Energy Barrier in Quantum Error Correcting Codes

GUANGQI ZHAO

Supervisor: Andrew C. Doherty
Associate Supervisor: Isaac H. Kim

A thesis submitted in fulfillment of the requirements
for the degree of
Doctor of Philosophy (Science)

This research reported in this thesis was supported by the award of
a Sydney Quantum Academy scholarship to the PhD Candidate.

School of Physics
Faculty of Science
The University of Sydney
Australia

March 2026

Abstract

Quantum error correction is necessary for the building of a useful fault-tolerant quantum computer. However, in practice, error correction requires a large amount of fault-tolerance overheads and fast decoding. One promising approach to overcome this challenge is to use codes with a macroscopic energy barrier, which can be used to build self-correcting quantum memory. So far, rigorous bounds on the energy barrier have typically been derived only for specific codes. Determining a lower bound for the energy barrier of codes is generally challenging due to two reasons: there are too many possible error paths (sequences of local errors) that can implement a fixed logical operator, and the logical operator itself is defined only modulo stabilizers. In this thesis, we prove tight bounds on the energy barrier that are applicable to any quantum code obtained from the hypergraph product of two classical codes. If the underlying classical codes are low-density parity-check codes (LDPC), the energy barrier of the quantum code is shown to be the minimum energy barrier of the underlying classical codes (and their transposes) up to an additive $O(1)$ constant.

Hypergraph product codes, at best, preserve the energy barrier of the underlying classical codes. Constructing a quantum code with a macroscopic energy barrier requires the underlying classical codes to feature an extensive energy barrier. However, determining the energy barrier for classical codes is challenging, although it is comparatively simpler than for quantum codes because of the absence of stabilizer effects. Utilizing a property of tensor product codes, we demonstrate that higher-dimensional hypergraph products may construct quantum codes with a macroscopic energy barrier, even when the underlying classical codes lack this property. Specifically, the 3D hypergraph product enhances the energy barrier of one type of logical operator. In contrast, in the 4D case, both the X and Z logical operators may be designed to exhibit an extensive energy barrier.

In addition, we demonstrate the use of periodic driving to identify the toric code phase, characterized by a 2T oscillation in the logical Z operator's measurement outcomes. Simulations were conducted for various parameters, including error rates and system sizes. To enhance the 2T oscillation, we incorporated a decoder read-out process. The results show that lower error rates lead to longer-lasting oscillations, and the decoder can help to improve oscillation persistence. Notably, the impact of the decoder becomes more pronounced as the size of the system increases. These findings confirm that the periodic driving term, combined with the decoder readout process, can be used to identify toric code states in quantum memory systems effectively.

Statement of originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes. I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Guangqi Zhao

December 2025

Statement of Student Contribution

This thesis consists of an introduction, three main body chapters, and a conclusion. I wrote both the introduction and the conclusion. The three body chapters are adapted from three distinct research projects. My contributions to each project are detailed below.

Chapter 2: Topologically Ordered Time Crystal in the Toric Code

Contributions: The core research ideas were conceived by Isaac Kim. Isaac and I developed the software package associated with the project. I conducted the data analysis and composed the draft. Isaac Kim and Andrew Doherty provided supervision and mentorship throughout. The rationale for arguing that the simulation parameter $b_j = 0.25\pi$ is optimal is credited to Raditya W. Bomantara.

Chapter 3: Energy Barrier of Hypergraph Product Codes

Author List: Guangqi Zhao, Andrew C. Doherty, Isaac H. Kim

Contributions: I am the first author of this article and the originator of the underlying research ideas. I was responsible for all analytical results, including the development of lemmas and proofs, and I drafted the initial version of the manuscript. Isaac Kim restructured the manuscript and refined the proofs. Andrew Doherty offered supervision, mentorship, and crucial insights that guided the proofs of Lemma 3.3.1 and Lemma 3.5.1. This chapter has been published in Physical Review Letters as Phys. Rev. Lett. 134, 180601.

Chapter 4: Improved energy Barrier of Higher-Dimensional Hypergraph Product Codes

Author List: Guangqi Zhao

Contributions: I proposed the research ideas and derived the results presented in this chapter. I wrote all the lemmas and proofs. Isaac Kim and Andrew Doherty

provided supervision and mentoring. This chapter has been published in Physical Review A as Phys. Rev. A 113, 012440.

Guangqi Zhao

December 2025

As a supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Andrew Doherty

December 2025

Artificial Intelligence Attribution Statement

During the conduct of the projects in this thesis, generative AI tools were used to a very limited extent. Specifically, Cursor (with Claude Code Models) were used to assist with coding in Section 4, in order to verify Conjecture 4.3.1. The corresponding code can be found as referenced in Appendix A 4.3. All code and numerical results were reviewed, tested, and validated by the author, who takes full responsibility for their correctness and interpretation.

During the preparation of the thesis manuscript, a generative AI tool within Overleaf was used for language editing, including correction of spelling, grammar, and sentence clarity. No generative AI tools were used to generate original research content, results, or substantive arguments.

The author confirms that the work is their own, and has used generative AI in accordance with University of Sydney guidelines and policies on the responsible use of generative AI.

Guangqi Zhao

March 2026

Acknowledgements

First, I would like to express my deepest gratitude to my supervisors, Andrew Doherty and Isaac Kim, for their unwavering guidance, invaluable insights, and continuous support throughout my PhD journey. Your expertise and mentorship have been pivotal in shaping my research and my growth as a scholar. I am especially grateful to Andrew for patiently answering my dumb questions every week. Your willingness to answer every question, complex or simple, has been invaluable. To Isaac, I sincerely appreciate your great insight into my research problems. Your way of thinking has broadened my perspective and taught me so much. It is a great regret that I have never had the opportunity to meet Isaac in person, but I hope to remedy this one day. Most importantly, I am grateful to you both for allowing me the freedom to pursue research topics that genuinely interest me. Although my PhD journey has not been as fruitful in terms of outcomes, I have greatly valued the opportunity to explore and deepen my understanding of the subjects I am passionate about.

I am sincerely grateful to the Sydney Quantum Academy for providing the Ph.D. scholarship that made this program possible. I also extend my heartfelt thanks to Arne Grimsmo for his recommendation and steadfast support, which were instrumental in securing this opportunity. Your belief in my potential opened doors that I could only have dreamed of, and I will always be deeply indebted to your kindness and advocacy.

I would like to express my sincere gratitude to Professor Tom Stace and Professor Dan Browne for serving as examiners of my thesis. I deeply appreciate the considerable time and effort they dedicated to thoroughly reviewing my work. Their insightful feedback, thoughtful comments, and constructive suggestions have been invaluable in strengthening the quality and clarity of this thesis.

To my family - my mother and father - your unconditional love, unwavering encouragement, and belief in my potential have been the foundation of my strength and perseverance. Despite being farmers in remote mountain regions, your tireless efforts and sacrifices have

allowed me to venture beyond our world and pursue my dreams. I am forever grateful for everything you have done and have always supported me regardless of distance or challenges. I hope you take more time to care for yourself and stay healthy.

Finally, to my girlfriend, Wenxuan, thank you for your patience, understanding, and unwavering support throughout this journey. After three years apart and countless FaceTime calls, we have finally made it, and it is time to reunite. Your love and belief in me have kept me grounded and motivated, and I am truly lucky to have you by my side.

Contents

Abstract ii

Statement of originality iv

Statement of Student Contribution v

Artificial Intelligence Attribution Statement vii

Acknowledgements viii

Contents x

List of Figures xiv

Chapter 1 Introduction to quantum error correction 1

1.1 Classical error correction . . . 3

1.1.1 Classical noisy channel . . . 3

1.1.2 Classical error correction properties . . . 5

1.1.3 The repetition code . . . 6

1.1.4 Code parameters: rate and distance . . . 7

1.1.5 Linear codes and Tanner graph . . . 8

1.1.6 LDPC code . . . 9

1.2 Quantum error correction (QEC) . . . 9

1.2.1 Quantum state and qubits . . . 9

1.2.2 Qubit error . . . 11

1.2.3 Quantum error channels and superoperators . . . 12

1.2.3.1 Superoperators . . . 12

1.2.3.2 Depolarizing channel . . . 13

1.2.3.3 Phase damping . . . 14

1.2.3.4 Amplitude damping 15
1.2.4 Correting quantum errors 16
1.2.5 Necessary and sufficient conditions for QEC 17
1.2.6 Stabilizer codes 18
1.2.7 Steane code 20
1.2.8 Quantum LDPC code 21
1.2.9 Decoder 21
1.3 Universal quantum computation 26
1.3.1 Single qubit gate 27
1.3.2 Clifford hierarchy 27
1.3.3 Solovay-Kitaev theorem 29
1.4 Fault tolerant and threshold theory 29
1.4.1 Noise and control model 30
1.4.2 Transversal gates and Eastin-Knill theorem 31
1.4.3 Concatenated codes and threshold 32
1.5 Difficulties in QEC and the structure of this thesis 34

Chapter 2 Topological ordered time crystal in Toric code 36
2.1 Introduction 36
2.1.1 Toric code 37
2.1.1.1 Code construction 37
2.1.1.2 Errors on the toric code 39
2.1.1.3 Error model and decoder 40
2.1.2 Discrete time crystal 41
2.2 Periodical driving 43
2.3 Simulation results 45
2.3.1 Different stabilizer prefactor 45
2.3.2 Different error rate 49
2.3.3 Different system size 51
2.4 Outlook 52

Chapter 3 Energy barrier of hypergraph product codes 54
3.1 Introduction 54
3.2 Energy barrier of codes 56
3.3 Quantum LDPC codes and their energy barriers 59
3.4 Hypergraph product code and its logical operators 61
3.4.1 The graph structure of hypergraph product codes 62
3.4.2 Canonical logical operators 63
3.5 Energy barrier of hypergraph product code. 65
3.5.1 Upper bound 66
3.5.2 Lower bound 67
3.5.3 Energy barrier 73
3.6 Expander codes 74
3.7 Outlook 75

Chapter 4 Improved energy barrier of higher-dimensional hypergraph product codes 76
4.1 Preliminaries 78
4.1.1 Quantum LDPC codes 78
4.1.2 Energy barrier of codes 79
4.1.3 Soundness 80
4.1.4 Confinement 83
4.1.5 Higher-dimensional hypergraph product 85
4.2 Confinement and energy barrier 89
4.3 Tensor product code and its energy barrier 91
4.3.1 Code construction 92
4.3.2 Energy barrier of tensor product code 93
4.3.2.1 Lower bound 94
4.3.2.2 Upper bound - “strip” argument 97
4.4 Energy barrier of 3D hypergraph product codes 99
4.4.1 Code construction of 3D hypergraph product codes 99
4.4.2 Logical operators of 3D hypergraph product codes 100

