilon^2$ where each term is the coefficient for $(\sigma_6)^i\epsilon^2$, the reason for this special treatment will be clear when we generalize to level- k . Theorem 4.5.2 (by taking $\epsilon = \epsilon'_0$ we have a supremum for f_2). We will then generalize this to level- k . In fact, *Interface+EPP* is guaranteed to function effectively as long as $\epsilon \leq \epsilon_0$, demonstrating one main advantage of this scheme.

4.5.5 Scheme A

For scheme A, we encode physical qubits to level- k and then perform EPP repeatedly. So we would like to compute the logical error rate of $\overline{\text{ebit}}^{(k)}$ first. We will again consider errors of different order. For the second-order terms, the contribution from (2I) terms are upper bounded by $\sum_{i,j} \alpha(i,j) \sigma_{U,i}^{\sup} \sigma_{U,j}^{\sup} \left(\nu_5^{(k-1)} \right)^2 \leq 2084.2 \left(\nu_5^{(k-1)} \right)^2 \leq 2084.2/2.1^{2^k-2}\epsilon^2$.

Thus we see that the contribution from $\mathcal{P}_2$ diminishes exponentially as k increases. In fact, when $k \geq 5$, this term is negligible. Next, (2II) terms persist. For third-order terms, again (3I) and (3II) are negligible as they give $\mathcal{O}(\epsilon^5)$ contribution; (3III) terms remain. When $k \geq 2$, there is an additional case, that is, when there are two faults in one $Enc_{1 \rightarrow 2}$ and one fault in some other interface that combine and have a canceling effect. From the proof of Lemma 4.5.1, we can upper bound such cases by $\frac{1}{2}(\sigma_2^{(1)} + 2\sigma_3^{(1)} + 2)\epsilon_5^{(1)} \cdot (2.8 \cdot 64\epsilon + 34\epsilon_6)$, from the simulated values of $\sigma_i^{(1)}$ we have $445775.5\epsilon^3 + 84577.9\epsilon^2\epsilon_6$. If we denote the k -th level encoded $\overline{\text{ebit}}$ after l rounds of $\overline{\text{EPP}}$ by $\overline{\text{ebit}}_l^{(k)}$, therefore we would have the upper bound $\forall k \geq 2$

$$\begin{aligned} \mathbb{P}(\overline{\text{ebit}}_1^{(k)} \text{ bad}) &\leq (2084.2/2.1^{2^k-2}\epsilon^2 + 29.1\epsilon^2 + 11.6\epsilon\epsilon_6 \dots \\ &\dots + 1.32\epsilon_6^2 + 454366.8\epsilon^3 + 87223.3\epsilon^2\epsilon_6)(1 - f_2(\epsilon))^{-1} \\ &\leq \kappa_2^A(k, \sigma_6)\epsilon^2. \end{aligned}$$

where $\kappa_2^A(k, \sigma_6)$ is obtained through the same method as above. An example value is $\kappa_2^A(2, 1) = 644.1$. For the lower bound, we will obtain an approximation. From steps in the analysis of Cor. 4.5.2.1 we obtain, $\forall k \geq 2$,

$$\begin{aligned} &\mathbb{P}(\overline{\text{ebit}}_1^{(k)} \text{ bad}) \\ &\geq 29.1\epsilon^2(1 - \epsilon_6)^4(1 - \epsilon)^{-2} + 11.6\epsilon\epsilon_6(1 - \epsilon_6)^3(1 - \epsilon)^{-1} + \\ &\quad 1.32\epsilon_6^2(1 - \epsilon_6)^2 \left(1 - 8 \sum_{t=1}^7 n_t \epsilon_t\right) (1 - f_1(\epsilon))^{-1} \\ &\geq \sum_{i=0}^2 \kappa_{1,i}^A(k, \sigma_6)\epsilon^2. \end{aligned}$$

Thus we have

$$\sum_{i=0}^2 \kappa_{1,i}^A(k, \sigma_6)\epsilon^2 \leq \mathbb{P}(\overline{\text{ebit}}_1^{(k)} \text{ bad}) \leq \kappa_2^A(k, \sigma_6)\epsilon^2.$$

Now to achieve the same exponential suppression of logical error rate, we would need to perform further rounds of EPP. We may denote the upper bound of the logical error rate after the l -th EPP as $g_A^{(l)}(k, \sigma_6)$, where $g_A^{(1)}(k, \sigma_6) = \kappa_2^A(k, \sigma_6)\epsilon^2$. We may first obtain an upper bound on EPP rejection probability. Following a similar argument as in 4.5.2,

$$\mathbb{P}(\overline{\text{ebit}}_l^{(k)} \text{ rejected}) \leq 4g_A^{(l-1)}(k, \sigma_6) + 12\nu_5^{(k)}.$$

Thus for the upper bound, we establish the following recursive relationship for $2 \leq l \leq k$,

$$\begin{aligned} g_A^{(l)}(k, \sigma_6) = & \left(6 \left(g_A^{(l-1)}(k, \sigma_6) \right)^2 \dots \right. \\ & \dots + 2084.2 \left(\nu_5^{(k-1)} \right)^2 + 24 g_A^{(l-1)}(k, \sigma_6) \nu_5^{(k-1)} \left. \right) \dots \\ & \dots \left(1 - 4 g_A^{(l-1)}(k, \sigma_6) - 12 \nu_5^{(k)} \right)^{-1}, \end{aligned} \quad (\dagger)$$

which follows from the union bound and the constants are obtained by counting the feasible solutions. In this case, some feasible solutions are combined. For example, both errors into 0 and 13 in Figure 4.7 are included by $g_A^{(l)}$. When $k \geq 2$, the contribution from the rejection probability term is negligible. If we examine the magnitude of the above terms more closely and denote the second term by b , we can establish the following simple lemma,

Lemma 4.5.3. $\forall k \geq 2, k \in \mathbb{N}, \forall \epsilon > 0$ and $\forall \delta > 24b^2, \exists l' \in \mathbb{N}$ s.t. when $g_A^{(1)} \leq \frac{1-\sqrt{b}}{6}$, $\forall l \geq l', |g_A^{(l)} - b| < \delta$.

Proof. The above recursive relation can be written as $g_A^{(l)} = 6 \left(g_A^{(l-1)} \right)^2 + 0.53\sqrt{b}g_A^{(l-1)} + b$. By solving $g_A^{(l)} < g_A^{(l-1)}$ and suitably lower-bounding the larger solution we obtain the upper bound on $g_A^{(1)}$. This implies that under this condition the series is decreasing, it's also bounded below by b , thus it converges. Since our recursion function is continuous, by Banach fixed-point theorem the fixed point is the converging limit. Hence, by solving the quadratic equation and upper-bounding the solution we obtain a bound on allowed δ . l' is chosen to be the minimal that satisfies the conditions. $\square$

In the following context, we shall use l' to denote the constant that satisfies Lemma 4.5.3. This Lemma indicates that when we perform sufficiently many EPP, $g_A^{(l)}$ will be dominated by the $\left(\nu_5^{(k-1)} \right)^2$ term and no further rounds of EPP will noticeably reduce the logical error rate. This aligns with the observation in the non-encoded case, that the effectiveness of EPP is limited by the error rate of local operations. In the later section, l'_A will be determined numerically given k, ϵ and σ_6 . This lemma greatly simplifies the upper bound

as some terms dominate others. So we have

$$g_A^{(l)}(k, \sigma_6) = \begin{cases} \frac{1}{6} \left(\sqrt{6\kappa_2^A} \epsilon \right)^{2^l}, & 2 \leq l < l'_A \\ 2084.2 \left(\nu_5^{(k-1)} \right)^2, & l > l'_A \end{cases}.$$

Given the same threshold value ϵ'_0 , solving $\sqrt{6\kappa_2^A} = 1/\epsilon'_0$ numerically would give a sufficient bound for ϵ_6 (That is, when σ_6 is below this bound, the logical error rate converges as k increases). For $k \geq 3$, $\sigma_6 \leq 813$, so $\epsilon_6 \leq 0.183$. And this surely satisfies the bound in the lemma. Thus theoretically if we start with noisy EPR pairs with infidelity of less than 18.3% this protocol will work well. But in practice, if we hope to minimize the use of non-local resources, it might be better off to do some initial purifications. This consideration will be accounted for in more detail in the later section where we analyze resources.

Now for the lower bound, we will first consider the derivation of a lower bound for the combined circuit when we have two sequential block components under our construction, with known bounds of the logical error rate of each component. An illustration is shown in Figure 4.11

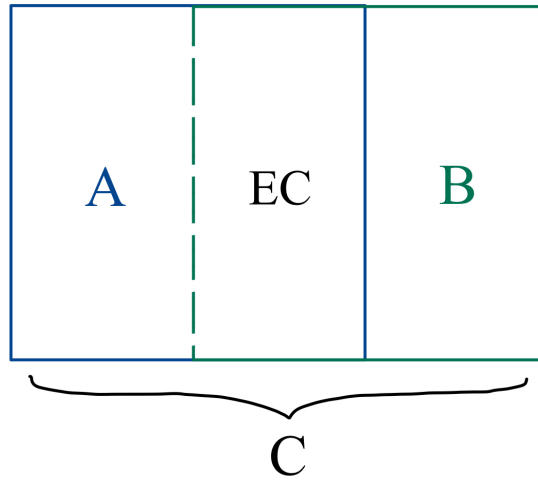


Figure 4.11: Two components A and B with an overlapping set of ECs

According to the definition of badness, components A and B can be treated as inde-

pendent and thus

$$\begin{aligned}\mathbb{P}(C \text{ bad}) &\geq 1 - (1 - \mathbb{P}(A \text{ bad}))(1 - \mathbb{P}(B \text{ bad})) \\ &= \mathbb{P}(A \text{ bad}) + \mathbb{P}(B \text{ bad}) - \mathbb{P}(A \text{ bad}) \cdot \mathbb{P}(B \text{ bad}).\end{aligned}$$

Note that as we encode to higher levels the logical error rate is rather small, so the last term is negligible. Since we can compute lower bounds for $\mathbb{P}(A \text{ bad})$ and $\mathbb{P}(B \text{ bad})$, the lower bound for $\mathbb{P}(C \text{ bad})$ can be established. Further rounds of $\overline{\text{EPP}}$ follow the same rationale. Let us denote the probability of terms of order $(\sigma_6)^i$ of $\kappa_1^A(k, \sigma_6)$ being Z -logical error by $p_{Z,i}^{(k)}(\sigma_6)$, which is determined when we considered Z feasible solutions. It's necessary to separate out $\overline{Z}$ -error from $\overline{X}/\overline{Y}$ -error. As previously indicated, 6 cases of logical errors arise from the failures of two ebit pairs, in which 4 are $\overline{Z}$ -error and 2 are $\overline{X}/\overline{Y}$ -error. We will denote the lower bound on $\overline{Z}$ -error rate and $\overline{X}/\overline{Y}$ -error rate after the l -th $\overline{\text{EPP}}$ as $h_{A,Z}^{(l)}(k, \sigma_6)$ and $h_{A,X/Y}^{(l)}(k, \sigma_6)$ respectively, thus we arrive at the following recursive relations

$$\begin{aligned}h_{A,Z}^{(l)}(k, \sigma_6) &= 4 \left(h_{A,Z}^{(l-1)} \right)^2 + 517.4 \left(\mu_5^{(k-1)} \right)^2 \\ h_{A,X/Y}^{(l)}(k, \sigma_6) &= 2 \left(h_{A,X/Y}^{(l-1)} \right)^2 + 1252.4 \left(\mu_5^{(k-1)} \right)^2,\end{aligned}\tag{‡}$$

where $h_{A,Z}^{(1)} = 4 \sum p_{Z,i}^{(2)} \kappa_{1,i}^A(2, \sigma_6) \epsilon^2$ and $h_{A,X/Y}^{(1)} = 2 \left(\kappa_1^A \epsilon^2 - h_{A,Z}^{(1)}/4 \right)$. Again in this case we will have similar lemma, thus $\exists l_A''$ s.t.

$$h_A^{(l)}(k, \sigma_6) = \begin{cases} \frac{1}{4} \left(\sqrt{h_{A,Z}^{(1)}} \right)^{2^l} + \frac{1}{2} \left(\sqrt{h_{A,X/Y}^{(1)}} \right)^{2^l}, & \text{if } l < l_A'' \\ 1770 \left(\mu_5^{(k-1)} \right)^2, & \text{if } l \geq l_A'' \end{cases}$$

4.5.6 Scheme B

For scheme B , we perform interface encoding and EPP iteratively, i.e. encode to $k = 1$ with $Enc_{0 \rightarrow 1}$ followed by $\overline{\text{EPP}}^{(1)}$ and then encode with $Enc_{1 \rightarrow 2}$ followed by $\overline{\text{EPP}}^{(2)}$ etc. For the logical error rate in this scheme, we follow the same methodology as above. The only difference here is that in every round of $\overline{\text{EPP}}$, contributions from $\mathcal{P}_2$ have the same

magnitude and thus need to be included. Hence for $k \geq 2$ and $2 \leq l \leq k$ we have

$$\begin{aligned}
g_B^{(l)}(k, \sigma_6) &= \left(6 \left(g_B^{(l-1)} \right)^2 + 24 g_B^{(l-1)} \nu_5^{(l-1)} + 2084.2 \left(\nu_5^{(l-1)} \right)^2 \right) \dots \\
&\quad \dots \left(1 - 4 g_B^{(l-1)}(k, \sigma_6) - 12 \nu_5^{(l)} \right)^{-1} \\
h_{B,Z}^{(l)}(k, \sigma_6) &= 4 \left(h_{B,Z}^{(l-1)} \right)^2 + 517.4 \left(\mu_5^{(l-1)} \right)^2 \\
h_{B,X/Y}^{(l)}(k, \sigma_6) &= 2 \left(h_{B,X/Y}^{(l-1)} \right)^2 + 1252.4 \left(\mu_5^{(l-1)} \right)^2
\end{aligned}$$

where $g_B^{(1)} = \kappa_2(1, \sigma_6)\epsilon^2$, $h_{A,Z}^{(1)} = 4 \sum p_{Z,i}^{(1)} \kappa_{1,i}(1, \sigma_6)\epsilon^2$ and $h_{A,X/Y}^{(1)} = 2 \left(\kappa_1(1, \sigma_6)\epsilon^2 - h_{A,Z}^{(1)}/4 \right)$. In this case, the lemma no longer works because for each level $l \leq k$, the magnitudes of each term are comparable, and since $\kappa_2(1, \sigma_6) \geq 2815.6 > 1/\epsilon_0$. To simplify the upper bound, we may convert them to $C^{(l)} \left(\nu_5^{(l-1)} \right)^2$ at each level, where $C^{(l)}$ satisfies the difference equation

$$C^{(l+1)} = \left(6 \left(C^{(l)} \right)^2 + \frac{24}{\epsilon_0} C^{(l)} \right) (\epsilon_0)^2 + 2084.2$$

and $C^{(1)} = \kappa_2(1, \sigma_6)$. It's easy to check that 2114.0 is a stable equilibrium and as long as $\sigma_6 \leq 528$, the sequence will converge to this fixed point. This puts a sufficient bound of 11.9% on ϵ_6 . When $l > k$, we are back to equation (†), except in this case, we start with an initial point of $C^{(k)} \left(\nu_5^{(k-1)} \right)^2$ and previous results hold. Depending on the initial value, if $C^{(k-1)}$ is already δ -close to 2114.0 for small δ , then one additional round of $\overline{\text{EPP}}$ will diminish the logical error rate to $2084.2 \left(\nu_5^{(k-1)} \right)^2$, but further EPP iterations will not enhance the outcome. We will denote the minimal number of $\overline{\text{EPP}}$ for this to occur by l_B'' . Similarly, for the lower bound we can establish the difference equations and the lower bound for the logical error rate converges to $1774 \left(\mu_5^{(l-1)} \right)^2$ provided the converges occurs for $l \leq k$; otherwise after $l = k$, the suppression of logical error rate follows equation (‡). Similarly, we denote this boundary l by l_B'' . Hitherto we have completed analysis for these two schemes.