4.4.3 Energy barrier of Z logical operators . . . 104
4.4.3.1 Upper bound: “strip” argument . . . 104
4.4.3.2 Lower bound . . . 105
4.4.4 Energy barrier of X logical operators . . . 112
4.5 Energy barrier of 4D hypergraph product codes . . . 112
4.6 Conclusion and outlook . . . 117

Chapter 5 Conclusion and outlook **119**

Bibliography **122**

Appendix A Appendix **144**
A1 Homology . . . 144
A2 CSS code and its homological description . . . 147
A3 Hypergraph product code from the homological construction . . . 149
A4 Vector reshaping . . . 150
A4.1 Proof of Lemma 4.4.1 . . . 151
A4.2 Proof of Lemma 4.4.2 . . . 154
A4.3 Discussion on Conjecture 4.3.1 . . . 157

List of Figures

2.1 Stabilizers in the Toric code. Here, the lattice has a periodic boundary condition, and the stabilizer S_X (S_Z) is denoted in red (blue). 38

2.2 Logical operators in the toric code are represented by nontrivial circles on the torus. 39

2.3 The performance of the MWPM decoder is depicted in two sections: the upper section illustrates its performance for even lattice sizes, while the lower section shows its performance for odd lattice sizes. The transverse axis is the physical error rate, while the vertical axis is the logical error rate after decoding. Different colors label different system sizes. The threshold is approximately 10%. When the physical error rate is less than the threshold, an increase in the system size leads to better performance (lower logical error rate), while this becomes not true when the error rate is larger than the threshold. 42

2.4 Simulation results with different b_j. System size is 3×3 with 18 qubits. $a_l = 0.4\pi$ with a perturbation uniformly selected from $(-0.1, 0.1)$ for each l. $c_k = 0.02\pi$, with a perturbation uniformly selected from $(-0.1, 0.1)$ for each k. Different color corresponding to different value of b_j, which is chosen from the set $[0.00\pi, 0.05\pi, 0.15\pi, 0.25\pi, 0.35\pi, 0.45\pi, 0.50\pi]$, with perturbations uniformly sampled from $(-0.5, 0.5)$ for each j. The transverse axis is the time step, while the vertical axis is the average measurement outcome of the logical Z operator. The plot on the left is the result without the decoder, while the plot on the right is the case with the decoder. 46

2.5 Simulation results across varying error rates. The logical operator coefficient $a_l = 0.4\pi$ with a perturbation uniformly selected from $(-0.1, 0.1)$. The system size is 3×3 with 18 qubits. $b_j = 0.5\pi$ with a perturbation uniformly sampled from $(-0.5, 0.5)$ for each j. Error rates c_k are selected from $[0.01\pi, 0.02\pi, 0.03\pi, 0.06\pi, 0.10\pi]$ with uniform perturbations in $(-0.1, 0.1)$. For

different error rates, they compare decoder (green) and without-decoder (purple) results, with time steps on the x-axis and logical Z operator measurements on the y-axis 50

2.6 Simulation results with different system sizes. In U_T, $b_j = 0.5\pi$ with a perturbation uniformly sampled from $(-0.5, 0.5)$ for each j. $a_l = 0.4\pi$ with a perturbation uniformly selected from $(-0.1, 0.1)$ for each l. $c_k = 0.02\pi$, with a perturbation uniformly selected from $(-0.02, 0.02)$ for each k. We consider three system sizes 3×2, 5×2, and 7×2, with 12, 20, and 28 qubits, respectively. 51

2.7 Simulation results for different system sizes are shown. The parameters are the same as before, but we separately plot the cases with and without the decoder to directly analyze the effect of system size. 52

3.1 Graphical illustration of the energy barrier for any stabilizer S of quantum LDPC codes. Any stabilizer can be decomposed as $S = s_1 s_2 \cdots s_m$, where $s_1, s_2, \ldots, s_m$ are constant-weight stabilizers. This naturally defines a construction path for S by sequentially constructing $s_1, s_2, \ldots, s_m$. Consequently, the energy barrier of S is bounded by the energy barriers of these constant-weight stabilizers, which are constant. 60

3.2 Graphical illustration of the Tanner graph of the hypergraph product codes. The Tanner graphs of $\mathcal{G}_X$ $(\mathcal{G}_1 \times \mathcal{G}_2)$ and $\mathcal{G}_Z$ $(\mathcal{G}_1 \times \mathcal{G}_2)$ are given by the edges of $\mathcal{G}_1 \times \mathcal{G}_2$ which join V to $C_1 \times V_2$ for $\mathcal{G}_X$ (the corresponding edges are denoted by solid lines) and which join V to $V_1 \times C_2$ for $\mathcal{G}_Z$ (the corresponding edges are indicated by dashed lines) [40]. 63

3.3 Graphical illustration of the hypergraph product structure: (a) The qubit subset $V_1 \otimes V_2$ is partitioned into $\{V_1 \otimes v_2^1, V_1 \otimes v_2^2, \ldots, V_1 \otimes v_2^{n_2}\}$, where $\{v_2^1, v_2^2, \ldots, v_2^{n_2}\} \in V_2$. (b) The qubit subset $C_1 \otimes C_2$ is divided into $\{c_1^1 \otimes C_2, c_1^2 \otimes C_2, \ldots, c_1^{r_1} \otimes C_2\}$, with $\{c_1^1, c_1^2, \cdots, c_1^{r_1}\} \in C_1$. An elementary canonical logical operator $(\bar{x}_k \otimes y_j, 0_{r_1 r_2})^T$ resides on the qubit subset $V_1 \otimes v_2^j$ if $y_j = v_2^j$. Similarly, the logical operator $(0_{n_1 n_2}, a_\ell \otimes \bar{b}_m)^T$ is localized on the subset $c_1^i \otimes C_2$ if $c_1^i = a_\ell$. 65

3.4 The hypergraph product construction of the Toric code from two repetition codes (periodic boundary conditions). (a) Delineates the two subsets of qubits, $V_1 \otimes V_2$ and $C_1 \otimes C_2$. (b) Show the logical X operator (blue circles) and the logical-Z operator (red circles) within the subset $V_1 \otimes V_2$. (c) Showcasing the logical X (blue circles) and Z (red circles) operators within the subset $C_1 \otimes C_2$. 66

4.1 A composite system consisting of a 2D quantum repetition code (Region A) and a 1D quantum repetition code arranged in a snake pattern (Region B). The overall code exhibits an energy barrier determined by the 2D quantum repetition code, which scales as the linear dimension of Region A. However, confinement properties are influenced by the 1D quantum repetition component, resulting in the absence of confinement for the composite system. 91

4.2 The 3D toric code can be structured as a cubic lattice (with periodic boundary conditions). From left to right, the figure illustrates its decomposition into a-slices, b-slices, and c-slices. The 3D Toric code encodes 3 logical qubits, with corresponding 3 logical Z operators. Each Z elementary canonical logical operator is supported on a single slice, with slices of the same color representing logical operators equivalent up to stabilizers. Generally, given any 3D hypergraph product code, one can create such a decomposition, although the structure within each slice will be more complicated and depends on the underlying classical codes. 108

CHAPTER 1

Introduction to quantum error correction

Quantum computing is considered to offer the ability to solve certain problems that cannot be solved by conventional computers. The introduction of quantum computing is credited to Feynman [1], who designed it to simulate quantum effects effectively. The advantages of quantum computers are thought to arise from the properties of a quantum system, such as superposition and entanglement. Notable examples of quantum speedup include Shor's algorithm [2], the Deutsch-Jozsa algorithm [3], and Grover's algorithm [4]. The theoretical potential of quantum computing inspires people to realize its applications.

Any two-level quantum-mechanical system can be used as a quantum bit (qubit). Various quantum systems have been developed to implement qubits, each with special strengths and challenges. Recently, numerous breakthroughs have been made in the fields of superconductors [5, 6], trapped ions [7, 8], and neutral atom qubits [9]. As these qubit implementations advance, they have enabled pioneering experiments to address the power of quantum computing. Recently, several experiments have been conducted to address quantum speedup by solving problems that are currently beyond the capabilities of classical computers [10, 11, 12, 13]. Those factors make quantum computation a promising field of study.

However, quantum computing has more challenges than conventional (classical) computing, particularly in the case of error correction. Typically, quantum bits are exposed to noisy environments, and there is a high probability of losing information about the qubits due to decoherence [14]. This fragility arises from the fundamental properties of entanglement. Thus, to harness the power of quantum computing, the error correction process is crucial [15]. In 1995, Shor and Steane pioneered the first quantum error correction (QEC) codes [16, 17].

This demonstrated that errors from faulty hardware can be identified and corrected, ensuring reliable storage of quantum information.

Quantum computing not only stores the quantum information but also processes it. In 1996, Shor proposed that quantum computation can be performed in a fault-tolerant manner [18], i.e., the error-corrected qubits perform better than physical qubits during computational processes. Based on Shor's work, threshold theorems have been proved for various error models [19, 20, 21, 22, 23], which states that by employing quantum error correction schemes, a quantum computer with a physical error rate below a specific threshold can reduce the logical error rate to arbitrarily low levels. The above achievements make it possible to build an extensive and performable quantum computer.

Since the introduction of quantum error correction and fault-tolerant quantum computing, significant developments have been made in the past three decades. Numerous promising quantum error correction codes with advantageous properties have been developed, along with improvements in the physical platforms for implementing these codes. However, difficulties remain in this field. This thesis addresses two main issues related to quantum error correction codes. The first issue is code state identification for the Toric code, discussed in Chapter 2. The second issue is the concept of the energy barrier of a quantum code, which determines logical robustness at finite temperatures and is essential in practice due to the inherent noise in qubits and quantum gates. Codes with a macroscopic energy barrier often exhibit desirable features, such as single-shot error correction and self-correcting properties. In Chapter 3, we examine the energy barrier of a specific quantum code, the hypergraph product code, followed by an analysis of its higher-dimensional case in Chapter 4.

In this introductory chapter, we provide an in-depth overview of quantum error correction to lay the basis for the issues discussed in the following chapters. We begin with classical error correction, establishing core concepts such as noise channels and code parameters. Next, we explain the differences between classical bits and quantum bits (qubits), as well as between classical and quantum errors. Building on this, we introduce the renowned stabilizer codes, a family of codes that includes the Toric code and hypergraph product codes. We also discuss the decoding process in quantum codes, as it aids in identifying the state of the code and

is also associated with the code's energy barriers. To complete the overview, we briefly introduce fault-tolerant schemes, showing how quantum computing remains viable despite noisy qubits and gates. Finally, we show the detailed structure of the thesis by listing the specific problems we aim to solve and the results we have achieved.

1.1 Classical error correction

Error correction is essential for communications and storage reliability. Classical memory systems, such as hard drives, are designed to be error-free through robust electronic design. However, the robustness of the information can be challenged during the transport process due to the various noises and interference inherent in the communication channels. Thus, classical error correction is typically used to detect and correct errors that occur during data transmission. This can be accomplished by adding redundancy to the original information. The redundancy allows the receiver to detect and correct errors via the decoder, thereby improving information reliability over noisy channels.