4.6 Fault-tolerant Magic Square Game

Having prepared the logical EPR pairs fault-tolerantly, we are now ready to play the magic square game. In this section, we will perform an analogous analysis of the failure probability of the magic square game for all three approaches considered in this paper. In addition, we address the main concern on the number of nonlocal and local resources used. We will first state the main result formally. More explicit numerical comparisons based on these bounds will be presented at the end of this section.

Theorem 4.6.1. *Given physical error rate $\epsilon \leq \epsilon'_0 \approx 2.25 \times 10^{-4}$, if we hope to play the Magic Square Game with success probability $1 - \Delta$, $0 < \Delta \ll 1$. Assuming the physical EPR pairs have infidelity 0.1, let χ denote the number of EPR pairs used to achieve this. Then there exists a method that uses an average*

$$\chi \leq 2 \left\lceil \left(\frac{\log(\Delta/9.64c)}{\log \epsilon/c} \right)^2 / 0.54 \right\rceil$$

number of EPR pairs where $c = 1/2129.4$. In particular, when $\epsilon = \epsilon'_0$, we further have a lower bound on χ for this method

$$\chi \geq 2 \left\lceil \left(\frac{\log(\Delta/23.0\mu_0)}{\log \epsilon/\mu_0} \right)^2 / 0.85 \right\rceil,$$

where $\mu_0 = 1/1061.0$.

Next we establish essential elements for the proof of the theorem.

4.6.1 Failure probability

Here we will derive bounds on the failure probability of the magic square game when the encoded EPR pairs are prepared using *Direct Encoding Scheme A*, *Interface+EPP Scheme A* and *B* discussed previously. There is another component in the circuit, that is, the measurement of the corresponding operators which depends on the row/column the party is given. We want to carry out this non-destructive measurement fault-tolerantly. To achieve this, we will adopt Shor's approach of non-destructive measurement, utilizing fault-tolerantly prepared cat-states, as discussed in Chapter 3 Section 3.2.3. However, as they will measure operators such as $\bar{X} \otimes \bar{Z}$, to allow subsequent measurements, a 'big' cat

state, consisting of 14 qubits (when $k = 1$) is needed (Of course, other methods of Shor measurement that save ancillary qubits or use fewer rounds of syndrome measurements can be applied. Since this measurement procedure is the same for all three methods discussed, we will choose the original method and it will serve our purpose). Note that we want Alice and Bob to win the game with probability Δ . We just need to ensure that the row/column assignments⁵ that potentially induce the greatest failure probability is less than Δ , that is when they are both assigned indices 2. A circuit for the FT magic square game is shown below,

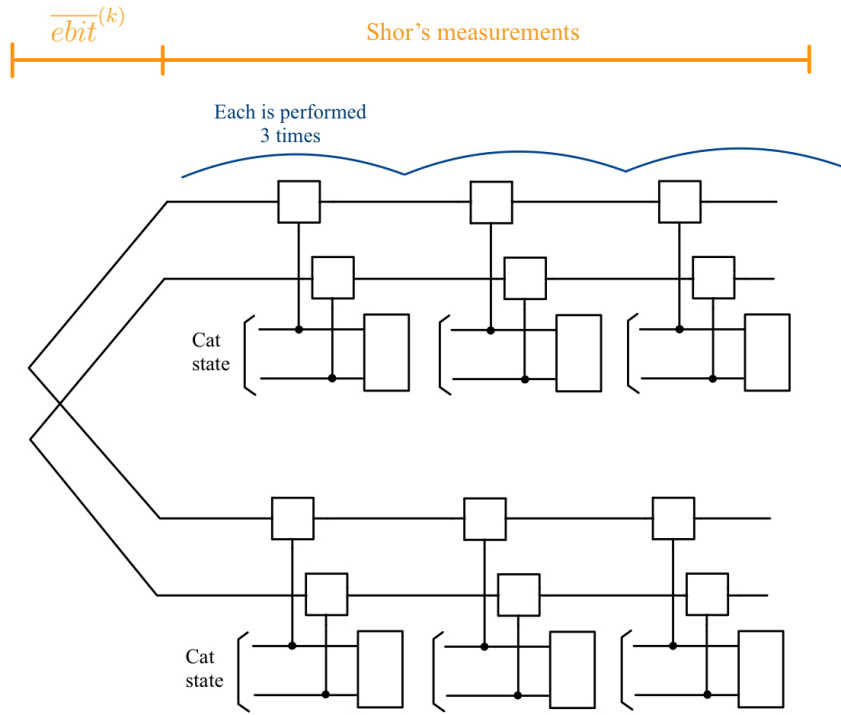


Figure 4.12: Circuit for FT magic square game

In Appendix D.2 we compute bounds on one full round of Shor's measurement. Thus the upper bound for the failure probability of the magic square game follows from the union bound

$$\mathbb{P}(\text{Magic square game fails}) \leq 2 \cdot \mathbb{P}(\overline{ebit}^{(k)} \text{ bad}) + 6\nu_{\text{Shor}}^{(k)}.$$

⁵Recall Table 4.1

Similarly, an approximate lower bound can be obtained, with negligible higher-order terms

$$\mathbb{P}(\text{Magic square game fails}) \geq 2 \cdot \mathbb{P}(\overline{\text{ebit}}^{(k)} \text{ bad}) + 6\mu_{\text{Shor}}^{(k)}.$$

Thus for the three methods, after simplification, we have the bounds for $k \geq 2$ and $\epsilon \leq \epsilon'_0$.

	Upper bound	Lower bound
<i>Direct Encoding</i>	$9.42\epsilon_0 (\epsilon/\epsilon_0)^{2^k}$	$21.6\mu_0 (\epsilon/\mu_0)^{2^k}$
<i>Interface+EPP-A</i>	$9.61\epsilon_0 (\epsilon/\epsilon_0)^{2^k}$	$23.0\mu_0 (\epsilon/\mu_0)^{2^k}$
<i>Interface+EPP-B</i>	$9.64\epsilon_0 (\epsilon/\epsilon_0)^{2^k}$	$23.0\mu_0 (\epsilon/\mu_0)^{2^k}$

Table 4.2: Bounds for the failure probability of the magic square game under different $\overline{\text{ebit}}$ preparation methods at concatenation level k .

Note that in the last two cases, we also have another parameter l , the number of logical EPP performed. Thus the above represents the lowest upper bounds and highest lower bounds achievable with k -th level encoding with our analytical methods. Therefore if we hope to play the magic square game with a success probability of at least $1 - \Delta$, let us denote k_0 to be the minimal number of concatenation levels that could achieve this, we can bound k_0 by

$$\left\lfloor \log_2 \left(\frac{\log \Delta / c_i \mu_0}{\log \epsilon / \mu_0} \right) \right\rfloor \leq k_0 \leq \left\lceil \log_2 \left(\frac{\log \Delta / d_i \epsilon_0}{\log \epsilon / \epsilon_0} \right) \right\rceil,$$

where c_i and d_i correspond to the leading constants of the lower and upper bounds for each method.

4.6.2 Nonlocal resources

Here we first compare the non-local resources(noisy ebits) required to prepare one $\overline{\text{ebit}}^{(k)}$ for each method. Since the magic square game requires Alice and Bob to share 2 ebits, we would multiply by 2. Note that before encoding to logical qubits, we would first perform rounds of physical EPP as in Section 4.2. As argued earlier, for *Direct Encoding* we would need to perform at least two rounds to ensure that the initial error is suppressed enough.

Let m denote the number of rounds of initial EPPs. Thus for *Direct Encoding*, $m \geq 2$. Given the local gates also have an error rate $\epsilon \leq \epsilon'_0$, it's rather hard to work with an analytical form, thus we may resort to numerical tools and set the local physical error rate to be $\epsilon'_0 = 2.25 \times 10^{-4}$ in our subsequent analysis. Assuming the infidelity of the initial EPR pair is 10%, the success probability of the first round of EPP is 59.6% and the second is 95.6%. If we denote the number of EPR pairs required to prepare a $\overline{\text{ebit}}^{(k)}$ by $\chi^{(k)}$, for *Direct Encoding* we have $\forall k \geq 2$,

$$\chi_1^{(k)} = \frac{5}{0.596} \cdot \frac{5}{0.956} \cdot 7^k \approx 43.9 \cdot 7^k.$$

This is actually the exact number of noisy ebits we need in this case since transversal CNOT in the Steane code is the logical CNOT. For the *Interface+EPP* methods, a more meticulous treatment is needed due to the extra parameter l', l'' which are determined by both ϵ and the initial infidelity. We will first obtain lower bounds on the total acceptance probability for the *Interface+EPP* methods. For Scheme A, we have the lower bound,

$$\begin{aligned} & \mathbb{P}(\text{All } \overline{\text{EPP}} \text{ accepted}) \\ & \geq (1 - f_2(\epsilon, \sigma_6)) \prod_{l=2}^{l'} \left(1 - 4g_A^{(l-1)}(k, \sigma_6) - 12\nu_5^{(k)} \right) \\ & \geq (1 - f_2(\epsilon, \sigma_6)) \left(1 - \frac{2}{3} \frac{1}{1/(6\kappa_2(2, \sigma_6)\epsilon^2) - 1} - 12l'\nu_5^{(k)} \right). \end{aligned}$$

For Scheme B it's similar, except when $l \leq k$ we may replace the last term above by $\nu_5^{(l)}$. For the upper bound we will use $1 - f_1(\epsilon, \sigma_6)$ in both cases. According to analysis in [4.5.4](#), we obtain the following upper and lower bounds on χ .

Upper bound on χ for Interface+EPP

To compute the best bounds on the resource overheads, we need to first determine the relationships between the parameters k, m and l', l'' . Note that as we have $m = 2$ for *Direct Encoding*, in the following we will discuss results for $m \leq 2$. In the following table, we show the values of l' for Interface+EPP scheme A and B as in Lemma [4.5.3](#) for given k and m .

		A	B
$k = 2$	$m = 0$	$l'_A = 4$	$l'_B = 4$
	1	2	3
	2	2	2
$k = 3$		A	B
	$m = 0$	$l'_A = 5$	$l'_B = 4$
	1	3	3
	2	2	3
$k = 4, 5$		A	B
	$m = 0$	$l'_A = k + 1$	$l'_B = k + 1$
	1	$k - 1$	k
	2	$k - 1$	k
$k \geq 6$		A	B
	$m = 0$	$l'_A = k + 1$	$l'_B = k$
	1	$k - 1$	k
	2	$k - 2$	k

Table 4.3: Results for l'_A, l'_B as in Lemma 4.5.3 for given k and m

Combining the above results we can see that when $k = 2$, the minimal upper bound on $\chi_2^{(2)}$ can be achieved with Scheme A and $m = 0$. Using this, an upper bound on χ is 484.2, representing over 77% improvement over *Direct Encoding*. For $k \geq 6$, the optimal scheme would be Scheme B with $m = 0$ and the corresponding upper bound on χ is

$$\chi \leq \lceil 4^k / 0.54 \rceil$$

Lower bound on χ for Interface+EPP

For the lower bound on χ we can obtain a similar table when $\epsilon = \epsilon'_0$

$k = 2$		A	B
	$m = 0$	$l''_A = 3$	$l''_B = 3$
	1	2	2
$k = 3$	2	1	2
		A	B
	$m = 0$	$l''_A = 3$	$l''_B = 4$
$k \geq 4$	1	2	3
	2	2	3
		A	B
$k \geq 4$	$m = 0$	$l''_A = k$	$l''_B = k$
	1	$k - 1$	k
	2	$k - 2$	k

Table 4.4: Results for l''_A, l''_B as in Lemma 4.5.3 for given k and m

For the lower bounds corresponding to methods discussed in *Upper bounds*, when $k = 2, m = 0$ with Scheme A , on average a lower bound on $\chi_2^{(2)}$ is 74.9. For $k \geq 6$ (actually works for $k \geq 4$), $m = 0$ and Scheme B gives

$$\chi_2^{(k)} \geq \lfloor 4^k / 0.85 \rfloor$$

These bounds are sufficient to establish our theorem. By contrast, if we hope to win the magic square game with probability $1 - \Delta$ with *Direct Encoding*, we would have the bounds

$$87.7 \cdot 7^{k_1} \leq \chi \leq 87.7 \cdot 7^{k_2}$$

where $k_1 = \left\lceil \log_2 \left(\frac{\log \Delta / d_i \epsilon_0}{\log \epsilon / \epsilon_0} \right) \right\rceil$ and $k_2 = \left\lceil \log_2 \left(\frac{\log \Delta / c_i \mu_0}{\log \epsilon / \mu_0} \right) \right\rceil$. A more explicit numerical comparison will be described in detail later.

4.6.3 Space-time overheads

One potential drawback of *Interface+EPP* seems to be the use of more local resources. However, we claim that while our method achieves exponential savings on nonlocal resources, the space-time overhead grows only linearly in k . Below we will provide estimates and compare the total overhead used by the two parties. Here we first state the assumptions:

1. Qubits can be recycled and reset relatively easily and quickly.
2. Qubits can undergo parallel operations, with each qubit engaged in a single operation simultaneously.
3. Qubits have coherence time long enough to allow consecutive EPPs.

For *Direct Encoding*, to prepare one logical EPR pair, we have 2 code blocks and an additional 2 for error correction. Observing the structure for *Interface+EPP*, the largest overhead comes from the logical EPP part, which consists of 8 data blocks and 8 ECs. Accounting for the overall EPP acceptance probability, *Interface+EPP* uses at most 7.5 times more qubits for all k .