Classical error correction codes can be broadly classified into two types: block codes and convolutional codes. Block codes divide the message into fixed-size blocks, each of which is encoded independently. The well-known block codes include Hamming Codes [24], Reed-Solomon Codes [25], and Cyclic Redundancy Check (CRC) [26]. The other type is convolutional code [27], which encodes the entire message into a single long codeword, which is generated by passing the message through a sequence of shift registers.

To better understand the error correction process, we first introduce the concept of error channels in the following. These models effectively depict real-world errors.

1.1.1 Classical noisy channel

Generally, a channel is a model for describing the process of information transfer. A noisy channel can be characterized by a conditional probability distribution $p(y \mid x)$, which represents the probability of receiving the message y given the sent message x. Here, x belongs to

the input message set $\mathcal{X}$, and y belongs to the output message set $\mathcal{Y}$. The channel behavior can be fully described by the channel matrix $[p(y \mid x)]$, where each matrix entry provides the transition probability between input x and output y. The matrix dimensions are determined by the sizes of $\mathcal{X}$ and $\mathcal{Y}$.

For a given input distribution $p(x)$, the output distribution $p(y)$ can be computed by the relation:

$$p(y) = \sum_{x \in \mathcal{X}} p(y \mid x) p(x). \tag{1.1}$$

The channel capacity can then be analyzed, which is the maximum rate at which information can be reliably transmitted over the channel. Formally, the channel capacity C is defined as:

$$C = \max_{p(x)} I(X;Y), \tag{1.2}$$

where $I(X;Y)$ is the mutual information between the input X and the output Y, it can be calculated as:

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}, \tag{1.3}$$

and $p(x,y) = p(x)p(y \mid x)$ is the joint distribution of x and y.

The noisy channel theory, proposed by Shannon in 1948 [28], developed a theoretical framework to assess and improve communication systems in the presence of noise. In the following, we introduce the binary symmetric channel (BSC), binary asymmetric channel (BAC), and binary error channel (BEC) as examples.

1. Binary Symmetric Channel (BSC): The transition probabilities of BSC are given by a symmetric probability p of flipping each bit:

$$p(y \mid x) = \begin{cases} 1-p & \text{if } x = y \\ p & \text{if } x \neq y \end{cases}. \tag{1.4}$$

where $x, y \in \{0, 1\}$.

2. Binary Asymmetric Channel (BAC): Instead of symmetric probability for flipping each bit, in a BAC, the bit-flip error probabilities differ for 0 and 1. Let p be the probability that a

0 turns into a 1 and q be the probability that a 1 turns into a 0 in the channel. The transition probabilities of BAC are:

$$p(y \mid x) = \begin{cases} 1-p & \text{if } x=0, y=0 \\ p & \text{if } x=0, y=1 \\ 1-q & \text{if } x=1, y=1 \\ q & \text{if } x=1, y=0 \end{cases}. \tag{1.5}$$

3. Binary Erasure Channel (BEC): In BEC, instead of flipping those bits, it just erases them with probability ϵ. A symbol represents an erased bit "?", and the transition probabilities are:

$$p(y \mid x) = \begin{cases} 1-\epsilon & \text{if } x=y \\ \epsilon & \text{if } y=? \end{cases}. \tag{1.6}$$

1.1.2 Classical error correction properties

Detectability: An error E is detectable by a code if and only if for two distinct codewords x and y, we have $Ex \neq y$. (For instance, in a repetition code, any single- or double-bit flips are detectable.)

Correctability: Consider a code C and a set of error operators $\mathcal{E} = \{E_0 = I, E_1, E_2, \ldots\}$. This set $\mathcal{E}$ is said to be correctable by C if and only if for all distinct codewords $x, y \in C$ and for all indices i, j, the condition $E_i x \neq E_j y$ is satisfied. (In the repetition code, any single-bit flip is correctable.)

The criteria for correcting and detecting errors are interrelated under the assumption that each error operator is invertible. Specifically, correctability is equivalent to $E_j^{-1}E_i$ being detectable for all i, j. If $E_j^{-1}E_i x = y$ for $x \neq y$, then it follows that $E_i x = E_j y$, making a clear correction is impossible. Conversely, if $E_j^{-1}E_i x \neq y$ for all i, j and $x \neq y$, then $E_i x \neq E_j y$, implying that no two errors will map distinct codewords to the same state, thereby ensuring the code's correctability. Thus, to correct t errors, at least $2t$ errors must be detectable.

1.1.3 The repetition code

In coding theory, a code is a set of symbols or sequences (often binary) designed to represent information in a way that allows for reliable communication or storage, even in the presence of noise or errors.

Consider the BSC with an error probability p. The reliability of information can be improved by employing a repetition strategy. For example, by repeating each bit three times, the error tolerance is increased:

$$0 \to 000; \tag{1.7}$$

$$1 \to 111. \tag{1.8}$$

Here, 000 and 111 serve as the encoded codewords. The decoding algorithm is based on a majority vote; the original information is determined by the majority of the bits. Thus, this repetition encoding can tolerate a single-bit error. However, it cannot deal with the cases where two or three bits are flipped. Thus, the probability of success of decoding, denoted as p_{error}, is given by:

$$p_{error} = \binom{3}{2} p^2(1-p) + 3p^3 = 3p^2 - 2p^3. \tag{1.9}$$

The utility of the error-correction scheme is justified if $p_{error} < p$, which simplifies to:

$$3p^2 - 2p^3 < p \Rightarrow 3p - 2p^2 < 1 \Rightarrow p < \frac{1}{2}. \tag{1.10}$$

Hence, the code proves to have an advantage when the bit error rate p is less than $\frac{1}{2}$. Extending this approach for a n-fold repetition code:

$$0 \to 0 \cdots 0 \ (n\ 0s); \tag{1.11}$$

$$1 \to 1 \cdots 1 \ (n\ 1s). \tag{1.12}$$

The overall error probability, p_{error}, is determined by:

$$p_{error} = \sum_{l > n/2} \binom{n}{l} p^l (1-p)^{n-l}. \tag{1.13}$$

The criterion for the effectiveness of the code, $p_{\text{error}} < p$, must be met for the repetition strategy to be considered beneficial. In this case, we encode a 1-bit information into a n-bit information; we denote such a code as a $[n, 1]$ code.

1.1.4 Code parameters: rate and distance

The code rate denotes the efficiency of the code encoding performed. It is defined as the ratio of the number of original bits to that of encoded bits, with formula $R = \frac{k}{n}$, where k and n correspond to the number of bits in an original word and that in the codeword, respectively. In particular, the reciprocal rate, $r = \frac{n}{k}$, is also called the code overhead. A higher code rate means better coding but usually gives less robustness against errors since there are fewer bits available for error correction.

The distance of a code or Hamming distance is the minimum number of symbol changes required to change one valid codeword to another. The code distance is important in determining the capability for error detection and correction. Formally, for code C, suppose that there are two codewords $c_i, c_j \in C$, then the Hamming distance $d(c_i, c_j)$ between these two codewords is defined as the number of positions at which the corresponding symbols differ. The distance d of the code C itself is the minimum of all such distances between distinct codewords,

$$d = \min_{c_i, c_j \in C, c_i \neq c_j} d(c_i, c_j).$$

This distance d of the code is directly related to its error-correcting properties. A code can detect up to $d - 1$ errors in a codeword and correct up to $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors. Therefore, the larger the distance d, the greater the robustness of the code against errors. It should be noted that the ratio $\frac{d}{n}$ is known as the relative minimum distance (or simply the relative distance) of the code. The relative distance is a measure of the code's error-detecting and error-correcting capabilities in relation to its length. Higher relative distances generally indicate a stronger ability to detect and correct errors. We call a code with a constant rate and constant relative distance a good code.

1.1.5 Linear codes and Tanner graph

Linear codes are characterized by their linear properties in finite fields. A $[n, k]$ linear code C over a finite field $\mathbb{F}_q$ is a subspace of dimensions k of the vector space $\mathbb{F}_q^n$. This implies that every linear combination of codewords of C also belongs to C.

A linear code C can be defined using a matrix of size $k \times n$ G called the generator matrix, where each row of G represents a basis vector of the code. Any encoded codeword $\mathbf{c}_i$ in C can be generated by multiplying a message $\mathbf{m}_i$ of length k by the generator matrix,

$$\mathbf{c}_i = \mathbf{m}_i G, \tag{1.14}$$

where $\mathbf{m}_i \in \mathbb{F}_q^k$ and $\mathbf{c}_i \in \mathbb{F}_q^n$.

The parity check matrix H is a matrix $(n-k) \times n$ that provides a way to check the validity of a codeword. A vector $\mathbf{c}_i$ is in the codeword space of C if and only if it satisfies

$$H\mathbf{c}_i^T = \mathbf{0}^T, \tag{1.15}$$

where $\mathbf{0}^T$ is an zero vector. The row space of H is orthogonal to the row space of G. In another way, the code space of C is the set of vectors that are orthogonal to every row vector of H.

Given a linear code C, one can define a dual code $C^\perp$: If C is a linear code over a finite field $\mathbb{F}_q$ with length n, then $C^\perp$ consists of all vectors $\mathbf{v} \in \mathbb{F}_q^n$ such that:

$$\mathbf{v} \cdot \mathbf{c} = 0 \quad \text{for all } \mathbf{c} \in C.$$

Moreover, the parity check matrix H of C is the generator matrix of $C^\perp$, and vice versa. This relationship illustrates a fundamental duality in which the roles of the generating matrix and the parity check matrix are interchanged between C and $C^\perp$. Note that each codeword in the dual code $C^\perp$ is orthogonal to every codeword in C. If C is a $[n, k]$ code, then its dual $C^\perp$ will be a $[n, n-k]$ code.

A Tanner graph is a graphical representation of codes [29]. It is a bipartite graph consisting of two sets of nodes: variable nodes and check nodes. If H is the parity check matrix of a linear code, where H is an $m \times n$ matrix over the binary field $\mathbb{F}_2$, then the Tanner graph has n variable nodes (one for each column of H) and m check nodes (one for each row of H). An edge connects the variable node j to the check node i if the entry h_{ij} in the parity check matrix H is 1. Tanner graph is helpful for decoding, especially for decoding a specific family of code called the low-density parity check code, which we will discuss in the following.

1.1.6 LDPC code

The low-density parity-check (LDPC) code [30] is a type of linear code with sparse parity check matrices, which means its parity check matrix H contains only $O(1)$ number of nonzero entries in both columns and rows. This sparsity allows for the efficient implementation of decoding algorithms, notably the belief propagation algorithm [31].

One advantage of LDPC codes is that they can be designed to closely meet the Shannon limit. The Shannon limit is the theoretical maximum rate at which information can be transmitted over a noisy channel with an extremely low error rate. This has made LDPC codes popular in modern communication systems, including satellite communication, digital television broadcasting, and data storage.

1.2 Quantum error correction (QEC)

1.2.1 Quantum state and qubits

A quantum state is typically represented by a state vector $|\psi\rangle$ in a Hilbert space $\mathcal{H}$. For a system with a finite number of states, the state vector $|\psi\rangle$ can be expressed as:

$$|\psi\rangle = \sum_i c_i |i\rangle \tag{1.16}$$

where $|i\rangle$ are basis vectors of the Hilbert space, and c_i are complex coefficients satisfying the normalization condition $\sum_i |c_i|^2 = 1$.

A more general formulation, called density matrices, is used to describe both pure states and mixed states. A density operator, ρ for a quantum system, is a positive semidefinite operator with trace one acting on the Hilbert space of the quantum system. It is defined as

$$\rho = \sum_k p_k |\psi_k\rangle \langle\psi_k| \tag{1.17}$$

where $|\psi_k\rangle$ are state vectors representing the pure states in the system, p_k are probabilities associated with corresponding state $|\psi_k\rangle$, and $\langle\psi_k|$ is the conjugate transpose of $|\psi_k\rangle$. This representation ensures that ρ is Hermitian ($\rho = \rho^\dagger$), positive semi-definite, and has trace 1 $(\mathrm{Tr}(\rho) = 1)$.

For a pure state $|\psi\rangle$, the density operator simplifies to:

$$\rho = |\psi\rangle\langle\psi|, \tag{1.18}$$

which indicates that the system is just in the state $|\psi\rangle$.

As mentioned before, qubits differ from classical bits because of two fundamental quantum mechanical properties: superposition and entanglement. Superposition means that a qubit can exist simultaneously in the states $|0\rangle$ and $|1\rangle$ with different probabilities.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1.19}$$

where $|\alpha|^2$ and $|\beta|^2$ are probabilities of the qubit being in the $|0\rangle$ or $|1\rangle$ state, respectively. They satisfy the condition $|\alpha|^2 + |\beta|^2 = 1$. The superposition allows quantum computation to process many calculations simultaneously and is considered to be the resource that provides exponential speedup in certain computational tasks.

Another distinction between qubits and bits is the quantum no-cloning theorem [32, 33], which states that creating a copy of an unknown quantum state is impossible. This ensures the security of quantum information.

THEOREM 1.2.1 (No-cloning). *For any unknown quantum state $|\psi\rangle$. There does not exist a unitary operator U_c, such that*

$$U_c|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle. \tag{1.20}$$

PROOF. Assume that such an operator U_c exists and works for all states. For two general states $|\psi\rangle$ and $|\phi\rangle$,

$$U_c|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle \tag{1.21}$$

$$U_c|\phi\rangle|0\rangle = |\phi\rangle|\phi\rangle. \tag{1.22}$$

Then we have $\langle\phi \mid \psi\rangle = (\langle\phi \mid \psi\rangle)^2$, for any two arbitrary quantum states, which is not true. □

Entanglement also sets qubits apart from classical bits. When qubits are entangled, the measurement of one qubit will instantaneously influence the state of the other. This phenomenon challenges traditional beliefs about the separability and independence of the classical world, and it is crucial for quantum communication protocols.

1.2.2 Qubit error

Qubit errors can be sorted into three types, namely Pauli errors: X (bit-flip), Z (phase-flip), and Y (both bit-flip and phase-flip). This classification is based on the single Pauli matrices described above.

X Type Error (Bit-Flip Error): The X-type error, known as the bit-flip error, is like the classical bit-flip. This means that a qubit is flipped from $|0\rangle$ to $|1\rangle$ or vice versa. This error is represented by the Pauli X matrix. One can show that when the X operator is applied to a qubit in the state $|0\rangle = (1, 0)^T$ or $|1\rangle = (0, 1)^T$, it swaps out these states $X|0\rangle = |1\rangle, X|1\rangle = |0\rangle$. Moreover, in a superposition state $\alpha|0\rangle + \beta|1\rangle$, the action of an X error is

$$X(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle. \tag{1.23}$$

Z Type Error (Phase-Flip Error): The Z-type error, or phase-flip error, affects the phase of the qubit. The Z operator is applied to the basis states $|0\rangle = (1, 0)^T$ and $|1\rangle = (0, 1)^T$. It keeps $|0\rangle$ unchanged but adds a phase sign of -1 to $|1\rangle$, $Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle$. For a superposition state $\alpha|0\rangle + \beta|1\rangle$, the action of a Z error results in

$$Z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle. \tag{1.24}$$

1.2.3 Quantum error channels and superoperators

1.2.3.1 Superoperators

Loosely speaking, Superoperators are operators on operators. Normally, it operates on density operators of the system. They can be used to describe interactions with an environment (decoherence), measurements, or other kinds of quantum operations. Superoperators are defined as completely positive, trace-preserving (CPTP) maps. When the input is a quantum state ρ, the output is another quantum state $S(\rho) = \rho'$. To be valid, superoperators must follow these properties:

1. Linearity: $S(a\rho + b\sigma) = aS(\rho) + bS(\sigma)$. It makes sure that superoperators follow the principles of superposition.

2. Hermiticity-Preserving: Superoperators must map Hermitian operators to Hermitian operators. This is because the eigenvalues of Hermitian operators represent observable quantities.

3. Positivity: $S(\rho) \geq 0$. This condition makes sure that the output state ρ' does not have negative eigenvalues. This maintains the probabilistic interpretation of quantum states.

4. Complete Positivity: For S which acts as $S_{AB} = S_A \otimes I_B$ in a composite system ρ_{AB}, $S_{AB}(\rho_{AB}) \geq 0$. This guarantees that the superoperator does not create non-physical states when used on the part of an entangled system.

Formally, a superoperator S can be expressed in the Kraus form as:

$$S(\rho) = \sum_{\mu=1}^{N} M_\mu \rho M_\mu^\dagger, \tag{1.25}$$

where $\{M_\mu\}$ are Kraus operators. These operators satisfy

$$\sum_{\mu=1}^{N} M_\mu^\dagger M_\mu = I, \tag{1.26}$$

so that $\mathrm{tr}(S(\rho)) = \mathrm{tr}(\rho) = 1$.

In the following, we will use the form of superoperators to describe certain quantum noisy channels.

1.2.3.2 Depolarizing channel

The depolarizing channel can be mathematically expressed as:

$$S(\rho) = \rho' = (1-p)\rho + \frac{p}{3}\left(X\rho X^\dagger + Y\rho Y^\dagger + Z\rho Z^\dagger\right), \tag{1.27}$$

p is a real number in the interval $0 < p < 1$, representing the probability of depolarization. For a single qubit, this channel shows that there is a probability of $(1-p)$ the states to stay the same and a $p/3$ chance for errors of type X, Z, and Y, respectively.

In the Kraus representation, the associated Kraus operators are defined as:

$$M_0 = \sqrt{1-p}I, \quad M_1 = \sqrt{\frac{p}{3}}X, \quad M_2 = \sqrt{\frac{p}{3}}Y, \quad M_3 = \sqrt{\frac{p}{3}}Z. \tag{1.28}$$

It can be verified that $\sum_{k=0}^{3} M_k^\dagger M_k = I$.

For a geometric description, consider a qubit in an arbitrary state $\rho = \frac{1}{2}(I + \vec{p}\cdot\vec{\sigma})$, with the polarization vector $\vec{p} = p_x\hat{x} + p_y\hat{y} + p_z\hat{z}$. In cases of mixed states, $0 \leq |\vec{p}| < 1$, and for pure states, $|\vec{p}| = 1$. Consider the case where $\vec{p}$ along the $\hat{z}$ axis, thus $\rho = \frac{1}{2}(I + pZ)$. Depolarization then modifies the state to

$$S(\rho) = \frac{1}{2}I + \frac{p'}{2}Z, \tag{1.29}$$

where $p' = \left(1 - \frac{4p}{3}\right) p$. This effect reduces the length of the polarization vector. For the process to remain physical (ensuring $p' \geq 0$), p must satisfy $p \leq \frac{3}{4}$.

1.2.3.3 Phase damping

The phase damping channel is another important model in quantum information theory. It affects a single qubit without energy loss. Mathematically, this channel is described by:

$$S(\rho) = \rho' = \left(1 - \frac{p}{2}\right) \rho + \frac{p}{2} Z \rho Z^{\dagger}. \tag{1.30}$$

where p is the probability parameter.

The corresponding Kraus operators are

$$M_0 = \sqrt{1-p} I, \quad M_1 = \sqrt{p} |0\rangle \langle 0|, \quad M_2 = \sqrt{p} |1\rangle \langle 1|. \tag{1.31}$$

Also, $\sum_{k=0}^{2} M_k^{\dagger} M_k = I$.

This channel changes a density matrix ρ to

$$S(\rho) = \left(1 - \frac{p}{2}\right) \rho + \frac{p}{2} Z \rho Z = \begin{pmatrix} \rho_{00} & (1-p)\rho_{01} \\ (1-p)\rho_{10} & \rho_{11} \end{pmatrix} \tag{1.32}$$

This channel reduces the off-diagonal elements of the density matrix, illustrating its role in dephasing or decoherence. Coherence, which is the off-diagonal element, decreases over time.

Considering a continuous time case where the probability of scattering per unit time δt is small ($\Gamma \delta t = p \ll 1$), the effect of the channel over a time $t = n \delta t$ is

$$S_{\delta t} \circ \cdots \circ S_{\delta t}(\rho) = \begin{pmatrix} \rho_{00} & e^{-\Gamma t} \rho_{01} \\ e^{-\Gamma t} \rho_{10} & \rho_{11} \end{pmatrix} \tag{1.33}$$

where the off-diagonal terms exponentially decay with the dephasing rate Γ, capturing the total dephasing effect over time.

1.2.3.4 Amplitude damping

Amplitude damping is a model that describes the decay of a quantum system into an environment. This model can be described by a unitary interaction between the system A and the environment E. For the ground state, the system remains unchanged

$$|0\rangle_A|0\rangle_E \xrightarrow{U_{AE}} |0\rangle_A|0\rangle_E. \tag{1.34}$$

For the excited state, the system transitions to a superposition

$$|1\rangle_A|0\rangle_E \xrightarrow{U_{AE}} \sqrt{1-p}|1\rangle_A|0\rangle_E + \sqrt{p}|0\rangle_A|1\rangle_E. \tag{1.35}$$

The corresponding Kraus operators for this process are

$$M_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{pmatrix}, \quad M_1 = \begin{pmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{pmatrix}. \tag{1.36}$$

It transforms the density matrix ρ as

$$S(\rho) = \begin{pmatrix} \rho_{00} + p\rho_{11} & \sqrt{1-p}\rho_{01} \\ \sqrt{1-p}\rho_{10} & (1-p)\rho_{11} \end{pmatrix} \tag{1.37}$$

In a continuous-time description where the decay rate is Γ, and $\Gamma\delta t = p \ll 1$, the map after time $t = n\delta t$ is represented by:

$$S(\rho) = \begin{pmatrix} 1 - e^{-\Gamma t}\rho_{00} & e^{-\Gamma t/2}\rho_{01} \\ e^{-\Gamma t/2}\rho_{10} & e^{-\Gamma t}\rho_{11} \end{pmatrix}. \tag{1.38}$$

The coherence decays at rate $\Gamma/2$.