Now we estimate the time overhead for both methods. Let N denote the time-overhead, we will first consider N for preparing $|\bar{0}\rangle^{(k)}$. Since transversal gates can be implemented in parallel and accounting for ancilla rejection probability, we have $N_0^{(k)} = a^{k-1} + 7 \cdot \frac{a^k - 1}{a - 1}$ where $a = 1.0024$. For $k \leq 50$, $N_0^{(k)} \approx 7k + 1$. By examining the Steane EC, we have $N_{EC}^{(k)} = 2N_0^{(k)} + 4$. Thus for *Direct Encoding*, $N_{DE}^{(k)} = 2N_{EC}^{(k)} + N_0^{(k)} + 1 \approx 35k + 14$. For *Interface+EPP-A*, note that preparation of $|\Omega\rangle^{(k)}$ can be done in parallel. Thus the interface uses approximately $(N_0^{(k)} + 11) + \sum_{l=1}^k N_{EC}^{(l)} + 2k$ timesteps. The EPP procedure (regardless of rejection) uses $3N_{EC}^{(k)} + 2$ timesteps. Therefore given $m = 0$ and $k \leq 50$ accounting for rejection, the average time overhead after l' rounds of EPP is $1.89l'_A(7k^2 + 50k + 26) = 1.93(k + 1)(7k^2 + 50k + 26)$. For Scheme *B*, since we have interface-EPP alternately to encode to level- k , the time overhead is $1.89 \sum_{l=1}^k (N_0^{(l)} + 13 + 3N_{EC}^{(l)} + 2) = 0.945(49k^2 + 115k)$. Thus Scheme *B* outperforms Scheme *A* in terms of time-overhead and it is only a linear increase in k compared to *Direct Encoding*. Together with the analysis on space-overhead, our claim is confirmed.

4.6.4 Numerical results

Firstly, we will provide a full stabilizer circuit Monte Carlo simulation comparing the two methods for $k = 1$. Numerically we determined the pseudo-threshold for CNOT-exRec, the largest gadget in the circuit, to be 4.90×10^{-4} . As argued before, to guarantee fault-tolerance, we would need the gate-teleported CNOT to have an error rate below this value. Therefore for a fair comparison, we restrict to $\epsilon \leq 2.42 \times 10^{-4}$ (although Interface+EPP tolerates higher error rates). Since the final basis measurements in FT magic square game are the same regardless of $\overline{\text{ebit}}^{(1)}$ preparation, we will only present the logical error rate against ϵ for preparing $\overline{\text{ebit}}^{(1)}$. We describe the two specific methods we are comparing

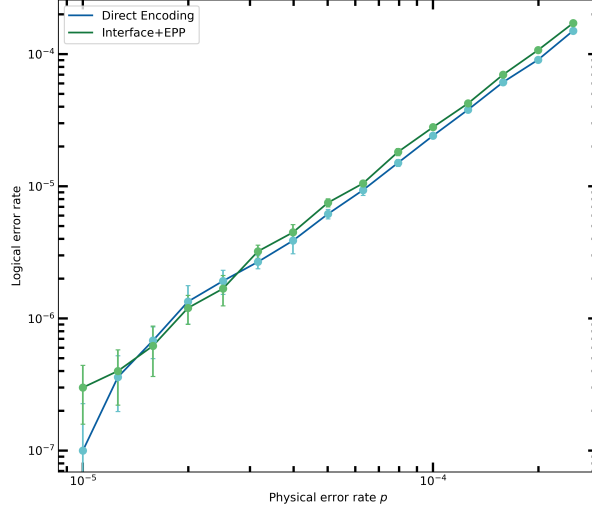
1. For Direct Encoding, we perform two rounds of 5-to-1 physical EPP followed by the gate-teleported encoding.
2. For Interface+EPP, at $k = 1$, the two schemes are the same. For this method, while it can operate with $m = 0$ or $m = 1$ and save more initial ebits, we will present an optimal method ⁶, that is, performing two rounds of 4-to-1 EPP ⁷ followed by one round of Interface+ $\overline{\text{EPP}}^{(1)}$.

We note that in Direct Encoding we use the 5-to-1 EPP instead of 4-to-1 because 5-to-1 is more powerful and can reduce the infidelity to below the threshold value in 2 rounds of initial EPP, whereas the 4-to-1 EPP takes 3 rounds. Thus using 5-to-1 EPP will save ebits.

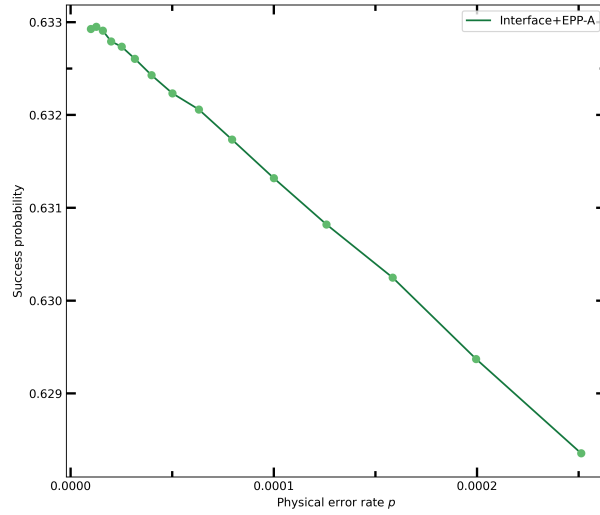
In Figure 5.1, we show the results. Figure 4.13a shows the logical error rates of $\overline{\text{ebit}}^{(1)}$ resulting from the two preparation methods. Figure 4.13b demonstrates the overall success probability for Interface+EPP. We run 10^6 rounds for each value of ϵ and 5 iterations. Multiprocessing was used to maximize performance. The mean and standard deviation are also exhibited by the error bars.

⁶Optimal in the sense that, the logical error rate of the resulting $\overline{\text{ebit}}^{(1)}$ is comparable to Direct Encoding while the ebit overhead and local space-time overhead are suitably minimized.

⁷This 4-to-1 EPP is the same as in Figure 4.5 at the physical level.



(a)



(b)

Figure 4.13: (a) Logical error rates of $\overline{\text{ebit}}^{(1)}$ prepared with *Direct Encoding* and *Interface+EPP* against physical error rate p . Both methods yield similar logical error rates, thereby justifying a subsequent comparison of resource overheads. (b) Overall success probability against ϵ for *Interface+EPP*.

From the first figure, we observe that both methods exhibit similar effectiveness in reducing the logical error rate. As shown in Figure 4.13a, even accounting for statistical error, *Direct Encoding* performs only slightly better than *Interface+EPP*. Since the logical error rates are comparable, it is meaningful to compare the ebit overheads. In Figure 4.13b, we show the overall success probability of the *Interface+EPP* method against ϵ . Now we consider the average number of ebits needed at $\epsilon = 2.42 \times 10^{-4}$. For *Direct Encoding*, we use $\frac{5^2 \times 7}{0.57} \approx 301.7$ ebits. For *Interface+EPP*, we use $\frac{4^3}{0.63} \approx 101.6$ ebits, representing a 66.3% saving relative to *Direct Encoding*. This provides numerical evidence for our claims and demonstrates the advantage of our schemes even for near-term implementation.

Finally, we present results for the Magic Square Game. In Figure 4.14, by using the bounds obtained previously, we compare the two methods in terms of the average number of raw ebits required if we hope to win the magic square game with probability $1 - \Delta$. In the following, for *Interface+EPP*, we plot for Scheme B, which we proved to have better performance.

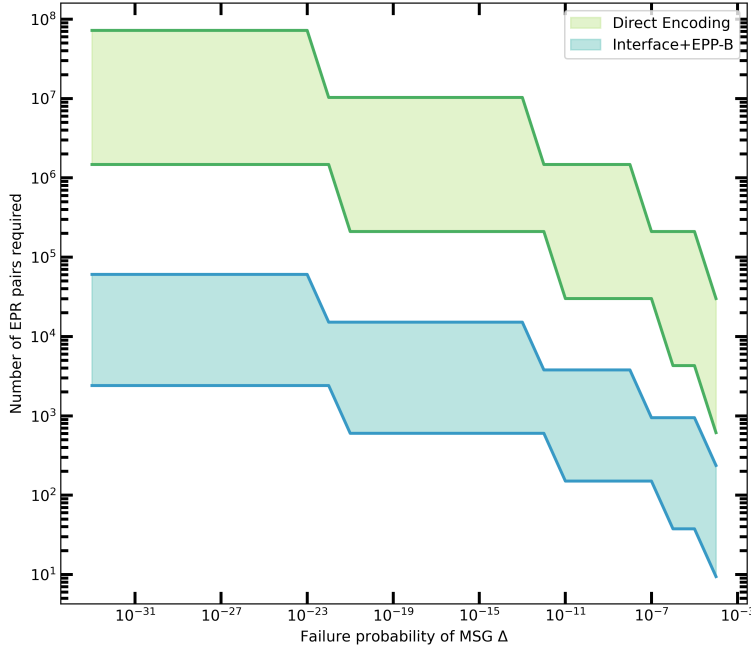


Figure 4.14: The number of raw ebits required against failure probability of the magic square game Δ . Log-scale is used for both axes.

From the above figure, we can conclude that *Interface+EPP-B* is ‘definitely’ more superior than Direct Encoding for $\Delta \leq 10^{-3}$. As log-scale is used, the advantage of *Interface+EPP-B* increases significantly as Δ decreases further. Moreover, by comparing the lower bound of *Direct-Encoding* and the upper bound of *Interface+EPP-B*, we can see that for $10^{-33} \leq \Delta \leq 10^{-3}$, the least improvement is at least 61.4%. As Δ decreases, the improvement reaches at least 95.9% for $\Delta \leq 10^{-23}$. The disparity potentially becomes even more evident as Δ decreases further.

Chapter 5

Extension to the Surface Code

In this chapter, we extend the methodologies in Chapter 4 to the surface code. We will first review the current methods of preparing $\overline{\text{ebit}}$. Next, we describe a potentially feasible method for extending the Interface+EPP method to the surface code, specifically creating $\overline{\text{ebits}}$ by encoding qubits into surface code of length L . To do so, we first review a corresponding interface introduced by Mazurek et al. [66] combined with lattice-surgery. Then as an easier step, we discuss how to perform EPP efficiently with these logical qubits. Note that no theoretical analysis of logical error rate has been done due to the hardness of lattice-surgery when it comes to analytics. However, numerical simulation (in progress) under phenomenological noise has shown promising results.

5.1 Direct Encoding

Now we introduce the current conceived ways of preparing $\overline{\text{ebit}}$ with surface code. The most naive approach is through transversal CNOT as the surface code is a CSS code. The CNOT, again, comes from gate-teleportation from an ebit as in Figure 4.3. An alternative way is through lattice surgery, as described by Leone et al. [59]. Lattice surgery was described in Chapter 3 Section 3.3.4. It consists of merging and splitting, where rough merging followed by splitting is equivalent to performing a $M_{\overline{XX}}$ -measurement. In order to prepare $\overline{\text{ebit}}$, we note that when two surfaces are initialized in $|\overline{0}\rangle$ and a $M_{\overline{XX}}$ -measurement is performed, this produces an $\overline{\text{ebit}}$. To perform lattice surgery, measurements of stabilizers are needed and a column of ancilla qubits is needed. Alice has her code patch and her part of measurement ancillas along with merging ancillas. Bob connects his measurement ancillas

to Alice’s merging ancillas using CNOT gates to perform the stabilizer measurements required for the lattice surgery. The connecting CNOTs are implemented again via gate teleportation with good ebits. Here initial rounds of EPP are also needed to bring the infidelity of ebits down to some threshold value. Surprisingly, the threshold for the infidelity of ebits is fairly high for this lattice surgery procedure. The gate-teleported CNOT can have a threshold up to 8.54% as shown by Ramette et al. [74]. This procedure is shown in Figure 5.1b.

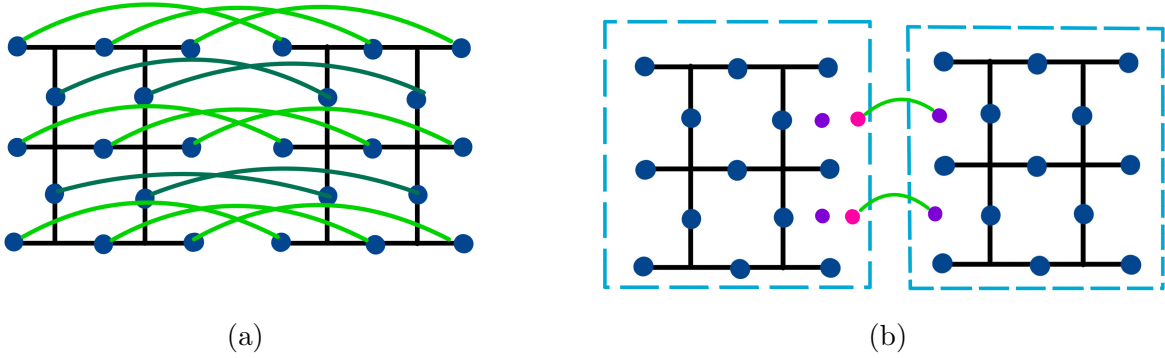


Figure 5.1: (a) Example of a transversal implementation of an $\overline{\text{ebit}}$. (b) Example of a distant surface code $\overline{\text{ebit}}$ with lattice surgery. Alice and Bob (at a distance) each prepare $|\overline{0}\rangle$. Alice holds her code patch and all ancilla qubits for lattice surgery and stabilizer measurements. The pink dots are the ancilla qubits for lattice surgery and the purple dots are ancilla qubits for measurements. To perform the X_v measurements, Bob will need to connect to pink qubits via long-distance CNOTs, which are implemented via gate teleportation with a high-fidelity ebit.

For the ebit overhead, it suffices to estimate the number of non-local CNOTs required because in both schemes ebits are consumed to implement these CNOTs. Given the surface code’s robustness with a high FT threshold of up to 1% as shown by Wang et al. [91], fewer rounds of initial physical EPP would be needed. For the first scheme, L^2 non-local CNOTs will be necessary. For the second, there are $L - 1$ distant CNOTs. However, we note that to ensure FT, we need to perform at most L -rounds of measurements. Thus in the worst case, this method uses $L(L - 1)$ CNOTs. In practice, which of these two methods is preferable depends on a number of factors. The most evident difference between is probably the number of local qubits the distant CNOTs interact with. The distant links are most likely in photons while local qubits can be in various platforms. Therefore the

performance will highly depend upon the efficiency of the interface between photons and matters, e.g. trapped-ions [67] and atom arrays [62].