Experimentally, characteristic decoherence times are defined as: T_1 is a time such that $e^{-\Gamma_{decay}T_1} = 1/e$, thus $T_1 = \frac{1}{\Gamma_{\text{decay}}}$. This is the decoherence time due to amplitude damping. T_2 incorporates both the amplitude and phase damping, given by $T_2 = \frac{1}{\Gamma_{\text{dephase}} + \Gamma_{\text{decay}}/2}$, is the decoherence time due to the amplitude and phase damping.

1.2.4 Correting quantum errors

Like protecting classical information, quantum error correction (QEC) involves the initial encoding of a quantum state, denoted $|\phi_{in}\rangle$, into a logical quantum code. This encoded state then goes through a noisy channel that introduces errors from both environmental and control interactions. The process ends with a decoding process that gives the output state $|\phi_{\text{out}}\rangle$.

Encode quantum information. Qubits are mapped into a higher-dimensional Hilbert space using more physical qubits. This encoding process distributes the information of one logical qubit across many physical qubits, making a quantum error-correcting code. For example, Shor code[16], Steane code [34], and the more general class of stabilizer codes[35].

Instead of only protecting against X type errors, quantum error-correcting codes are designed to protect against both X and Z type errors. Moreover, if qubits are in a superposition, a direct measurement will lead to the collapse of the superposition. Quantum error correction requires indirect measurements to determine the existence and type of errors. In the stabilizer formalism, which we will discuss later, this can be done through the measurement of error syndromes of stabilizers, which are non-destructive and do not reveal the encoded information. In addition, syndromes are measured by introducing auxiliary qubits (ancilla) that interact with the code qubits, and these ancilla qubits are measured.

Based on the measurements of the syndrome, a classical decoding algorithm determines the most likely error that has occurred. The corresponding quantum operation is then applied to correct this error, restoring the quantum state. Correction operations use Pauli gates that reverse the effects of detected errors.

1.2.5 Necessary and sufficient conditions for QEC

Consider a single-qubit system S interacting with an environment E, initialized to state $|0\rangle_E$. The unitary evolution U_{SE} affects the system and environment as follows:

$$\begin{aligned} U_{SE} : |0\rangle_S|0\rangle_E \rightarrow |0\rangle_S |e_{00}\rangle_E + |1\rangle_S |e_{01}\rangle_E \,, \\ U_{SE} : |1\rangle_S|0\rangle_E \rightarrow |0\rangle_S |e_{10}\rangle_E + |1\rangle_S |e_{11}\rangle_E \,, \end{aligned} \tag{1.39}$$

where the states $\{|e_{ij}\rangle_E\}$ in the environment are not necessarily orthonormal. For a qubit initially in a superposition state $a|0\rangle + b|1\rangle$, the unitary interaction yields:

$$U_{SE} : (a|0\rangle + b|1\rangle)|0\rangle_E \rightarrow a|0\rangle_S |e_{00}\rangle_E + a|1\rangle_S |e_{01}\rangle_E + b|0\rangle_S |e_{10}\rangle_E + b|1\rangle_S |e_{11}\rangle_E \,, \tag{1.40}$$

which can be decomposed using the basis $\{I, X, Y, Z\}$ for qubit operators:

$$= (I_2|\psi\rangle) |e_I\rangle_E + (X|\psi\rangle) |e_X\rangle_E + (Y|\psi\rangle) |e_Y\rangle_E + (Z|\psi\rangle) |e_Z\rangle_E \,, \tag{1.41}$$

where $|\psi\rangle = a|0\rangle + b|1\rangle$ and the environmental states $\{|e_\alpha\rangle_E\}$ corresponding to $\alpha = \{I, X, Y, Z\}$ are also not necessarily orthonormal.

For a system of n qubits, any operator can be represented as a combination of operators from the expanded basis $\overline{\mathcal{E}} = \{I, X, Y, Z\}^{\otimes n}$, leading to 4^n linearly independent operators. The interaction with the environment is then generally expressed as:

$$U_{SE} : |\psi\rangle|0\rangle_E \rightarrow \sum_{a=0}^{4^n-1} E_a|\psi\rangle |e_a\rangle_E \,, \tag{1.42}$$

where each E_a is a unitary operator from $\overline{\mathcal{E}}$, and the environmental states $\{|e_a\rangle_E\}$ are not orthonormal.

To find errors, one first needs to identify a subset $\mathcal{E} \subset \overline{\mathcal{E}}$, which includes correctable errors. Error analysis and correction are performed by measuring all n qubits, identifying the error E_a that occurred, and then applying the correction operation $E_a^{-1} = E_a^\dagger$. This error correction strategy relies on the ability to tell apart different induced environmental states linked to errors in $\mathcal{E}$. For the recovery of quantum information, conditions must be met by quantum error correction codes.

Necessary Condition:

For the effective correction of errors, the inner product between different codewords, transformed by any two error operators E_a and E_b from the set $\mathcal{E}$, must be zero if the codewords are different:

$$\left\langle j_L \left| E_b^\dagger E_a \right| i_L \right\rangle = 0 \quad \text{if} \quad i \neq j \tag{1.43}$$

This condition makes error identification clear. Overlapping error effects would hide which error happened.

Sufficient Condition:

A stronger requirement for error correction is

$$\left\langle j_L \left| E_b^\dagger E_a \right| i_L \right\rangle = \delta_{ab}\delta_{ij} \tag{1.44}$$

This means that each error moves a codeword into a different subspace, which helps find and fix errors.

Combined Necessary and Sufficient Conditions:

The relation can be summarized as

$$\left\langle j_L \left| E_b^\dagger E_a \right| i_L \right\rangle = C_{ba}\delta_{ij}, \tag{1.45}$$

where C_{ba} is a part of a Hermitian matrix that does not depend on the codewords. This makes sure that error states are different for different codewords. This orthogonality prevents the decoherence of information during measurement.

1.2.6 Stabilizer codes

The Schrödinger and Heisenberg pictures give the same way to follow quantum states in quantum mechanics. Specifically, the Schrödinger equation describes state evolution as $|\psi(t)\rangle = U(t)|\psi(0)\rangle$, with measurements shown by $\langle\psi(t)|M|\psi(t)\rangle$. The Heisenberg picture

expresses the time evolution $M(t) = U^{\dagger}(t)MU(t)$ and the corresponding measurements $\langle\psi(0)|M(t)|\psi(0)\rangle$.

Stabilizer formalism provides another way to track quantum states. For example, the state $|\psi\rangle = |000\rangle$ is $+1$ co-eigenstate of Z_1, Z_2, and Z_3, we say $|\psi\rangle$ is stabilized by the operator set $S = \{Z_1, Z_2, Z_3\}$. These operators are orthogonal and commute with each other. State evolution under a unitary transformation U gives a new set of stabilizers, $S' = USU^{\dagger}$.

Stabilizer codes [35] were conceptualized by Daniel Gottesman in 1996. Stabilizer codes encode a quantum state in the $+1$ co-eigenspace of a set of m commuting Pauli operators, $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$, defined on an n-qubit Hilbert space $\mathcal{H}_2^{\otimes n}$. Each operator, S_i, can be any tensor product of the Pauli matrices $\{I, X, Y, Z\}^{\otimes n}$, with the requirement that all these operators commute, i.e., $[S_i, S_j] = 0$ for any i, j. It is crucial that $\mathcal{S}$ excludes the operator $-I$.

An $[[n, k]]$ stabilizer code encodes k logical qubits into n physical qubits, with code rate k/n and stabilizer generators $\{S_1, S_2, \ldots, S_m\}$, where $m = n - k$. The code space is the $+1$ co-eigenstate of these stabilizer generators. The code distance d is defined as the minimum number of Pauli operations needed to transform one codeword into another, enabling correction of up to $t = \lfloor \frac{d-1}{2} \rfloor$ errors. Error detection is feasible for any error $E_i \in \mathcal{E}$ as long as E_i is not in the centralizer $\mathcal{Z}(S)$, which includes the stabilizer generators and logical operators that commute with every element of $\mathcal{S}$.

Calderbank-Shor-Steane (CSS) codes [36, 34] are an important type of stabilizer code. A quantum CSS code is constructed from two classical linear codes C_1 and C_2 where $C_2 \subseteq C_1$. The minimum distance d of a CSS code is determined by the minimum distances d_1 and d_2 of the classical codes C_1 and C_2. CSS codes have a set of stabilizer generators that are either X-type or Z-type Pauli operators. Moreover, all the X type stabilizers can be written in a parity check matrix H_X, and all the Z type stabilizers can be written as a parity check matrix H_Z. Finally, the parity check matrix for the CSS code is

$$H = \begin{pmatrix} H_X \\ H_Z \end{pmatrix}. \tag{1.46}$$

The distance d of a quantum error correction code is defined as the minimum weight (i.e., the smallest number of nonidentity Pauli errors) that results in an undetectable error. Similarly to the classical case, a quantum error correction code characterized by parameters $[[n, k, d]]$- where n is the number of physical qubits (block size), k is the number of encoded logical qubits, and d is the code distance, then one can correct up to $\frac{d-1}{2}$ errors.

1.2.7 Steane code

Introduced by Andrew Steane in 1996 [17], the Steane code is a CSS code. It is a $[[7, 1, 3]]$ quantum code, which encodes one logical qubit into seven physical qubits and corrects arbitrary single-qubit errors. It is the smallest qubit CSS code used to correct a single-qubit error. The code is constructed using the classical $[7, 4, 3]$ Hamming code to protect against both X and Z errors.

The code's stabilizer generator matrix blocks H_X and H_Z are both the parity-check matrix for the $[7, 4, 3]$ Hamming code,

$$H_X = H_Z = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \tag{1.47}$$

The stabilizer group for the Steane code has six generators, three X-type and three Z-type, $IIIXXXX, IXXIIXX, XIXIXIX, IIIZZZZ, IZZIIZZ, ZIZIZIZ$. The logical codewords are

$$\begin{aligned} |\overline{0}\rangle =& \frac{1}{\sqrt{8}}(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle \\ &+ |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle) \\ |\overline{1}\rangle =& \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle \\ &+ |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle). \end{aligned} \tag{1.48}$$

1.2.8 Quantum LDPC code

The quantum LDPC code is a stabilizer code with a sparse parity check matrix; see [37, 38] for recent reviews. The sparsity parameters are w_c and w_q, which are the maximum weights of the rows and columns of the parity check matrix, respectively. These represent the maximum weight among all the checks and the maximum number of checks associated with a single bit. A code is LDPC if $w_c, w_q = O(1)$.

This recent interest in quantum LDPC codes is in part due to Gottesman, who showed that with quantum LDPC codes, one can achieve constant overhead for fault-tolerant quantum computation [39]. A well-known approach to constructing such codes is the hypergraph product construction [40]. More recent studies led to the development of other families of quantum LDPC codes with improved parameters[41, 42, 43, 44, 45, 46, 47, 48, 49]. In 2022, Panteleev and Kalachev presented the first construction of good quantum LDPC codes[47], characterized by both a constant rate and a constant relative distance. Similar outcomes have been reported in [48, 49]. Moreover, recent studies suggest that viable fault-tolerant quantum computing architectures can host such codes [50, 5, 51].

1.2.9 Decoder

Quantum decoding processes work to find and correct errors while keeping the quantum information intact. Typically, decoders analyze error syndromes generated by measuring stabilizers. After that, they figure out which error is the most probable one to have happened.

Usually, a decoding process has three steps. First is the syndrome measurement, which is used to reveal information about the error without accessing the encoded quantum information. This can be done by measuring the stabilizers. Second, the measurements should be analyzed to determine the type and location of the error. This analysis gives a map between the syndrome outcomes and the possible errors, and the map depends on the stabilizer structure and the decoding strategy. Finally, the original quantum state can be restored by implementing the corresponding quantum gates.

There are many obstacles to decoding quantum information. Noise models for quantum systems are complicated. Thus, advanced statistical methods are required to interpret error syndromes accurately. Moreover, an efficient and fast decoder is needed to correct the message before the information decays again because, usually, the decay occurs rapidly in a quantum system. In the following, we examine several well-known quantum decoders.

1. Minimal Weight Perfect Matching Decoder (MWPM)

A perfect matching of a graph is a matching (i.e., an independent edge set) in which every vertex of the graph is incident to exactly one edge of the matching. In the context of quantum error correction, when we deal with X and Z errors independently, which normally happens in the CSS quantum code, then one can draw a graph to show the relation between qubits and stabilizers. In this case, the edges represent the qubits, while the vertices are labeled with measurements of stabilizers. This graph is used to detect possible errors given the stabilizer measurement. The idea of the MWPM decoder is to find a minimal weight perfect matching of the graph.

One way to find the minimal weight perfect matching is the Blossom algorithm, commonly used in classical computing. For complexity, the Dijkstra step has runtime $O(N^2 \log(N))$, and the blossom step has complexity $O(N^3 \log(N))$. For the 2D matching graph of a N-qubits surface code, the runtime of the decoder is therefore dominated by the blossom algorithm with an overall runtime $O(N^3 log(N))$.

The underlying logic of the MWPM decoder is that the system's error rate is considered to be sufficiently small, so the most likely error pattern consistent with the syndrome is the one that has a minimal number of errors. That is the reason for finding a minimal weight for a given perfect match.

The MWPM decoder can decode Toric and surface codes[52], The subsystem surface code[53], 2D hyperbolic code[54], Subsystem hyperbolic codes[55], 3D toric and surface codes (for X errors), The XZZX surface code[56], Some compass codes[57], including the heavy hexagon code[58]. MWPM can also be used as a subroutine for single-shot decoding of the 3D toric code[59], the gauge color code[60], decoding the color code[61], fraction

topological codes[62], the Fibonacci code[63], The repetition code with noisy syndrome measurements[64].

2. Union-find decoder

The Union-Find decoder is based on the observation that erasure error decoding is more straightforward than Pauli errors. An erasure error can be seen as a Pauli error at an already-known location. If the only source of error is erasure, decoding is much easier since errors can only be found within erasure. The peeling decoder utilizes this property to identify corrections in cases where only erasure errors are present.

The Union-find decoder is thus composed of two stages. The first stage is to generate an erasure error, ϵ', from a syndrome caused by both Pauli errors and erasure, ϵ. In addition, the modified erasure must contain a valid correction operator. This stage is referred to as syndrome validation. After that, the peeling decoder can be applied to find a correction. To perform syndrome validation, the decoder identifies "invalid" clusters of erasures and then grows them iteratively until the erasure decoder can correct the updated state. To make the decoder faster, one can update the clusters as they grow rapidly. This dynamic update is the most significant factor contributing to the decoding complexity [65].

3. Maximum likelihood decoder

Given the error syndrome, the maximum likelihood decoder (MLD) is an algorithm designed to identify a correction operation that optimizes the probability of effective error correction. Consequently, MLD is, by definition, the most effective error correction algorithm for quantum codes and associated noise models.

In 2002, Dennis et al. established the initial formal definition of MLD for two-dimensional surface codes [52]. A pivotal finding in this research is that the computational challenge linked to MLD can be simplified to determine the partition function of a classical Ising-like Hamiltonian in a two-dimensional lattice. This finding has led to considerable investigation into the relationships between MLD and the statistical physics of disordered Ising-like Hamiltonians.

The foremost constraint of Maximum Likelihood Decoding (MLD) is its high computational complexity. As the system size increases, the calculation of error probabilities increases exponentially, making this method unfeasible for larger systems. Fortunately, insights from [52] have informed the development of efficient MLD. Several approximate algorithms that operate in polynomial time have been identified. Prominent examples are the renormalization-group decoder introduced in [66] and the Markov chain Monte Carlo method established in [67].

In the context of the two-dimensional surface code, it is important to emphasize that the Minimum Weight Perfect Matching (MWPM) decoder identifies the most efficient error chain that aligns with the given syndrome. While this method focuses solely on minimizing energy rather than the free energy, it operates effectively at zero temperature. Occasionally, the entropic component influences the decoding process. For example, two possible errors have identical minimum energy, but one should be chosen because it has higher entropy due to degeneracy. As a result, the success probability of MWPM is inferior compared to that of free-energy minimization. In Ref. [68], researchers demonstrated exactly that decoders for topological quantum error correction can be improved by accounting for degeneracy in matching algorithms. Therefore, the Maximum Likelihood Decoder (MLD), which aims to minimize free energy, is recognized as possessing a superior threshold in such scenarios.

4. Belief propagation decoder

The belief propagation (BP) algorithm (or the sum-product algorithm), first proposed by Judea Pearl in 1982 [31], is effective at decoding classical LDPC codes. Its running time is linear in the code's block length.

BP is an iterative algorithm that estimates the transmitted codeword by passing messages along the Tanner graph of the code. This Tanner graph consists of variable nodes (bits) and check nodes (checks). In the beginning, each variable node is assigned a log-likelihood ratio based on the received signal. In the decoding process, the variable nodes send updated belief messages to the connected check nodes and intentionally omit information from the recipient node. Conversely, check nodes send messages back to variable nodes based on parity constraints and incoming messages from other variable nodes. This belief flow continues iteratively, refining

the bit estimates until either convergence or a pre-set number of iterations is reached. Finally, decisions on the bit values are made based on the aggregated information, and the decoder checks if the estimated codeword satisfies all parity-check equations, indicating successful decoding.

Like classical LDPC codes, quantum LDPC codes use the traditional belief propagation (BP)-based decoding algorithm. However, due to degenerate errors, the standard BP algorithm alone does not achieve good decoding performance in quantum codes [69, 70]. These errors cause "split beliefs" and prevent the algorithm from converging. This issue can be resolved by incorporating a post-processing technique known as ordered statistics decoding (OSD). The OSD step uses the output of the probability distribution of BP to select a low-weight recovery operator that satisfies the syndrome equations.

The BP+OSD algorithm was first applied to quantum expander codes by Panteleev and Kalachev [71]. Following this, Roffe et al. revealed that the BP+OSD decoder has broader applicability across various quantum LDPC codes, which encompasses both two-dimensional surface codes and toric codes [72].

There are many other techniques, such as ambiguity clustering (BP+AC)[73] and localized statistics decoding (BP+LSD)[74], use matrix inversion methods to solve the decoding problem during post-processing. Although these inversion-based decoders achieve high error thresholds, they increase the worst-case runtime complexity to $O(n^3)$, significantly slower than the $O(n)$ runtime of standard BP. In contrast, the closed-branch decoder (BP+CB) reduces the worst-case runtime complexity but is less accurate than inversion-based decoders [75]. Recently, the belief propagation plus ordered Tanner forest (BP+OTF) algorithm has been proposed as an almost linear time decoder for quantum LDPC codes [76].

5. Local decoders

The decoders examined above rely on global classical data from measurement outcomes. Such global data requires communication across distant regions of the system. In practice, this communication can cause significant errors. Collecting and processing global syndrome

data also risks new failures [77, 78]. Therefore, it is vital to develop rapid decoders that utilize only local information to mitigate error accumulation.

Cellular automata (CA) provide a promising way to decode topological quantum codes [79, 80, 81]. These local decoders excel at parallel processing and can run on specialized hardware. They do not depend on long-range interactions. A simple CA technique, often called Toom's rule [82, 83], can protect quantum information in the 4D toric code on a hypercubic lattice [52]. Recent results [84, 85, 86] show that heuristic decoders using Toom's rule can reach nonzero thresholds in higher-dimensional toric codes.

From a statistical physics viewpoint, codes with a large energy barrier pair well with local decoders. This barrier requires substantial energy to apply a logical operator. When the temperature stays below a certain point, the system remains near one code state. Minor local errors occur, but the system does not jump to another state. Error reduction then becomes a local process, much like gradual cooling. The system settles itself. Hence, the energy barrier plays a key role in a successful error correction.

1.3 Universal quantum computation

Universal quantum computation refers to the ability of a quantum computing model to do any kind of computation. In classical computing, it is well-known that the NAND gate is universal. In the following, we first give a brief review of universal quantum computation.

Meanwhile, unlike classical computation, the quantum gate could also be noisy, and error protection should be considered not only for information storage or transformation, but also for computation. After that, we discuss fault-tolerant quantum computation, which is a scheme to perform reliable computation even with noisy quantum gates and measurements. This technology enables the scaling up of quantum computing.

1.3.1 Single qubit gate

The Pauli operators are a set of three 2×2 Hermitian and unitary matrices. These matrices are denoted by σ_x, σ_y, and σ_z, corresponding to the quantum gates X, Y and Z used in quantum computing, respectively.

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

The Pauli operators are crucial for quantum gates, and the reason is any 2×2 matrix O can be written down as a linear combination of I, X, Y, and Z

$$O = \alpha I + \beta X + \gamma Y + \delta Z, \tag{1.49}$$

where $\alpha, \beta, \gamma, \delta$ are complex coefficients.