5.2 Interface

We now present the Interface+EPP method within the context of the surface code. To begin with, we construct a valid interface that encodes an unknown quantum state into a surface code with a code distance L . This procedure can be divided into two distinct parts. The first part involves creating an interface that maps a single qubit to a surface code patch with a distance L' , $L' \leq L$. The second part is an interface that transfers a logical state from $|\psi\rangle_{L'}$ to $|\psi\rangle_L$ ¹.

5.2.1 Part 1: $|\psi\rangle \rightarrow |\psi\rangle_{L'}$

The following idea is presented in the work by Mazurek et al. [66]. We will review it here and draw a connection with the interface used for the Steane code. Recall that in Chapter 4 Figure 4.4 we have an interface circuit for the Steane code, using teleportation. Here we use a similar idea. We note that teleportation can be performed using measurements only. Assume Alice possesses qubit 1, which contains the state to be teleported, and qubit 2, which is one part of an ebit. Bob holds the other part of the ebit, qubit 3. The procedure begins by measuring X_1X_2 . If the measurement result is -1, the operation Z_2Z_3 is applied. Subsequently, Z_1Z_2 is measured, and if the outcome is -1, the operation X_2X_3 is executed. This sequence of measurements and conditional operations completes the teleportation process. On the surface code, we also need to identify 'qubits' 1, 2 and 3. The choices are shown in Figure 5.2, with the corresponding X, Z -operators acting on them, where the qubit 1 to be teleported is on the left-bottom corner.

¹In the following we denote $|\psi\rangle_L$ instead as a logical state encoded in a surface code of distance L .

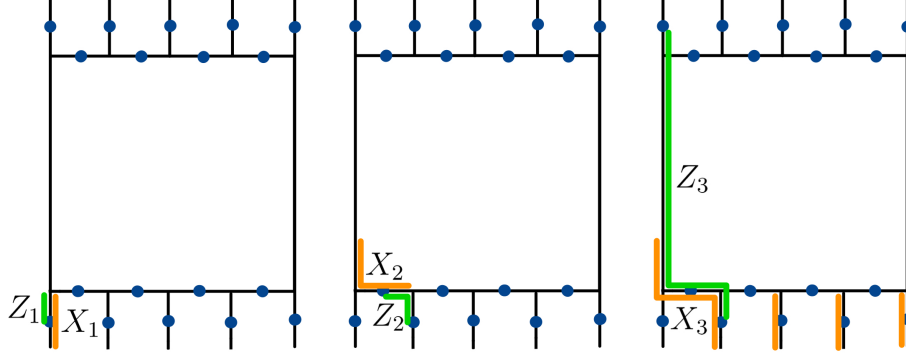


Figure 5.2

Surely, to start doing any computation with the surface code, we need to initialize it by measuring all stabilizers (in this case, everyone not touching qubit 1) and correcting errors. We can see that, after this, the code patch (excluding qubit 1) is in the common eigenspace of X_2X_3 and Z_2Z_3 , thus in some sense, 'maximally entangled'. We then measure stabilizers that include qubit 1, X_1X_2 and Z_1Z_2 . If the measurement outcome of X_1X_2 is -1, we apply Z_2Z_3 , which is the left-most vertical excluding qubit 1; if the measurement outcome of Z_1Z_2 is -1 we apply X_2X_3 , which is the lowest X_L excluding qubit 1. Due to the correspondence to teleportation and the fact that X_1X_2 and Z_1Z_2 are exactly stabilizers of the surface code, we see that this procedure indeed teleports the state in qubit 1 to the whole code patch.

5.2.2 Part 2 $|\psi\rangle_{L'} \rightarrow |\psi\rangle_L$

Technically we can use the same procedure for code patches of arbitrarily large distances, however, an error on qubit 1 will translate to a logical error on the code patch, as in the Steane code case. Therefore, to achieve fault-tolerance, either we perform initial EPP to reduce the infidelity below the threshold and then perform the above encoding, or we can encode while purifying. To do so, we would need a way of transferring logical states between code patches of different sizes. This can be done via lattice surgery. To do so, we will first present the above measurement-based teleportation more explicitly, including the generation of the ebit, as in Figure 5.3.

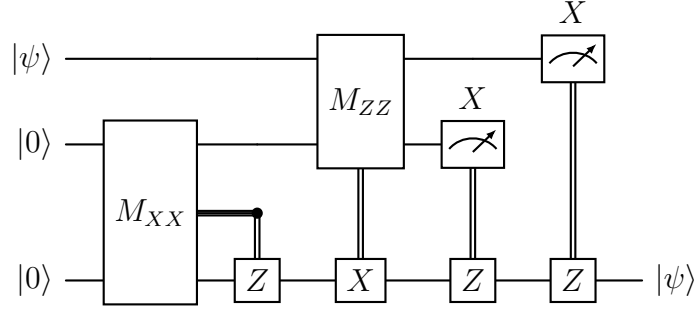


Figure 5.3: Measurement-based teleportation

Now we can use this circuit to achieve teleportation between two surface codes with different distances L', L . For the purposes of this discussion, we assume $L' \leq L$. Initially, we have a code patch with a distance L' in state $|\psi\rangle_{L'}$, and another code patch with distance L in state $|0\rangle$, which is the target logical qubit for the teleportation process. Now we still need a logical ancilla qubit 2. We will prepare it in logical- $|0\rangle$ state with a code patch of dimension $L' \times L^2$. In this way, we can perform all the above operations.

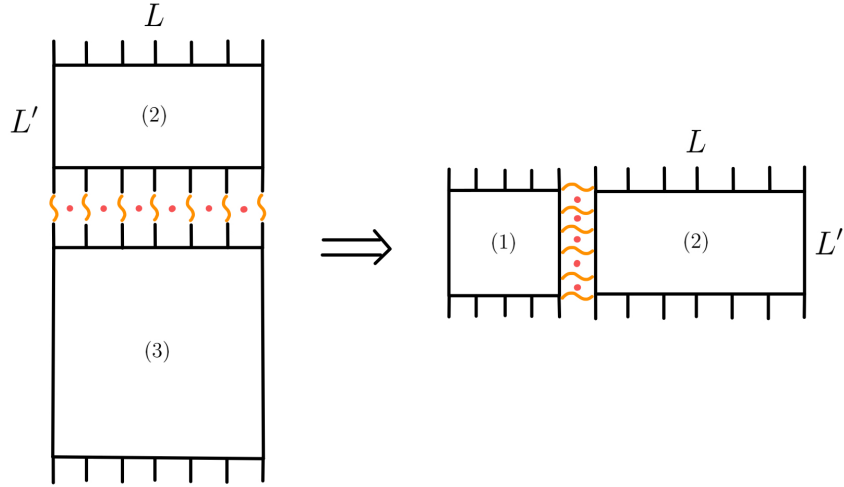


Figure 5.4: Lattice-surgery based teleportation

²The rectangular surface code initialization will be the same as the square case.

As shown in Figure 5.4, we first use rough merging on logical qubits (2) and (3) to measure $\overline{X}_2\overline{X}_3$ followed by error correction (and possibly postselection if we want to prepare a good ebit as in Figure 4.3). Then $\overline{Z}_1\overline{Z}_2$ is measured via a smooth merging between logical qubits (1) and (2). At the end, we measure $\overline{X}_1$ and $\overline{X}_2$ respectively to infer the correction on logical qubit 3, completing the teleportation.

In the above, we discuss a way of interfacing one qubit with a surface code of distance L . Assuming that Alice and Bob have collectively prepared several ebits utilizing this interface, they can subsequently execute $\overline{\text{EPP}}$ by locally performing CNOT operations, Hadamard gates, and measurements. These operations can be carried out either using transversal gates or through lattice-surgery.

5.3 Discussion

Analytical results are challenging to derive in this context, especially when aiming to achieve more affirmative comparisons that require tighter bounds. However, with numerical simulations, we have obtained evidence that with an optimized choice of L' and L in Interface+EPP, fewer ebits are needed to achieve a similar logical error rate compared to the second Direct Encoding scheme (Definitive conclusions need further validation, which is pending).

In fact, the method in the second part (and more generally lattice-surgery) of the interface has more versatile uses. With some modifications, it can be made as the interface between different families of code. It was shown by Nautrup et al. [72, 83] that a logical state can be teleported from a color code to a surface code of the same edge length. Building upon this, and integrating with prior discussions, it is possible to achieve teleportation between color codes and surface codes of different edge lengths. This consideration is potentially important in scenarios where quantum repeaters, which transmit information over long distances [77, 93], use one type of QECC, while the end stations (quantum computers) are employed using a different QECC.

Chapter 6

Conclusion

In this thesis, we investigate the preparation of long-range $\overline{\text{ebit}}$ in the context of Magic Square Game. Specifically, we aim to minimize the number of ebits consumed while ensuring that the failure probability does not exceed Δ .

To this end, we constructed a novel way by leveraging an ‘interface’, which translates an unknown state to the logical space, and the purifying power of EPP. For the $[[7^k, 1, 3^k]]$ concatenated Steane code, our analytical bounds indicate that *Interface+EPP* scheme exhibits remarkable advantage over the conventional method (encoding qubits on both sides and performing transversal gates), in terms of the number of initial ebits, with this discrepancy becoming more significant as Δ decreases. Moreover, when $k = 1$, we did a full-circuit simulation to support our claims. We also attempted to extend this interface to the surface code. Lattice-surgery is particularly useful in this context. However, a more comprehensive comparison of Direct Encoding and Interface+EPP is still part of my ongoing work. On the non-local game side, our analysis provides an upper bound on the number of ebits needed to play the magic square game fault-tolerantly with the Steane code. Tighter upper bounds and a lower bound could be future research directions. It is not clear yet how a lower bound could be obtained.

Broadly speaking, our proposal presents a flexible and comprehensive framework for similar tasks. Many aspects of our construction can be adapted to suit different needs. For example, in Figure 4.5, instead of destructive measurements for $\overline{\text{EPP}}$, Alice and Bob can further save local qubits as well as initial ebits by performing non-destructive $\overline{X}/\overline{Z}$ measurements. The last three $\overline{\text{ebits}}$ can then be used for subsequent $\overline{\text{EPP}}$. However, the actual implementation of this will be limited by qubit decoherence time T_1/T_2 , thus some platforms are preferable to others [65]. In addition, the EPP used can also be altered. In

this thesis, we mainly used the 4-to-1 for $\overline{\text{EPP}}$ that is capable of correcting both $\overline{X}$ - and $\overline{Z}$ -errors. In fact, not only can we use more powerful EPP by using more ebits, but in some special cases, we can actually reduce the ebits needed. For example, in systems with biased noise [89, 95], it might suffice to use two $\overline{\text{ebit}}$ to purify X - or Z -noise.

Certainly, this framework can be generalized to other QECCs as long as a valid interface is constructed. In terms of local overhead saving, one can employ concatenation of quantum Hamming codes and develop a suitable interface [96]. However, one difficulty remains for this method is that when applying transversal CNOT locally, it's impossible to act on individual logical qubits, instead, the transversal CNOT needs to act on blocks of logical qubits simultaneously. It's also possible to extend to qLDPC codes. Since lattice-surgery has been adapted to qLDPC code to implement fault-tolerant gate set [27], an analogous interface could be developed for certain qLDPC codes. If we can adapt to other QECCs, various techniques developed for those codes can be used. For example, single-shot error correction can be used to avoid the need for repeated measurements [14, 40]. If a higher-distance code is used, it's also possible to use adaptive syndrome measurement to reduce the number of rounds of measurements[86][29].

Practically, our protocol for the Steane code requires all-to-all connectivity, thus experimental verification is feasible for certain platforms such as neutral atoms [12] and trapped ions [78]. On the other hand, in superconducting, where nearest-neighbour interactions are preferable, our proposal for the surface code is readily implementable. Moreover, as mentioned in Chapter 5 it is possible to interface color code and surface code of different sizes. Our framework also allows interfacing between different codes without significantly compromising the logical error rate. We hope our discussion will inspire further research in this area, which is crucial not only for FT nonlocal games but also for the development of modular quantum architecture and the quantum internet.

References

- [1] *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 452(1954):2551–2577, November 1996.
- [2] Dorit Aharonov and Michael Ben-Or. Fault tolerant quantum computation with constant error, 1996.
- [3] Panos Aliferis, Daniel Gottesman, and John Preskill. Quantum accuracy threshold for concatenated distance-3 codes, 2005.
- [4] Panos Aliferis, Daniel Gottesman, and John Preskill. Accuracy threshold for postselected quantum computation. 2007.
- [5] Jonas T. Anderson, Guillaume Duclos-Cianci, and David Poulin. Fault-tolerant conversion between the steane and reed-muller quantum codes. *Physical Review Letters*, 113(8), August 2014.
- [6] Alain Aspect, Philippe Grangier, and Gérard Roger. Experimental Tests of Realistic Local Theories via Bell’s Theorem. , 47(7):460–463, August 1981.
- [7] Koji Azuma, Sophia E. Economou, David Elkouss, Paul Hilaire, Liang Jiang, Hoi-Kwong Lo, and Ilan Tzitrin. Quantum repeaters: From quantum networks to the quantum internet. *Reviews of Modern Physics*, 95(4), December 2023.
- [8] Sara Bartolucci, Patrick Birchall, Hector Bombin, Hugo Cable, Chris Dawson, Mercedes Gimeno-Segovia, Eric Johnston, Konrad Kieling, Naomi Nickerson, Mihir Pant, Fernando Pastawski, Terry Rudolph, and Chris Sparrow. Fusion-based quantum computation, 2021.
- [9] Paula Belzig, Matthias Christandl, and Alexander Müller-Hermes. Fault-tolerant coding for entanglement-assisted communication. *IEEE Transactions on Information Theory*, 70(4):2655–2673, April 2024.

- [10] 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, Jan 1996.
- [11] Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed-state entanglement and quantum error correction. *Physical Review A*, 54(5):3824–3851, November 1996.
- [12] Dolev Bluvstein, Simon J. Evered, Alexandra A. Geim, Sophie H. Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, J. Pablo Bonilla Ataides, Nishad Maskara, Iris Cong, Xun Gao, Pedro Sales Rodriguez, Thomas Karolyshyn, Giulia Semeghini, Michael J. Gullans, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. Logical quantum processor based on reconfigurable atom arrays. *Nature*, December 2023.
- [13] Dolev Bluvstein, Harry Levine, Giulia Semeghini, Tout T. Wang, Sepehr Ebadi, Marcin Kalinowski, Alexander Keesling, Nishad Maskara, Hannes Pichler, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. A quantum processor based on coherent transport of entangled atom arrays. *Nature*, 604(7906):451–456, April 2022.
- [14] Héctor Bombín. Single-shot fault-tolerant quantum error correction. *Physical Review X*, 5(3), September 2015.
- [15] Gilles Brassard, Anne Broadbent, and Alain Tapp. Quantum pseudo-telepathy. *Foundations of Physics*, 35(11):1877–1907, November 2005.
- [16] Sergey Bravyi, David Gosset, Robert König, and Marco Tomamichel. Quantum advantage with noisy shallow circuits. *Nature Physics*, 16(10):1040–1045, July 2020.
- [17] Sergey Bravyi, David Poulin, and Barbara Terhal. Tradeoffs for reliable quantum information storage in 2d systems. *Physical Review Letters*, 104(5), February 2010.
- [18] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, January 2009.
- [19] Harry Buhrman, Richard Cleve, Serge Massar, and Ronald de Wolf. Nonlocality and communication complexity. *Reviews of Modern Physics*, 82(1):665–698, March 2010.
- [20] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098–1105, August 1996.