Pauli operators satisfy the commutation relations $[\sigma_j, \sigma_k] = 2i\varepsilon_{jkl}\sigma_l$, where ε_{jkl} is the Levi-Civita symbol, and here the Einstein summation notation is used. Meanwhile, they also have anti-commutation relations $\{\sigma_j, \sigma_k\} = 2\delta_{jk}I$.

1.3.2 Clifford hierarchy

The Pauli group P_n on n qubits is defined as

$$P_n = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}^{\otimes n}. \tag{1.50}$$

The Clifford hierarchy is a framework that categorizes quantum gates based on their relationship to the Pauli group. In the following, we show this hierarchy.

Level 1: Pauli Group

The first level of the Clifford hierarchy $\mathcal{C}_1$ is the Pauli group itself. As mentioned above, it consists of the Pauli matrix X, Y, Z, and the identity I. These gates perform basic bit and phase flips on qubits, but do not entangle them.

Level 2: Clifford Group

The second level of the Clifford hierarchy, $\mathcal{C}_2$, is known as the Clifford group, which is defined as gates that map the Pauli group to itself. The Clifford group includes three gates: 1. The Hadamard gate H that maps X to Z and vice versa. 2. The phase gate S (or $\sqrt{Z}$) that maps X to Y and Y to $-X$. 3. The CNOT gate, which is a two-qubit gate that performs a controlled X operation on the target qubit based on the state of the control qubit. These gates can be written in the matrix form

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad S = \sqrt{Z} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad \text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{1.51}$$

As stated in the Gottesman-Knill theorem, the Clifford gates can be simulated efficiently on classical computers,

THEOREM 1.3.1 (Gottesman-Knill). *Quantum circuits comprised solely of Clifford gates, acting on computational basis states and incorporating measurements within the Pauli group P_n, can be efficiently simulated on classical computers.*

Therefore, to achieve the quantum speed up, the quantum circuit should consist of non-clifford gates.

Higher Levels: $\mathcal{C}_k$

The higher levels Clifford hierarchy, $\mathcal{C}_k$ for $k \geq 3$, include gates that are defined recursively: a gate U belongs to $\mathcal{C}_k$ if, when conjugating any gate of $\mathcal{C}_{k-1}$, it results in a gate of $\mathcal{C}_{k-1}$. These higher-level gates enable more complex quantum operations that cannot be efficiently simulated classically and also include gates for universal quantum computation, such as the level 3 T gate (or $\pi/8$ gate), which can be used to achieve universality in quantum computing

when combined with the Clifford group.

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}. \tag{1.52}$$

1.3.3 Solovay-Kitaev theorem

A universal quantum gate set is a set that can be used to provide any unitary operation with arbitrary accuracy. The Solovay-Kitaev theorem [19, 87] guarantees that any single-qubit quantum gate can be approximated to any desired precision with a sequence of gates from a dense set. In practice, this means that complex quantum algorithms that require operations beyond the basic set can still be efficiently implemented using sequences of these universal gates in that dense set.

THEOREM 1.3.2 (Solovay-Kitaev). *Let G be a finite set of elements in $SU(2)$, containing its own inverses such that $\langle G\rangle$ is dense in $SU(2)$ (dense means give a matrix g' of $SU(2)$ you can always find a element g in G, such that $D(g,g') < \epsilon$), and $\epsilon > 0$. Then any element $h \in SU(2)$ can be approximately simulated using $\ell = O(\log^c(\frac{1}{\epsilon}))$ gates from G, and here c is a constant ($c < 4$). We say approximately simulated to indicate that the distance between h and the simulated gates $\tilde{h}$ satisfies $D(h,\tilde{h}) \leq \epsilon$. And $\tilde{h} = \Pi_{j=1}^{\ell} g_j$ with $g_j \in G$. The distance metric*

$$D(h,\tilde{h}) = 2 \max_{|\psi\rangle:\langle\psi|\psi\rangle=1} \|(h,\tilde{h})\,|\psi\rangle\,\| \tag{1.53}$$

For a single qubit gate, $G = \{H, T\}$, one can show that $\langle G\rangle$ is dense in $SU(2)$. In addition, for arbitrary n qubit computation, the CNOT gate, together with the Hadamard and T gates, is universal for quantum computation.

1.4 Fault tolerant and threshold theory

Fault tolerance refers to the ability of a quantum computer to perform a reliable computation in noisy quantum gates, qubit decoherence, and other imperfections. The error threshold

refers to a critical error rate. When the total error rate is below this threshold, the computation benefits from the fault-tolerant structure. The mathematical framework for fault tolerance involves the use of fault-tolerant gadgets in which errors propagate in a controllable way. Formally, for a quantum system described by a set of qubits $|\psi\rangle$, fault-tolerant operations ensure that for any operation $\mathcal{O}$ applied to the system, the error-corrected state $\mathcal{E}(\mathcal{O}(|\psi\rangle))$ can closely approximate $\mathcal{O}(|\psi\rangle)$.

1.4.1 Noise and control model

The performance of a fault-tolerant quantum computing (FTQC) protocol heavily depends on its architectural assumptions. Here, we outline key assumptions; for more details, see [88].

1. **Nonincreasing error rate.** The error rate remains independent of the circuit size, which enhances the performance of fault-tolerant circuits over time.

2. **Parallel operation.** The rate of asymptotic parallel processing exceeds a constant multiple of the asymptotic error rate, allowing error correction to consistently outpace the occurrence of errors.

3. **Reusable memory.** The rate of erasing or replacing qubits asymptotically exceeds a constant multiple of the asymptotic error rate. This enables the computer to eliminate the entropy faster than it is produced by errors.

4. **Reliable classical computation.** Classical computations always return the correct result.

5. **Fast classical computation.** Classical computations are efficient.

6. **No qubit leakage.** Qubits always remain within the computational Hilbert space.

7. **Uncorrelated noise.** Each qubit and gate are affected by an independent noise source.

8. **Local quantum processing.** Gates can only connect adjacent qubits as depicted in their layout graph.

Finally, we make some remarks about the noise model. We have already discussed the qubit error channel, but it is only for quantum information transport. In the case of quantum computation with the quantum circuit model, except for the qubit error, one also has to consider the measurement error. For the qubit error model, as mentioned above, there are also varies, such as stochastic adversarial noise [21, 89, 90, 91], purely depolarizing noise [92], where the noise has a strong bias, phase flip is more significant than bit flips [93].

i) **Code capacity noise model.** This noise model considers only qubit errors and excludes measurement noise by assuming that all measurements are perfect. Consequently, the threshold provided by this model is the highest among all error models, such as 10.9% for the toric code [94]. In this model, various channels can describe qubit errors, including depolarizing and erasure channels.

ii) **Phenomenological noise model.** In this noise model, we need to account for both qubit errors and measurement errors. Measurement errors are defined straightforwardly by assigning a probability of error to each measurement outcome. This approach is generally sufficient for most cases. Normally, the measurement error results in a significantly lower threshold compared to only counting the qubit error. For example, the toric code exhibits a threshold of 3.3% [95].

iii) **Circuit level noise model.** In the circuit-level noise model, measurement errors are more accurately represented. Unlike the phenomenological noise model, which assigns the error probability to each measurement result, the circuit-level model considers the specific circuits that implement the measurements. The individual error model is applied to each single-qubit gate and two-qubit gate. Consequently, measurement errors arise due to gate noise. This noise model results in a lower error threshold for the toric code, approximately 1.1% [96].

1.4.2 Transversal gates and Eastin-Knill theorem

The transversal gate is defined as applying logical operators by applying single Pauli operators to corresponding qubits in each block of a quantum error-correcting code without interaction between qubits of different blocks. For example, a transverse gate T operates as $T =$

$T_1 \otimes T_2 \otimes \ldots \otimes T_n$, where each T_i acts independently on the corresponding qubit i in each block. This requirement of acting independently across different blocks ensures that errors do not spread among blocks.

The Eastin-Knill theorem [97] significantly restricts the power of transversal gates.

THEOREM 1.4.1 (Eastin-Knill). *No quantum error-correcting code can possess a transversal implementation for a complete set of gates that form a universal set of quantum gates.*

This theorem implies that other kinds of strategy must be employed to achieve universality in fault-tolerant quantum computing, such as magic state distillation [98], code deformation [99, 100], or non-transversal gate in the quantum circuit.

1.4.3 Concatenated codes and threshold

Concatenated codes [101] construct complex quantum error-correcting codes from simpler ones by layering them in a hierarchical structure. A basic quantum code $\mathcal{C}$ encodes k logical qubits into n physical qubits. In a concatenated scheme, each qubit of $\mathcal{C}$ is encoded using the same or another quantum code $\mathcal{C}'$. This process can be recursively applied to generate multi-layered encoding.

Formally, if $\mathcal{C}_1$ is a $[[n_1, k_1, d_1]]$ code and $\mathcal{C}_2$ is a $[[n_2, k_2, d_2]]$ code, the concatenated code $\mathcal{C}_1 \circ \mathcal{C}_2$ is a $[[n_1 \cdot n_2, k_1 \cdot k_2, d \leqslant d_1 d_2]]$ code. Concatenation is particularly effective for enhancing the fault tolerance of quantum systems by increasing the effective distance of the code.

In a fault-tolerant scheme, it can be shown that the probability of an unrecoverable error scales like $O(p^2)$, where p denotes the physical gate error rate. Suppose that the overall error rate for a circuit is cp^2, where c is a constant representing the property of the circuit. This error probability outperforms the unencoded error rate of p, as $cp^2 < p$ when $p < 1/c$. We define the threshold error rate as $\rho_{th} = 1/c$; encoding is beneficial below this rate.

For a quantum algorithm with $g(n)$ gates, where n indicates the size of the problem, to achieve a final accuracy of ϵ (i.e., $\|U_{ideal} - U_{actual}\| \leq \epsilon$), the error per logical gate should not exceed $\epsilon/g(n)$. This accuracy can be achieved through concatenation.

In a concatenated framework, each gate at the initial level is further encoded at the subsequent level. The effective error rate becomes $c(cp^2)^2$ with two concatenation levels. Extending this to k levels, the effective error rate is reduced to $(cp)^{2^k}/c$, which decreases doubly exponentially with k. To achieve the ϵ accuracy, one can pick k such that

$$\frac{(cp)^{2^k}}{c} \leq \frac{\epsilon}{g(n)}. \tag{1.54}$$

In quantum computing, concatenation allows for arbitrary precision with a price of higher resource usage (qubits and logical gates). This additional overhead can be measured by checking the size of the circuit that is used to implement logical gates with error correction. This is represented by the formula $d^k g(n)$, where d denotes the maximum number of operations per encoded gate.