- [21] Christopher Chamberland, Tomas Jochym-O'Connor, and Raymond Laflamme. Overhead analysis of universal concatenated quantum codes. *Phys. Rev. A*, 95:022313, Feb 2017.
- [22] Rui Chao and Ben W. Reichardt. Quantum error correction with only two extra qubits. *Phys. Rev. Lett.*, 121:050502, Aug 2018.
- [23] Rui Chao and Ben W. Reichardt. Quantum error correction with only two extra qubits. *Physical Review Letters*, 121(5), aug 2018.
- [24] Rui Chao and Ben W. Reichardt. Flag fault-tolerant error correction for any stabilizer code. *PRX Quantum*, 1(1), September 2020.
- [25] Kevin S. Chou, Jacob Z. Blumoff, Christopher S. Wang, Philip C. Reinhold, Christopher J. Axline, Yvonne Y. Gao, L. Frunzio, M. H. Devoret, Liang Jiang, and R. J. Schoelkopf. Deterministic teleportation of a quantum gate between two logical qubits. *Nature*, 561(7723):368–373, September 2018.
- [26] Matthias Christandl and Alexander Müller-Hermes. Fault-tolerant coding for quantum communication, 2022.
- [27] Lawrence Z. Cohen, Isaac H. Kim, Stephen D. Bartlett, and Benjamin J. Brown. Low-overhead fault-tolerant quantum computing using long-range connectivity. *Science Advances*, 8(20), May 2022.
- [28] M. P. da Silva, C. Ryan-Anderson, J. M. Bello-Rivas, A. Chernoguzov, J. M. Dreiling, C. Foltz, F. Frachon, J. P. Gaebler, T. M. Gatterman, L. Grans-Samuelsson, D. Hayes, N. Hewitt, J. Johansen, D. Lucchetti, M. Mills, S. A. Moses, B. Neyenhuis, A. Paz, J. Pino, P. Siegfried, J. Strabley, A. Sundaram, D. Tom, S. J. Wernli, M. Zanner, R. P. Stutz, and K. M. Svore. Demonstration of logical qubits and repeated error correction with better-than-physical error rates, 2024.
- [29] Nicolas Delfosse and Ben W. Reichardt. Short shor-style syndrome sequences, 2020.
- [30] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller. Quantum repeaters based on entanglement purification. *Phys. Rev. A*, 59:169–181, Jan 1999.
- [31] Jack Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics*, page 125, 1965.

- [32] Rajeev Acharya et al. Suppressing quantum errors by scaling a surface code logical qubit, 2022.
- [33] Shai Evra, Tali Kaufman, and Gilles Zémor. Decodable quantum ldpc codes beyond the $\sqrt{n}$ distance barrier using high dimensional expanders, 2020.
- [34] Keisuke Fujii and Katsuji Yamamoto. Entanglement purification with double selection. *Phys. Rev. A*, 80:042308, Oct 2009.
- [35] Marissa Giustina, Marijn A. M. Versteegh, Sören Wengerowsky, Johannes Handsteiner, Armin Hochrainer, Kevin Phelan, Fabian Steinlechner, Johannes Kofler, Jan-Åke Larsson, Carlos Abellán, Waldimar Amaya, Valerio Pruneri, Morgan W. Mitchell, Jörn Beyer, Thomas Gerrits, Adriana E. Lita, Lynden K. Shalm, Sae Woo Nam, Thomas Scheidl, Rupert Ursin, Bernhard Wittmann, and Anton Zeilinger. Significant-loop-hole-free test of bell’s theorem with entangled photons. *Phys. Rev. Lett.*, 115:250401, Dec 2015.
- [36] D. Gottesman. Surviving as a quantum computer in a classical world, 2024.
- [37] Daniel Gottesman. Stabilizer codes and quantum error correction, 1997.
- [38] Daniel Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation, 2009.
- [39] Daniel Gottesman and Isaac L. Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402(6760):390–393, November 1999.
- [40] Shouzheng Gu, Eugene Tang, Libor Caha, Shin Ho Choe, Zhiyang He, and Aleksander Kubica. Single-shot decoding of good quantum ldpc codes, 2023.
- [41] Richard Wesley Hamming. Error-detecting and error-correcting codes. 29(2):147–160, 1950.
- [42] B. Hensen, H. Bernien, A. Dréau, Andreas Reiserer, Norbert Kalb, M. Blok, J. Ruitenberg, R. Vermeulen, R. Schouten, Carlos Abellan, Waldimar Amaya, V. Pruneri, Morgan Mitchell, M. Markham, Daniel Twitchen, David Elkouss, Stephanie Wehner, Tim Taminiau, and R. Hanson. Loop-hole-free bell inequality violation using electron spins separated by 1.3 kilometres. *Nature*, 526, 10 2015.
- [43] Charles D. Hill, Austin G. Fowler, David S. Wang, and Lloyd C. L. Hollenberg. Fault-tolerant quantum error correction code conversion, 2011.

- [44] Dominic Horsman, Austin G Fowler, Simon Devitt, and Rodney Van Meter. Surface code quantum computing by lattice surgery. *New Journal of Physics*, 14(12):123011, December 2012.
- [45] Mathlover (<https://math.stackexchange.com/users/22430/mathlover>). Finding the sum- $x + x^2 + x^4 + x^8 + x^{16} \dots$. Mathematics Stack Exchange, 2015. URL:<https://math.stackexchange.com/q/404521> (version: 2016-07-29).
- [46] Xiao-Min Hu, Cen-Xiao Huang, Yu-Bo Sheng, Lan Zhou, Bi-Heng Liu, Yu Guo, Chao Zhang, Wen-Bo Xing, Yun-Feng Huang, Chuan-Feng Li, and Guang-Can Guo. Long-distance entanglement purification for quantum communication. *Phys. Rev. Lett.*, 126:010503, Jan 2021.
- [47] Richard Jozsa. An introduction to measurement based quantum computation, 2005.
- [48] A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, January 2003.
- [49] E. Knill. Fault-tolerant postselected quantum computation: Schemes, 2004.
- [50] E. Knill. Scalable quantum computation in the presence of large detected-error rates, 2004.
- [51] E. Knill. Quantum computing with realistically noisy devices. *Nature*, 434(7029):39–44, March 2005.
- [52] Emanuel Knill and Raymond Laflamme. Concatenated quantum codes, 1996.
- [53] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Phys. Rev. A*, 55:900–911, Feb 1997.
- [54] Emanuel Knill, Raymond Laflamme, and Wojciech H. Zurek. Resilient quantum computation: error models and thresholds. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):365–384, January 1998.
- [55] Stefan Krastanov, Victor V. Albert, and Liang Jiang. Optimized entanglement purification. *Quantum*, 3:123, February 2019.
- [56] Stefan Krastanov, Victor V. Albert, and Liang Jiang. Optimized Entanglement Purification. *Quantum*, 3:123, February 2019.

- [57] Sebastian Krinner, Nathan Lacroix, Ants Remm, Agustin Di Paolo, Elie Genois, Catherine Leroux, Christoph Hellings, Stefania Lazar, Francois Swiadek, Johannes Herrmann, Graham J. Norris, Christian Kraglund Andersen, Markus Müller, Alexandre Blais, Christopher Eichler, and Andreas Wallraff. Realizing repeated quantum error correction in a distance-three surface code. *Nature*, 605(7911):669–674, May 2022.
- [58] Andrew J. Landahl and Ciaran Ryan-Anderson. Quantum computing by color-code lattice surgery, 2014.
- [59] Hudson Leone, S Srikara, Peter P. Rohde, and Simon Devitt. Upper bounds for the clock speeds of fault-tolerant distributed quantum computation using satellites to supply entangled photon pairs, 2024.
- [60] Debbie Leung. Course on quantum error correction and fault-tolerance. <https://www.math.uwaterloo.ca/~wcleung/qic890-w2024.html>, 2024.
- [61] Anthony Leverrier and Gilles Zémor. Quantum tanner codes, 2022.
- [62] Yiyi Li and Jeff Thompson. High-rate and high-fidelity modular interconnects between neutral atom quantum processors, 2024.
- [63] Daniel Litinski. A game of surface codes: Large-scale quantum computing with lattice surgery. *Quantum*, 3:128, March 2019.
- [64] Yi-Han Luo, Ming-Cheng Chen, Manuel Erhard, Han-Sen Zhong, Dian Wu, Hao-Yang Tang, Qi Zhao, Xi-Lin Wang, Keisuke Fujii, Li Li, Nai-Le Liu, Kae Nemoto, William J. Munro, Chao-Yang Lu, Anton Zeilinger, and Jian-Wei Pan. Quantum teleportation of physical qubits into logical code spaces. *Proceedings of the National Academy of Sciences*, 118(36), September 2021.
- [65] Shayan Majidy, Christopher Wilson, and Raymond Laflamme. *Building Quantum Computers: A Practical Introduction*. Cambridge University Press, 2024.
- [66] Paweł Mazurek, Andrzej Grudka, Michał Horodecki, Paweł Horodecki, Justyna Lodyga, Łukasz Pankowski, and Anna Przysiężna. Long-distance quantum communication over noisy networks without long-time quantum memory. *Physical Review A*, 90(6), December 2014.
- [67] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim. Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. *Physical Review A*, 89(2), February 2014.

- [68] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [69] Pavel Panteleev and Gleb Kalachev. Asymptotically good quantum and locally testable classical ldpc codes, 2022.
- [70] Christophe Piveteau and David Sutter. Circuit knitting with classical communication. *IEEE Transactions on Information Theory*, 70(4):2734–2745, April 2024.
- [71] Lukas Postler, Friederike Butt, Ivan Pogorelov, Christian D. Marciniak, Sascha Heußen, Rainer Blatt, Philipp Schindler, Manuel Rispler, Markus Müller, and Thomas Monz. Demonstration of fault-tolerant steane quantum error correction, 2023.
- [72] Hendrik Poulsen Nautrup, Nicolai Friis, and Hans J. Briegel. Fault-tolerant interface between quantum memories and quantum processors. *Nature Communications*, 8(1), November 2017.
- [73] Benjamin Rahn, Andrew C. Doherty, and Hideo Mabuchi. Exact performance of concatenated quantum codes. *Physical Review A*, 66(3), September 2002.
- [74] Joshua Ramette, Josiah Sinclair, Nikolas P. Breuckmann, and Vladan Vuletić. Fault-tolerant connection of error-corrected qubits with noisy links, 2023.
- [75] Narayanan Rengaswamy, Nithin Raveendran, Ankur Raina, and Bane Vasić. Entanglement purification with quantum ldpc codes and iterative decoding. *Quantum*, 8:1233, January 2024.
- [76] Wenjamin Rosenfeld, Daniel Burchardt, Robert Garthoff, Kai Redeker, Norbert Ortégel, Markus Rau, and Harald Weinfurter. Event-ready bell test using entangled atoms simultaneously closing detection and locality loopholes. *Phys. Rev. Lett.*, 119:010402, Jul 2017.
- [77] Filip Rozpedek, Kyungjoo Noh, Qian Xu, Saikat Guha, and Liang Jiang. Quantum repeaters based on concatenated bosonic and discrete-variable quantum codes. *npj Quantum Information*, 7(1), June 2021.
- [78] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, N. C. Brown, T. M. Gatterman, S. K. Halit, K. Gilmore, J. Gerber, B. Neyenhuis, D. Hayes, and R. P. Stutz. Realization of real-time fault-tolerant quantum error correction, 2021.

- [79] C. Ryan-Anderson, N. C. Brown, M. S. Allman, B. Arkin, G. Asa-Attuah, C. Baldwin, J. Berg, J. G. Bohnet, S. Braxton, N. Burdick, J. P. Campora, A. Chernoguzov, J. Esposito, B. Evans, D. Francois, J. P. Gaebler, T. M. Gatterman, J. Gerber, K. Gilmore, D. Gresh, A. Hall, A. Hankin, J. Hostetter, D. Lucchetti, K. Mayer, J. Myers, B. Neyenhuis, J. Santiago, J. Sedlacek, T. Skripka, A. Slattery, R. P. Stutz, J. Tait, R. Tobey, G. Vittorini, J. Walker, and D. Hayes. Implementing fault-tolerant entangling gates on the five-qubit code and the color code, 2022.
- [80] C. Ryan-Anderson, N. C. Brown, M. S. Allman, B. Arkin, G. Asa-Attuah, C. Baldwin, J. Berg, J. G. Bohnet, S. Braxton, N. Burdick, J. P. Campora, A. Chernoguzov, J. Esposito, B. Evans, D. Francois, J. P. Gaebler, T. M. Gatterman, J. Gerber, K. Gilmore, D. Gresh, A. Hall, A. Hankin, J. Hostetter, D. Lucchetti, K. Mayer, J. Myers, B. Neyenhuis, J. Santiago, J. Sedlacek, T. Skripka, A. Slattery, R. P. Stutz, J. Tait, R. Tobey, G. Vittorini, J. Walker, and D. Hayes. Implementing fault-tolerant entangling gates on the five-qubit code and the color code, 2022.
- [81] C. Ryan-Anderson, N. C. Brown, C. H. Baldwin, J. M. Dreiling, C. Foltz, J. P. Gaebler, T. M. Gatterman, N. Hewitt, C. Holliman, C. V. Horst, J. Johansen, D. Lucchetti, T. Mengle, M. Matheny, Y. Matsuoka, K. Mayer, M. Mills, S. A. Moses, B. Neyenhuis, J. Pino, P. Siegfried, R. P. Stutz, J. Walker, and D. Hayes. High-fidelity and fault-tolerant teleportation of a logical qubit using transversal gates and lattice surgery on a trapped-ion quantum computer, 2024.
- [82] Peter W. Shor. Fault-tolerant quantum computation, 1996.
- [83] Noah Shutty and Christopher Chamberland. Decoding merged color-surface codes and finding fault-tolerant clifford circuits using solvers for satisfiability modulo theories. *Physical Review Applied*, 18(1), July 2022.
- [84] K. M. Svore, A. W. Cross, I. L. Chuang, and A. V. Aho. A flow-map model for analyzing pseudothresholds in fault-tolerant quantum computing, 2006.
- [85] Theerapat Tansuwannont and Debbie Leung. Achieving fault tolerance on capped color codes with few ancillas. *PRX Quantum*, 3(3), August 2022.
- [86] Theerapat Tansuwannont, Balint Pato, and Kenneth R. Brown. Adaptive syndrome measurements for shor-style error correction. *Quantum*, 7:1075, August 2023.
- [87] Joy A. Thomas Thomas M. Cover. *Elements of Information Theory*. John Wiley Sons, Inc, 1991.

- [88] Jean-Pierre Tillich and Gilles Zemor. Quantum ldpc codes with positive rate and minimum distance proportional to the square root of the blocklength. *IEEE Transactions on Information Theory*, 60(2):1193–1202, February 2014.
- [89] David K. Tuckett, Andrew S. Darmawan, Christopher T. Chubb, Sergey Bravyi, Stephen D. Bartlett, and Steven T. Flammia. Tailoring surface codes for highly biased noise. *Physical Review X*, 9(4), November 2019.
- [90] Joel J. Wallman and Joseph Emerson. Noise tailoring for scalable quantum computation via randomized compiling. *Physical Review A*, 94(5), November 2016.
- [91] D. S. Wang, A. G. Fowler, A. M. Stephens, and L. C. L. Hollenberg. Threshold error rates for the toric and surface codes, 2009.
- [92] Adam Winick, Joel J. Wallman, Dar Dahlen, Ian Hincks, Egor Ospadov, and Joseph Emerson. Concepts and conditions for error suppression through randomized compiling, 2022.
- [93] Kah Jen Wo, Guus Avis, Filip Rozpedek, Maria Flors Mor-Ruiz, Gregor Pieplow, Tim Schröder, Liang Jiang, Anders S. Sørensen, and Johannes Borregaard. Resource-efficient fault-tolerant one-way quantum repeater with code concatenation. *npj Quantum Information*, 9(1), December 2023.
- [94] Jia-Min Xu, Yi-Zheng Zhen, Yu-Xiang Yang, Zi-Mo Cheng, Zhi-Cheng Ren, Kai Chen, Xi-Lin Wang, and Hui-Tian Wang. Experimental demonstration of quantum pseudotelepathy. *Phys. Rev. Lett.*, 129:050402, Jul 2022.
- [95] Qian Xu, Nam Mannucci, Alireza Seif, Aleksander Kubica, Steven T. Flammia, and Liang Jiang. Tailored xzzx codes for biased noise, 2022.
- [96] Hayata Yamasaki and Masato Koashi. Time-efficient constant-space-overhead fault-tolerant quantum computation. *Nature Physics*, 20(2):247–253, January 2024.
- [97] Xinlan Zhou, Debbie W. Leung, and Isaac L. Chuang. Methodology for quantum logic gate construction. *Physical Review A*, 62(5), October 2000.

Appendix A

Bounds on $\mathbb{P}(\text{exRec is bad})$

A.1 Bound on third-order terms

In this section, we provide proof for Lemma 4.3.1. We would convert this to a graph theory problem. To prove the theorem, we will first consider the case where there are two types of locations and then generalize to n locations. Let $G = (V, E)$ denote a graph, $G_s = (V_s, E_s)$ and $G_t = (V_t, E_t)$ denote two random subgraphs of G for two types of locations Loc_s and Loc_t respectively, where V_s, V_t are the locations and we have $V = V_s \cup V_t$. Let $n_s = |V(G_s)|$ and $n_t = |V(G_t)|$, E_s, E_t are determined by adjacency matrices $A \in \{0, 1\}^{n_s \times n_s}$ and $B \in \{0, 1\}^{n_t \times n_t}$. We further define an induced subgraph with adjacency matrix C where vertices of this subgraph are $V_s \cup V_t$ and the edges are $E \setminus (E_s \cup E_t)$. Now each entry in A, B, C is a Bernoulli variable such that

$$A_{ij} = \begin{cases} 0, & \text{with probability } a_{ij} \text{ if the } i\text{-th and } j\text{-th} \\ & Loc_s \text{ form a malignant pair} \\ 1, & \text{otherwise} \end{cases}$$

similar for B and C with probability matrices b and c except for C we have $C_{ij} = 0$ when the i -th Loc_s and the j -th Loc_t form a malignant pair. Now we wish to obtain the number of K_3 (triangle) in G . This is equivalent to the number we hope to obtain because if G is a complete graph, then the number of K_3 is $\binom{n_s+n_t}{3}$, all possible combinations. Now if $\{v_i, v_j\}$ is a malignant pair and we exclude all triples containing $\{v_i, v_j\}$, this is identical to finding number of triangles in $G' = (V, E \setminus \{v_i, v_j\})$. And from the definition of α we also have $\alpha_{ss} = \sum_{ij \in \binom{V_s}{2}} a_{ij}$, $\alpha_{tt} = \sum_{ij \in \binom{V_t}{2}} b_{ij}$ and $\alpha_{st} = \sum_{i \in V_s, j \in V_t} c_{ij}$. Since G is

a random graph, we compute the expectation to obtain a fair estimate. To this end, we recall Hölder's inequality for expectations, for non-negative random variables X, Y , for p and q satisfying $\frac{1}{p} + \frac{1}{q} = 1$,

$$\mathbb{E}(XY) \leq (\mathbb{E}(X^p))^{\frac{1}{p}} (\mathbb{E}(Y^q))^{\frac{1}{q}} \quad (\text{A.1})$$

Based on this we can immediately generalize to the following proposition,

Proposition A.1.0.1. *Suppose X, Y, Z are three nonnegative random variables, then*

$$\mathbb{E}(XYZ) \leq \sqrt[3]{\mathbb{E}(X^3)\mathbb{E}(Y^3)\mathbb{E}(Z^3)} \quad (\text{A.2})$$

Proof. We first treat $S = XY$ and invoke A.1 for $p = \frac{3}{2}$ and $q = 3$ to obtain $\mathbb{E}(XYZ) \leq (\mathbb{E}(XY)^{3/2})^{2/3} \mathbb{E}(Z^3)^{\frac{1}{3}} = (\mathbb{E}(X^{3/2}Y^{3/2}))^{2/3} \mathbb{E}(Z^3)^{\frac{1}{3}}$. Now we apply A.1 again on the first term on RHS with $p = q = 2$ to show the result. $\square$

Having this we can show the lemma below,

Lemma A.1.1. *Let $G = (V, E)$ be constructed from G_1 and G_2 as described above, let T be the random variable of the number of triangles in G and α be the malignant pair matrix, then*

$$\mathbb{E}(T) \leq f_{ss} + f_{tt} + f_{st} \quad (\text{A.3})$$

where $f_{ss} = \binom{n_s}{3} \sigma_s^3 - \frac{1}{3}(n_s - 2)\sigma_s \alpha_{ss}$, $f_{tt} = \binom{n_t}{3} \sigma_t^3 - \frac{1}{3}(n_t - 2)\sigma_t \alpha_{tt}$ and $f_{st} = \binom{n_s}{2} \cdot n_t \cdot \sigma_s^2 \sigma_t + \binom{n_t}{2} \cdot n_s \cdot \sigma_s \sigma_t^2 - \frac{1}{3}(n_s - 1)\sigma_t \alpha_{st} - \frac{1}{3}(n_t - 1)\sigma_s \alpha_{st}$

Proof. Let's denote the three vertices of the triangle by v_i, v_j and v_k . We consider different cases according to where the vertices lie in:

1. We first consider the case where they are all in G_s , so they form a triangle if and only if $A_{ij}A_{jk}A_{ki} = 1$. We shall also take into account the various error rates ϵ_i .

Therefore to compute the expectation,

$$\begin{aligned}
\mathbb{E}(T_{G_s}) &= \sigma_s^3 \mathbb{E} \left[\sum_{\{i,j,k\} \in \binom{V_s}{3}} A_{ij} A_{jk} A_{ki} \right] \\
&= \sigma_s^3 \sum_{\{i,j,k\} \in \binom{V_s}{3}} \mathbb{E}[A_{ij} A_{jk} A_{ki}] \\
&\leq \sigma_s^3 \sum_{\{i,j,k\} \in \binom{V_s}{3}} \sqrt[3]{\mathbb{E}(A_{ij}^3) \mathbb{E}(A_{jk}^3) \mathbb{E}(A_{ki}^3)}
\end{aligned}$$

where the inequality follows from [A.2](#). Since A_{ij} 's are Bernoulli random variables, $\mathbb{E}(A_{ij}^m) = \mathbb{E}(A_{ij}) \forall m \in \mathbb{N}$, thus

$$\begin{aligned}
\mathbb{E}(T_{G_s}) &\leq \sigma_s^3 \sum_{\{i,j,k\} \in \binom{V_s}{3}} \sqrt[3]{\mathbb{E}(A_{ij}) \mathbb{E}(A_{jk}) \mathbb{E}(A_{ki})} \\
&= \sigma_s^3 \sum_{\{i,j,k\} \in \binom{V_s}{3}} \sqrt[3]{(1 - a_{ij})(1 - a_{jk})(1 - a_{ki})} \\
&\leq \sigma_s^3 \sum_{\{i,j,k\} \in \binom{V_s}{3}} \frac{3 - a_{ij} - a_{jk} - a_{ki}}{3} \\
&= \binom{n_s}{3} \sigma_s^3 - \frac{(n_s - 2) \sigma_s^3}{3} \sum_{ij \in \binom{V_s}{2}} a_{ij} \\
&= \binom{n_s}{3} \sigma_s^3 - \frac{1}{3} (n_s - 2) \sigma_s \alpha_{ss}
\end{aligned}$$

where the second inequality follows by AM-GM inequality on each term and we only have σ_s of order 1 in the final term because in the simulation of α_{ss} we would have already included σ_s^2 .

2. The same procedure holds when $v_i, v_j, v_k \in V_t$ and we have

$$\mathbb{E}(T_{G_t}) = \binom{n_t}{3} \sigma_t^3 - \frac{1}{3} (n_t - 2) \sigma_t \alpha_{tt}$$

3. Now we consider the case where two vertices of the triangle are in one part while the third one is in the other. We discuss two separate cases. The first one is when $v_i, v_k \in V_1$ and $v_j \in V_2$, so following a similar procedure, the expected number of triangles will be upper bounded by

$$\begin{aligned}
& \sigma_s^2 \sigma_t \sum_{\{i,k\} \in V_s, j \in V_t} \sqrt[3]{\mathbb{E}(C_{ij})\mathbb{E}(C_{jk})\mathbb{E}(A_{ik})} \\
&= \sigma_s^2 \sigma_t \sum_{\{i,k\} \in V_s, j \in V_t} \sqrt[3]{(1 - c_{ij})(1 - c_{jk})(1 - a_{ik})} \\
&\leq \binom{n_s}{2} \cdot n_t \cdot \sigma_s^2 \sigma_t - \sigma_s^2 \sigma_t \sum_{\{i,k\} \in V_s, j \in V_t} \frac{c_{ij} + c_{jk} + a_{ik}}{3}
\end{aligned}$$

Similarly if $v_i \in V_s$ and $v_j, v_k \in V_t$ we have an upper bound $\binom{n_t}{2} \cdot n_s \cdot \sigma_s \sigma_t^2 - \sigma_s \sigma_t^2 \sum_{i \in V_s, \{j,k\} \in V_t} \frac{c_{ij} + c_{ik} + b_{jk}}{3}$. Now let T' denote the random variable of the number of triangles across G_s and G_t , we can add up these two upper bounds to obtain

$$\begin{aligned}
\mathbb{E}(T') &\leq \binom{n_s}{2} \cdot n_t \cdot \sigma_s^2 \sigma_t - \sigma_s^2 \sigma_t \sum_{\{i,k\} \in V_s, j \in V_t} \frac{c_{ij} + c_{jk} + a_{ik}}{3} \\
&\quad + \binom{n_t}{2} \cdot n_s \cdot \sigma_s \sigma_t^2 - \sigma_s \sigma_t^2 \sum_{i \in V_s, \{j,k\} \in V_t} \frac{c_{ij} + c_{ik} + b_{jk}}{3} \\
&\leq \binom{n_s}{2} \cdot n_t \cdot \sigma_s^2 \sigma_t + \binom{n_t}{2} \cdot n_s \cdot \sigma_s \sigma_t^2 \\
&\quad - \sigma_s^2 \sigma_t \sum_{\{i,k\} \in V_s, j \in V_t} \frac{c_{ij} + c_{jk}}{3} - \sigma_s \sigma_t^2 \sum_{i \in V_s, \{j,k\} \in V_t} \frac{c_{ij} + c_{ik}}{3} \\
&= \binom{n_s}{2} \cdot n_t \cdot \sigma_s^2 \sigma_t + \binom{n_t}{2} \cdot n_s \cdot \sigma_s \sigma_t^2 \\
&\quad - \frac{1}{3}(n_s - 1)\sigma_s \alpha_{st} - \frac{1}{3}(n_t - 1)\sigma_t \alpha_{st}
\end{aligned}$$

Having analyzed the different cases, the desired bound is obtained by adding them together. $\square$

Applying the lemma inductively would give the desired result.

A.2 Conversion to second-order bounds

The following proposition is a modification of a derivation in [3].

Proposition A.2.0.1. *Let $\epsilon_{(k)} = \mathbb{P}(k\text{-exRec not well-behaved})$, if*

$$A_1(\epsilon^{(k-1)})^2 - B_1(\epsilon^{(k-1)})^3 \leq \epsilon^{(k)} \leq A_2(\epsilon^{(k-1)})^2 + B_2(\epsilon^{(k-1)})^3$$

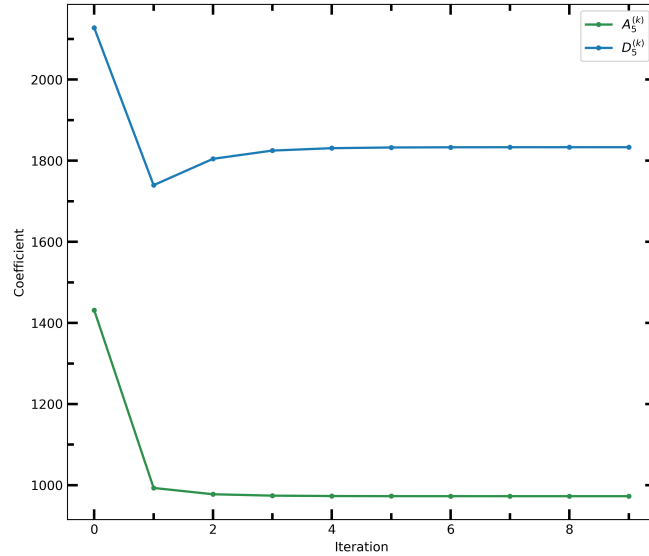
then we can bound $\epsilon^{(k)}$ with terms second order in $\epsilon^{(k-1)}$

$$A'_1(\epsilon^{(k-1)})^2 \leq \epsilon^{(k)} \leq A'_2(\epsilon^{(k-1)})^2$$

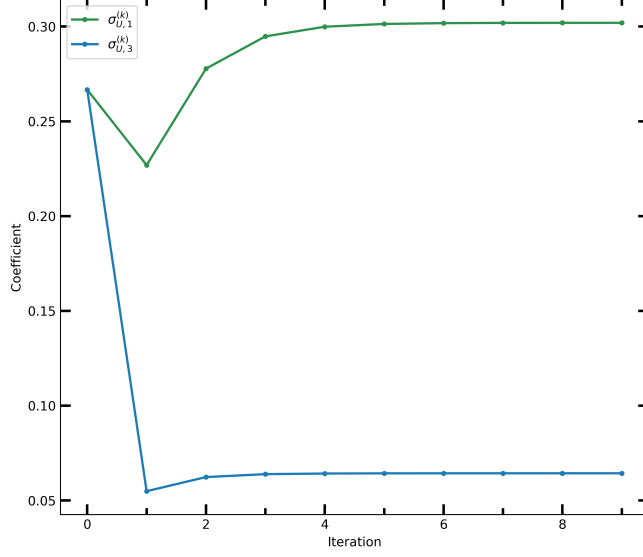
with $A'_2 = \frac{1}{2}A_2(1 + \sqrt{1 + 4B_2/A_2^2})$ and $A'_1 = A_1 - B_1/A'_2$, which holds for $\epsilon^{(k-1)} \leq 1/A'_2$.

A.3 System of equations for the error bounds

We will plot the convergence behaviour of $A_5^{(k)}$ and $D_5^{(k)}$ over 10 iterations



We can see that the sequences converge just after 5 iterations. Also we plot the behaviour of $\sigma_{U,1}^{(k)}$ and $\sigma_{U,3}^{(k)}$



$\forall k \geq 1$,

- When $\epsilon_0^{(k+1)} = \min_{i \leq (k+1)} \{1/D_5^{(i)}\}$, which is used when only local exRecs are involved. $\sigma_{U,1}^{\max} = \sigma_{U,2}^{\max} = 0.302$, $\sigma_{U,3}^{\max} = \sigma_{U,4}^{\max} = 0.0643$, $\sigma_{U,7}^{\max} = 0.540$ and $\sigma_{L,1}^{\min} = \sigma_{L,2}^{\min} = 9.77 \times 10^{-2}$, $\sigma_{U,3}^{\max} = \sigma_{L,4}^{\max} = 0$, $\sigma_{L,7}^{\min} = 0.166$.
- When $\epsilon_0^{(k+1)} = \min_{i \leq (k+1)} \{1/D_5^{(i)}, 1/D_6^{(i)}\}$, that is in *Direct Encoding*, when ϵ_6 is involved. The initial point of $\sigma_6 = 2.09$ and $\epsilon \leq \epsilon'_0 = 2.25 \times 10^{-4}$. Then $\sigma_{U,1}^{\max} = \sigma_{U,2}^{\max} = 0.269$, $\sigma_{U,3}^{\max} = \sigma_{U,4}^{\max} = 0.0587$, $\sigma_{U,6} = 1.92$, $\sigma_{U,7}^{\max} = 0.469$ and $\sigma_{L,1}^{\min} = \sigma_{L,2}^{\min} = 0.105$, $\sigma_{U,3}^{\max} = \sigma_{L,4}^{\max} = 0$, $\sigma_{L,6}^{\min} = 0.538$, $\sigma_{L,7}^{\min} = 0.182$.

A.4 Tightness of bounds

In this section, we compare the theoretical bounds for the CNOT-exRec derived in the main text to the error rates obtained through numerical simulations for the case of $k = 1$. Additionally, we plot the upper bound calculated using the procedure outlined in [3], but applied to our error correction gadget. For the simulation the results are obtained for 10^7 runs, with $\epsilon \leq \epsilon_0$.

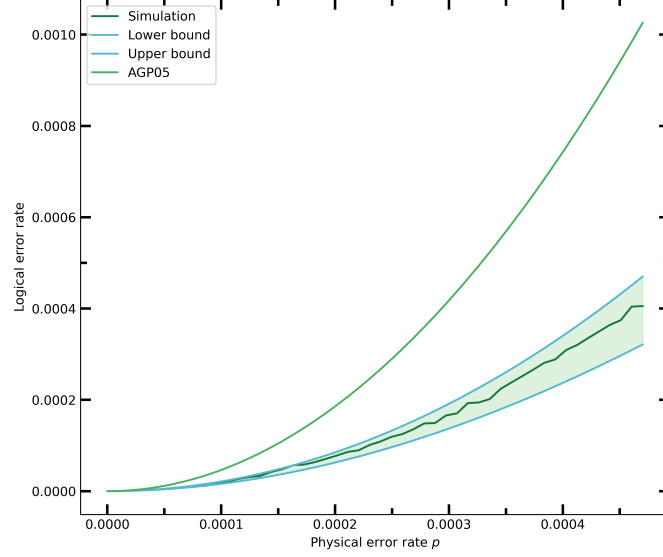


Figure A.1: Comparison of bounds on logical error rate of CNOT-exRec

By plotting the curves we are able to examine the tightness of bounds. We observe that our bounds at $k = 1$ are decent estimates of the true value. Notably, at $\epsilon = \epsilon_0$, the upper bound shows a 54% improvement over the original bound, thereby affirming the credibility of our bounds even as we extend the generalization to higher concatenation levels. This substantiates the significance of our subsequent comparison of the bounds of the two methods.

Appendix B

Direct Encoding Scheme B

B.1 Validity of Scheme B

Here, we justify that the process illustrated in Figure 4.2b indeed prepares an $\overline{\text{ebit}}$ encoded in the Steane code. We will focus on the case of $k = 1$ and use the evolution of stabilizers as discussed in Chapter 2, Section 2.5.

Initially, each ebit has stabilizers $\{XX, ZZ\}$. By labeling the qubits from 1 to 14 with 1-7 being qubits on Alice's side and 8-14 being qubits on Bob's side, e.g. the two qubits of the first ebit are labeled 1 and 8, we establish a starting point for the stabilizers. We first consider measurements performed by Alice, the X -type stabilizers S_1, S_2, S_3 in Table 2.1. Since they always commute with combinations of XX , we consider their effect on ZZ stabilizers. Note we don't have logical operators in this case.

		$Z_1 Z_8$	$Z_1 Z_8$	$b_3 X_1 X_3 X_5 X_7$
		$Z_2 Z_9$	$b_2 X_2 X_3 X_6 X_7$	$b_2 X_2 X_3 X_6 X_7$
$Z_1 Z_8$		$Z_3 Z_{10}$	$Z_2 Z_3 Z_9 Z_{10}$	$Z_1 Z_2 Z_3 Z_8 Z_9 Z_{10}$
$Z_2 Z_9$	$\xrightarrow{\text{measure } X_4 X_5 X_6 X_7}$	$b_1 X_4 X_5 X_6 X_7$	$\xrightarrow{\text{measure } X_2 X_3 X_6 X_7}$	$\xrightarrow{\text{measure } X_1 X_3 X_5 X_7}$
$\dots$			$b_1 X_4 X_5 X_6 X_7$	$b_1 X_4 X_5 X_6 X_7$
$Z_7 Z_{14}$		$Z_4 Z_5 Z_{11} Z_{12}$	$Z_4 Z_5 Z_{11} Z_{12}$	$Z_1 Z_4 Z_5 Z_8 Z_{11} Z_{12}$
		$Z_4 Z_6 Z_{11} Z_{13}$	$Z_2 Z_4 Z_6 Z_9 Z_{11} Z_{13}$	$Z_2 Z_4 Z_6 Z_9 Z_{11} Z_{13}$
		$Z_4 Z_7 Z_{11} Z_{14}$	$Z_2 Z_4 Z_7 Z_9 Z_{11} Z_{14}$	$Z_1 Z_2 Z_4 Z_7 Z_8 Z_9 Z_{11} Z_{14}$

Based on the measurement outcomes b_1, b_2, b_3 , Alice can make corresponding corrections to ensure the current state is in $+1$ eigenspace of the stabilizers. Next, Bob measures

the Z -type stabilizers and the XX stabilizers will have a similar transformation as above, resulting in 7 stabilizers which are obtained by exchanging X and Z in the final stabilizers above. Again Bob will make his corrections based on measurement outcomes. Now the 14 stabilizers are again split into 7 X -type and 7 Z -type. We list the X -type stabilizers below

$$\begin{aligned}
&X_1X_3X_5X_7 \\
&X_2X_3X_6X_7 \\
&X_1X_2X_3X_8X_9X_{10} \\
&X_4X_5X_6X_7 \\
&X_1X_4X_5X_8X_{11}X_{12} \\
&X_2X_4X_6X_9X_{11}X_{13} \\
&X_1X_2X_4X_7X_8X_9X_{11}X_{14}
\end{aligned}$$

The group generated by these stabilizers is actually equivalent to the one generated by the following set

$$\begin{aligned}
&X_1X_3X_5X_7 \\
&X_8X_{10}X_{12}X_{14} \\
&X_2X_3X_6X_7 \\
&X_9X_{10}X_{13}X_{14} \\
&X_4X_5X_6X_7 \\
&X_{11}X_{12}X_{13}X_{14} \\
&X_1X_2 \dots X_{14}
\end{aligned}$$

To see the equivalence, we note a fact: If we have two generator sets $\{P_1, \dots, P_n\}$ and $\{Q_1, \dots, Q_n\}$ and a well-defined binary operation. If some combinations of $P_1, \dots, P_n$ under the binary operation result in $Q_1, \dots, Q_n$ and vice versa, the two sets generate the same group. Based on this, we can verify that the given stabilizer sets indeed generate the same group. The same applies to Z -type operators. We further observe that these stabilizers are exactly the stabilizers for $\overline{\text{ebit}}$, thus justifying our claim. In fact, for any CSS codes, measuring the stabilizers will give an $\overline{\text{ebit}}$ ¹.

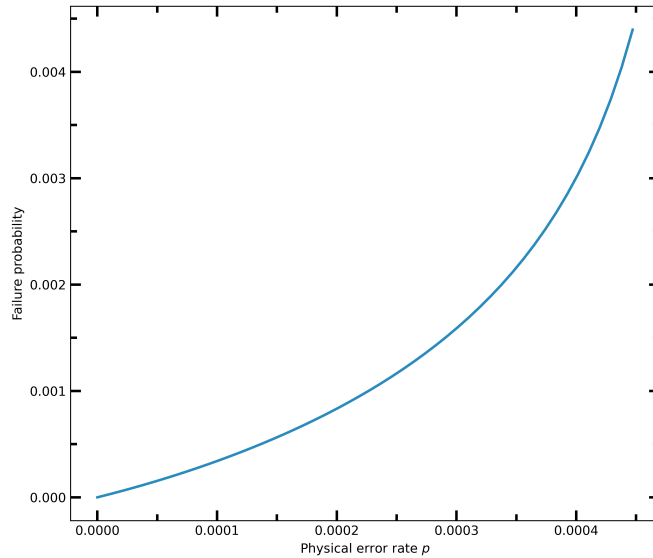
¹Upon completing this thesis, I found proof of the analogous result for all CSS codes in Appendix B in the work by Rengaswamy [75]. Interested readers can refer to this more general proof.

Appendix C

Interface+EPP error analysis

C.1 Interface error analysis

We will first plot the upper bound of $\mathbb{P}(\text{Enc}_l \text{ bad})$ with respect to ϵ below



Next, we will analyze the faults and resultant logical errors of the interface in more detail. In particular, we will include errors in the incoming state $|\phi\rangle$. We will need to take

into account the distinct ϵ_6 error and also the different logical error types. By observing the circuit of the interface, we infer that if there is an incoming X error, this results in a $\bar{Z}$ error; $Y \longrightarrow \bar{Y}$ and $Z \longrightarrow \bar{X}$. Moreover, we take into account the fact that an encoded EPR pair is stabilized by $\{\bar{X}\bar{X}, \bar{Z}\bar{Z}\}$. We arrive at the following enumeration:

- **Single fault location**

Excluding Loc_6 :

$\bar{X}$	[0,1,1,0,0.684,0,0]
$\bar{Y}$	[0,0,0,0,0.674,0,0]
$\bar{Z}$	[0,0,0,1,0.656,0,0]

To include the initial EPR pair, for the upper interface we have an extra $\bar{X}$ -error from Loc_2 and for the lower interface we have an extra $\bar{Z}$ -error from the Loc_1 . For Loc_6 , as explained before, for an EPR pair the possible errors are $XI/YI/ZI$, with XI occurring with probability $I_X(q, \epsilon_0)$ and similarly for YI/ZI .

- **Malignant pairs**

For the malignant pairs, we will only enumerate for $k = 1$ and thus we include the respective σ_i . Excluding Loc_6 :

$\bar{X}$	$\bar{Y}$	$\bar{Z}$
262.8	67.7	227.3

Malignant pairs involving Loc_6 that induce logical error on $\overline{\text{EPR}}_1$

$\bar{Y}\bar{Y}$	$\bar{X}\bar{X}$	$\bar{Z}\bar{Z}$	$\bar{X}\bar{I}$	$\bar{I}\bar{X}$
0	0	0	31.0	0.21
$\bar{Z}\bar{I}$	$\bar{I}\bar{Z}$	$\bar{Y}\bar{I}$	$\bar{X}\bar{Z}$	$\bar{Z}\bar{X}$
31.4	0.242	31.2	0.321	0.394
$\bar{I}\bar{Y}$	$\bar{Y}\bar{X}$	$\bar{X}\bar{Y}$	$\bar{Z}\bar{Y}$	$\bar{Y}\bar{Z}$
0	0	0	0	0

C.2 Proofs for bounds on EPP failure probabilities

Proof of 4.5.2. We can view the *Interface+EPP-A* circuit as consisting of two major components, one is the interface preparation of logical EPR pairs and the other is the EPP procedure. Below we use $\mathcal{G}_1$ to denote the instance that the preparation of 4 logical EPR pairs is bad and $\mathcal{G}_2$ to denote the cases where the EPP is bad. Thus we have

$$\mathbb{P}(\text{EPR rejected}) \leq \mathbb{P}(\mathcal{G}_2) + \mathbb{P}(\mathcal{G}_1)$$

We would first compute the upper bound on $\mathbb{P}(\mathcal{G}_2)$. For the upper bound, since we have 6 CNOT-exRecs and 6 mmt-exRecs,

$$\mathbb{P}(\mathcal{G}_2) \leq \sum_i |\text{Loc}_i| \varepsilon_i^{(k)} \leq 2 \cdot 6 \cdot \left(\nu_5^{(k)} + \nu_4^{(k)} \right)$$

which follows from the union bound and the factor 2 follows from P.13 of [3], resulting in a noisy simulation of the whole quantum circuit.

Now we will compute an upper bound on $\mathbb{P}(\mathcal{G}_1)$, to do so, observe that the preparation part consists of 4 EPR pairs, and on each side the qubits are expanded via the interface, so 8 interfaces in total. We may now invoke Lemma 4.5.1 and obtain the following

$$\mathbb{P}(\mathcal{G}_1) \leq 4\epsilon_6 + 8f_{in}(\epsilon)$$

we observe that $\mathbb{P}(\mathcal{G}_2) \leq 12\epsilon \forall k$, adding up two terms gives the desired upper bound. $\square$

Observe that this analysis is applicable not only to concatenated Steane code, any proposal of code concatenation that has the corresponding interface with a logical error rate upper bound independent of k will have EPP failure probability that is also independent of k (given the EPP part can be performed fault-tolerantly in the logical space).

Proof of Corollary. 4.5.2.1 We would now derive a crude lower bound on $\mathbb{P}(\text{EPR rejected})$ and we would consider terms up to $\mathcal{O}(\epsilon^3)$, so this bound will hold well for $\epsilon < \epsilon_0$ considered in this work. We may again use the Bonferroni inequality (or inclusion-exclusion principle) and thus,

$$\mathbb{P}(\text{EPR rejected}) \geq \mathbb{P}(\mathcal{G}_1) + \mathbb{P}(\mathcal{G}_2) - \mathbb{P}(\mathcal{G}_1 \wedge \mathcal{G}_2)$$

For the first term above, based on the observation that if certain locations are faulty then they will surely cause a logical error, for $\mathcal{O}(\epsilon)$ terms, this will cause a logical error if it

is in the initial $|\Phi\rangle$ or $\text{Enc}_{0 \rightarrow 1}$ while other locations are error-free. Note that the error probability for exRecs in $\text{Enc}_{l \rightarrow l+1}$ is $\varepsilon_i^{(l)}$, so for $\mathcal{O}(\epsilon)$ terms we have the lower bound (in the following $\gamma_7 = 111$ being the number of locations in $\text{Enc}_{0 \rightarrow 1}$),

$$\begin{aligned} & 8 \cdot 2.8 \cdot \epsilon(1 - \epsilon)^{-1}(1 - \epsilon_6)^4 \prod_{i=0}^{k-1} \prod_{t=1}^7 \left(1 - \nu_t^{(i)}\right)^{8n_{in,t}} \\ & + 4 \cdot \epsilon_6(1 - \epsilon_6)^3 \prod_{i=0}^{k-1} \prod_{t=1}^7 \left(1 - \nu_t^{(i)}\right)^{8n_{in,t}} \end{aligned} \quad (\dagger)$$

For the product term, we use the Weierstrass product inequality and obtain

$$\prod_{i=0}^{k-1} \prod_{t=1}^7 (1 - \nu_t^{(i)})^{8n_{in,t}} \geq \left(1 - 8 \sum_{i=0}^1 \sum_{t=1}^7 n_{in,t} \nu_t^{(i)}\right)$$

for the early terms, we apply $(1 + x)^r \geq 1 + rx$. Combining these with the specific values of $\nu_t^{(i)}$, we can simplify Eqn($\dagger$) as, to $\mathcal{O}(\epsilon^3)$,

$$\begin{aligned} & 22.4\epsilon(1 - 605.9\epsilon - 4\epsilon_6 + 2423.6\epsilon\epsilon_6 - 1.10 \times 10^6 \epsilon^2) \\ & + 4\epsilon_6(1 - 3\epsilon_6 - 606.9\epsilon - 1.10 \times 10^6 \epsilon^2 + 1820.7\epsilon\epsilon_6) \end{aligned}$$

Similarly, for two-fault terms, there are a number of cases to be considered,

1. Logical errors resulting from malignant pairs in $\text{Enc}_{0 \rightarrow 1}$, we have the lower bound,

$$\begin{aligned} & 8 \sum_{ij} \alpha_{in}(i, j) \sum_{s=0}^{k-1} \mu_i^{(s)} \mu_j^{(s)} (1 - \nu_5^{(s)})^{-2} \dots \\ & \dots \prod_{i=0}^{k-1} \prod_{t=1}^7 (1 - \nu_t^{(i)})^{8n_{in,t}} (1 - \epsilon_6)^4 \end{aligned}$$

2. For $k \geq 2$, logical errors resulting from one faulty-exRec in $\text{Enc}_{1 \rightarrow 2}$. Note that for exRecs in $\text{Enc}_{l \rightarrow (l+1)}$ $l \geq 3$ to be faulty, $\mathcal{O}(\epsilon^4)$ would be needed, so are not considered

here. Hence for this case, we have the lower bound

$$8 \cdot 2.8 \cdot \mu_5^{(1)} (1 - \nu_5^{(1)})^{-1} \dots \\ \dots \prod_{i=0}^{k-1} \prod_{t=1}^7 (1 - \nu_t^{(i)})^{8n_{in,t}} (1 - \epsilon_6)^4$$

3. If we have one fault in the initial EPR and one in one of the $\text{Enc}_{0 \rightarrow 1}$. From Table ?? we have

$$4 \cdot 94.7 \cdot \epsilon_6 \epsilon (1 - \epsilon)^{-1} (1 - \epsilon_6)^3 \prod_{i=0}^{k-1} \prod_{t=1}^7 (1 - \nu_t^{(i)})^{8n_{in,t}}$$

4. We further have cases where there are two faults in two interfaces respectively. Apart from naively computing all combinations, there is a subtlety here, being that if two logical EPR pairs induce the same type of logical error, after EPP they end up canceling each other, for example, if on Alice's side, the first and second logical EPRs have X -error, the first logical EPR will still be accepted, so we should not include such cases. To this end, we will refer to Table ?? and enumerate a lower bound of 192.8 such cases. So such cases occur with a probability

$$192.8 \cdot \epsilon^2 (1 - \epsilon)^{-2} (1 - \epsilon_6)^4 \prod_{i=0}^{k-1} \prod_{t=1}^7 (1 - \nu_t^{(i)})^{8n_{in,t}}$$

Summing up the above second-order cases and lower bound all the $(1 - x)^r$'s, we have, to $\mathcal{O}(\epsilon^3)$

$$26768.7\epsilon^2 + 378.8\epsilon\epsilon_6 - 1.62 \times 10^7 \epsilon^3 - 3.37 \times 10^5 \epsilon^2 \epsilon_6 - 1136.4\epsilon\epsilon_6^2$$

for $k \geq 2$. For the next term, we note that when $k \geq 2$, for $\mathbb{P}(\mathcal{G}_2)$ to cause a logical error we need at least $\mathcal{O}(\epsilon^4)$. Now for the last term, we note that for both components to have logical errors, we need at least $\mathcal{O}(\epsilon^5)$ in the circuit for $k \geq 2$ because for $\mathcal{G}_2$ to occur we need $\mathcal{O}(\epsilon^4)$ and one fault in the EPR preparation. Combining all the terms above and the fact that $\epsilon \leq \epsilon_0$ we obtain the desired result. $\square$

Appendix D

Miscellaneous

D.1 Distribution of logical error after CNOT-exRec

The probability distribution of malignant pairs that cause different types of logical errors for a CNOT-1 exRec (when $\epsilon_5 = \epsilon_6$).

$\overline{YY}$	$\overline{XX}$	$\overline{ZZ}$	$\overline{XI}$	$\overline{IX}$
7.21×10^{-5}	9.01×10^{-3}	1.53×10^{-2}	0.161	0.298
$\overline{ZI}$	$\overline{IZ}$	$\overline{YI}$	$\overline{XZ}$	$\overline{ZX}$
0.302	0.160	2.03×10^{-2}	5.65×10^{-4}	8.52×10^{-3}
$\overline{IY}$	$\overline{YX}$	$\overline{XY}$	$\overline{ZY}$	$\overline{YZ}$
1.97×10^{-2}	1.83×10^{-3}	1.20×10^{-4}	1.65×10^{-3}	1.44×10^{-4}

D.2 Bounds for Shor's parity measurement

Here we investigate the Shor parity measurement. We will exemplify this by examining the case where Alice performs a $\overline{XX}$ measurement on her side. Firstly we will obtain an upper bound on the rejection probability of the 14-qubit ancilla cat state. The number of single fault locations that will lead to rejection is [14, 0,0,1,8,0,0]. There are 38 locations in total with a breakdown of $\mathbf{n}_{\text{Shor}} = [14, 1, 7, 1, 15, 0, 0]$. Thus when we encode to level- k ,

we simply arrive at the upper bound

$$\begin{aligned}
\mathbb{P}(\text{Cat state rejected}) &\leq (14\sigma_{U,1}^{\max} + \sigma_{U,3}^{\max} + 8) \nu_5^{(k-1)} \\
&\quad + \sum_{\{i,j\} \in (\mathbf{n}_{\text{Shor}})} \sigma_{U,i}^{\max} \sigma_{U,j}^{\max} \left(\nu_5^{(k-1)} \right)^2 \\
&= 11.8\nu_5^{(k-1)} + 196.3 \left(\nu_5^{(k-1)} \right)^2 \\
&\leq 11.9\nu_5^{(k-1)}
\end{aligned}$$

In general, two scenarios of the parity measurement would lead to the failure of the magic square game. First, if output logical errors occur during the initial two parity measurements on the data block, subsequent measurements are adversely affected. Second, faults within the measurement process itself can induce a 'parity-change', contributing to the failure. We say the Shor's measurement is bad if either of these happens. Having this observation, we compute the bounds. As this procedure was justified to be fault-tolerant, we again start by enumerating the malignant pairs. The MPM is listed in Appendix D.3. So for level- k concatenation, if we denote $\varepsilon_{\text{Shor,joint}}^{(k)}$ to be the case when all cat state ancillas are accepted but Shor's measurement is bad, following the same procedure as in 4.4, we arrive at the following bounds

$$3.27\mu_0 \left(\frac{\epsilon}{\mu_0} \right)^{2^k} \leq \varepsilon_{\text{Shor,joint}}^{(k)} \leq 1.27\epsilon_0 \left(\frac{\epsilon}{\epsilon_0} \right)^{2^k}$$

Hence for the conditional probability $\varepsilon_{\text{Shor}}^{(k)}$

$$3.27\mu_0 \left(\frac{\epsilon}{\mu_0} \right)^{2^k} \leq \varepsilon_{\text{Shor}}^{(k)} \leq 1.27\epsilon_0 \left(\frac{\epsilon}{\epsilon_0} \right)^{2^k} \left(1 - 11.9\nu_5^{(k-1)} \right)^{-3}$$

We shall denote the lower and upper bounds by $\mu_{\text{Shor}}^{(k)}$ and $\nu_{\text{Shor}}^{(k)}$ respectively.

D.3 Malignant pair matrix(MPM)

1. $|\bar{0}\rangle / |\bar{+}\rangle$ -exRec

$$\mathbf{n}_1 = [11, 13, 9, 8, 47, 0, 0]$$

$$\alpha_1 = \begin{pmatrix} 10.0 & & & & & & \\ 0 & 11.0 & & & & & \\ 0 & 25.0 & 0 & & & & \\ 14.0 & 0 & 0 & 0 & & & \\ 70.0 & 69.9 & 81.2 & 74.4 & 172.2 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2. Z -mmt-exRec

$$\mathbf{n}_3 = [8, 8, 8, 15, 36, 0, 0]$$

$$\begin{pmatrix} 0 & & & & & & \\ 0 & 0 & & & & & \\ 0 & 0 & 0 & & & & \\ 18.0 & 0 & 0 & 63.0 & & & \\ 0 & 0 & 0 & 42.8 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

3. EC -gadget

$$\mathbf{n}_3 = [8, 8, 8, 8, 36, 0, 0]$$

$$\begin{pmatrix} 5.0 & & & & & & \\ 0 & 5.0 & & & & & \\ 0 & 7.0 & 0 & & & & \\ 7 & 0 & 0 & 0 & & & \\ 44.0 & 42.0 & 59.5 & 59.0 & 130.2 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

4. *Identity gate*

$$\mathbf{n}_{in} = [16, 16, 16, 16, 72, 0, 7]$$

$$\begin{pmatrix} 14.0 & & & & & & \\ 0 & 14.0 & & & & & \\ 0 & 49.0 & 42.0 & & & & \\ 49.0 & 0 & 0 & 42.0 & & & \\ 100.7 & 100.7 & 181.4 & 181.4 & 278.6 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \\ 28.0 & 28.0 & 56.0 & 56.0 & 148.8 & 0 & 16.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

5. $\overline{ebit}^{(k)}$ from Direct Encoding

$$\alpha_{\text{EPR}} = \begin{pmatrix} 53 & & & & & & \\ 0 & 53 & & & & & \\ 0 & 189.96 & 167.95 & & & & \\ 189.95 & 0 & 0 & 167.94 & & & \\ 360.36 & 359.9 & 640.91 & 646.04 & 916.07 & & \\ 43.73 & 44.93 & 89.56 & 90.08 & 231.49 & 12.7 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

6. Interface, $\text{Enc}_{0 \rightarrow 1} \circ \text{EC}$

$$\mathbf{n}_{in} = [14, 12, 9, 11, 58, 0, 7]$$

$$\alpha_{in} = \begin{pmatrix} 10.0 & & & & & & \\ 5.0 & 18.0 & & & & & \\ 5.0 & 39.0 & 7.0 & & & & \\ 20.0 & 15.0 & 15.0 & 7.0 & & & \\ 92.6 & 127.0 & 140.3 & 137.9 & 281.5 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \\ 19.8 & 30.9 & 35.0 & 35.1 & 121.6 & 0 & 16.4 \end{pmatrix}$$

7. *Logical EPP*

$$\alpha_{EPP} = \begin{pmatrix} 102 & & & & & & \\ 0 & 34.0 & & & & & \\ 0 & 110 & 84.0 & & & & \\ 426 & 0 & 0 & 420 & & & \\ 832 & 303 & 500 & 1626 & 1634 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

8. *Shor- $\overline{X}$ -mmt (full)*

$$\mathbf{n}_{S_3} = [106, 67, 85, 67, 375, 0, 0]$$

$$\alpha_{\text{Shor}} = \begin{pmatrix} 249 & & & & & & \\ 0 & 114 & & & & & \\ 0 & 546 & 756 & & & & \\ 315 & 0 & 0 & 252 & & & \\ 992 & 1094 & 2720 & 1492 & 3532 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$