To relate the error correction capabilities to accuracy ϵ and the number of logical gates $g(n)$, we derive $(cp)^{2^k}/c = \epsilon/g(n)$, leading to

$$2^k \log(cp) = \log\left(\frac{c\epsilon}{g(n)}\right), \tag{1.55}$$

then

$$k = \log\left(\frac{\log\left(\frac{c\epsilon}{g(n)}\right)}{\log(cp)}\right) = \log\left(\frac{\log\left(\frac{g(n)}{c\epsilon}\right)}{\log\left(\frac{1}{cp}\right)}\right). \tag{1.56}$$

Given $d^{\log_2 x} = x^{\log_2 d}$, we have

$$d^k = \left[\log\left(\frac{\log\left(\frac{g(n)}{c\epsilon}\right)}{\log\left(\frac{1}{cp}\right)}\right)\right]^{\log d}, \tag{1.57}$$

it can be simplified to

$$d^k = O\left(\text{poly}\left(\log\left(\frac{g(n)}{\epsilon}\right)\right)\right). \tag{1.58}$$

Thus, the overhead in terms of both the number of physical operations and the physical qubits is polylogarithmic in $g(n)/\epsilon$.

1.5 Difficulties in QEC and the structure of this thesis

In quantum computing, it is usually hard to initialize quantum error correction codes or determine their states. This is because preparing a quantum system in the ground state of a specific Hamiltonian is difficult due to the sensitivity of quantum states. Even small errors during initialization can corrupt the encoded information. Thus, it requires highly precise control technology and even fault-tolerant methods.

In Chapter 2, we demonstrate how to use a periodic driving process to identify the toric code phase. Under a specific driving term, the toric code phase is shown to exhibit a discrete time crystal (DTC) behavior. However, this DTC signal disappears as environmental errors accumulate. Moreover, by incorporating a decoder into the readout process, we show that the DTC phase becomes more robust.

In practice, quantum error correction requires a large amount of fault-tolerance overhead and fast decoding. One promising approach to overcome this challenge is to use codes with a macroscopic energy barrier. A macroscopic energy barrier is a necessary condition for self-correcting quantum memory, which reduces the need for syndrome measurements and thus improves fault-tolerance efficiency. Not only that, codes with macroscopic energy barriers are expected to have a simple process that iteratively reduces energy, serving as a decoder [102]. However, so far, rigorous bounds on the energy barrier have typically been derived only for specific codes[103, 104, 105, 106]. Determining a lower bound for the energy barrier of codes is generally challenging due to two reasons: there are too many possible error paths (sequences of local errors) that can implement a fixed logical operator, and the logical operator itself is defined only modulo stabilizers.

In Chapter 3, we establish tight bounds on the energy barrier of a family of quantum low-density parity-check (LDPC) codes constructed using hypergraph products, which is a flexible

framework for constructing quantum codes and offers numerous advantages [107, 108]. We accomplish this by first proving a general fact applicable to any quantum LDPC codes: equivalent logical operators up to stabilizers have the same energy barrier. Subsequently, we propose an approach that rigorously connects the energy barriers of quantum codes with the underlying classical codes. This chapter was published in Physical Review Letters [109].

The higher-dimensional hypergraph product [110] generalizes the hypergraph product to a higher dimension. These higher-dimensional codes tend to be more fault tolerant by offering better protection against errors. They possess numerous good properties, one of which is that the effective distance of those higher-dimensional codes matches their code-theoretic distance [111]. This extends the result that stabilizer measurement schedules preserve the distance of hypergraph product codes [112]. Moreover, higher-dimensional hypergraph product codes have efficient logical gates [113] and single-shot decoding [114, 59].

In Chapter 4, we demonstrate that higher-dimensional hypergraph product codes can potentially exhibit higher energy barriers than their classical component codes due to additional stabilizer redundancy. This enables the construction of quantum codes with macroscopic energy barriers from classical codes lacking such barriers. This approach offers a pathway toward discovering self-correcting quantum memories and may inform the design of local decoders. This chapter was accepted by Physical Review A [115].

CHAPTER 2

Topological ordered time crystal in Toric code

2.1 Introduction

Initializing quantum error correction codes and identifying their states offer significant challenges in quantum computing. Due to the sensitivity of quantum states, setting up a quantum system in the ground state of a particular Hamiltonian is hard. Minor inaccuracies during initialization can corrupt the encoded quantum information, necessitating precise control and fault-tolerant methods [116]. There are many papers that focus specifically on methods for initializing the well-known toric codes, and numerous challenges were encountered in this process [117, 118, 119, 120].

In this chapter, we focus on toric code [121] and show that through a periodic driving process, one can identify the toric code phase as a discrete time crystal. The non-local logical information of the toric code characterizes the time crystal. Furthermore, we show that a readout with a simple decoder makes the time crystal signal more robust.

State identification in quantum error correction involves measuring without disrupting the fragile quantum state. Techniques such as syndrome measurement are employed to detect errors indirectly, but these require a sophisticated implementation to avoid collapsing the quantum information [122].

In 2021, Ruben V. and colleagues explored the feasibility of realizing the toric code within a two-dimensional array of Rydberg atoms configured on a ruby lattice[123]. Their approach to identifying the phase involves observing a continuous transition between two featureless phases, measuring a topological entanglement entropy of $\ln 2$ across various geometries,

identifying degenerate topological ground states, and analyzing the expected modular matrix from ground-state overlap. This study provides motivation for further exploration of the dynamic properties and phase identification of the toric code through evolution.

Another key point of this work is exploring how a decoder might enhance or optimize the time crystal phase. By observing a nonlocal order parameter, here defined as logical information, in the evolution of the toric code, we show that a readout with a decoding process can yield improved results regarding these time crystal oscillations.

2.1.1 Toric code

The study of the toric code has substantially influenced theoretical and experimental research in quantum computing. Its conceptual simplicity and profound implications for fault tolerance make it a foundational model for exploring more complex quantum systems and realizing robust quantum computers.

2.1.1.1 Code construction

Consider a $L \times L$ square lattice defined on the torus for simplicity. By associating every edge of the lattice with a qubit, this lattice with periodic boundary conditions has $2L^2$ edges. Thus, a toric code defined on this torus has $n = 2L_xL_y$ qubits.

The toric code is a CSS stabilizer code. There are two types of stabilizer generators in the toric code. One is the plaquette operator $S_Z = \otimes_{\ell \in P} Z_\ell$, where P is the plaquette in the square lattice, which contains all Pauli Z in this plaquette. The other is the vertex generator $S_X = \otimes_{\ell \in V} X_\ell$, where V is the vertex in the lattice, as shown in Fig. 2.1. Notice that although the single Pauli operators X and Z are anti-commute with each other, there are always even numbers of overlapping Paulis for each plaquette operator and vertex operator. Thus, all stabilizer generators commute with each other, which means $[S_Z S_{Z'}] = [S_X, S_{X'}] = [S_Z, S_X] = 0$.

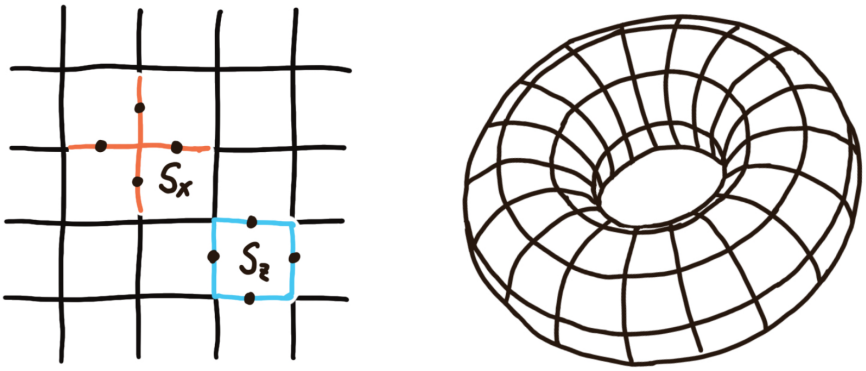


FIGURE 2.1: Stabilizers in the Toric code. Here, the lattice has a periodic boundary condition, and the stabilizer S_X (S_Z) is denoted in red (blue).

There are $L^2 - 1$ independent plaquette operators and $L^2 - 1$ independent vertex operators because $\Pi_P S_Z = I, \quad \Pi_V S_X = I$, the product of all S_X (or S_Z) stabilizers is indentity I. Recall that for the stabilizer code, denote n as the number of physical qubits, m as the number of independent stabilizers, then the number of logical qubits is $k = n - m$. For the toric code, $k = n - m = 2L^2 - 2(L^2 - 1) = 2$, thus it is a $[[n, 2]]$ code.

To find logical operators, we need to identify operators that commute with stabilizers but cannot be generated by the stabilizers themselves. Furthermore, the corresponding logical Z and logical X operators anti-commute with each other. In the toric code, a way to find such operators is to construct a loop that cannot be built by local stabilizers but goes around the entire surface of a torus.

We can define the logical Z operator as the Z loop that is topologically different from a local loop (stabilizers), and we can find that the two logical Z operators commute with stabilizers, Z_{L_1} and Z_{L_2}. Similarly, we can construct the logical X loop on a dual lattice like the logical Z operator; then, we have two corresponding logical X operators X_{L_1} and X_{L_2}. Furthermore, the logical Z operator defined on the lattice is anticommute with the logical X operator defined on the dual lattice as they overlap on a single qubit, as shown in Fig. 2.2.

$$\{X_{L_1}, Z_{L_1}\} = 0, \quad \{X_{L_2}, Z_{L_2}\} = 0 \tag{2.1}$$

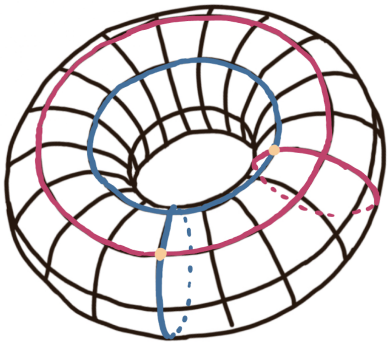


FIGURE 2.2: Logical operators in the toric code are represented by nontrivial circles on the torus.

Finally, notice that the code distance of the toric code is L, so the toric code is a $[[n = 2L^2, k = 2, d = L]]$ code.

2.1.1.2 Errors on the toric code

Consider a Z type error that occurred on a qubit; this error will commute with all the Z type stabilizers but anti-commute with two S_X stabilizers that contain this qubit. It is easy to determine where this error occurred. In the case of two Z Pauli errors, if these two errors are adjacent, the stabilizers that anti-commute with these two errors are two S_X stabilizers on the endpoint of the error line. But on the other hand, if we measure the stabilizers and find two S_X stabilizers that are flipped, then one cannot determine where the error occurred because all the errors that are connected and have the same endpoint have the same syndrome. This is referred to as error degeneracy. This is not so fatal that we cannot correct the error; we can simply apply Z operators that connect the two vertex stabilizers that are flipped and don't care about where the errors actually occurred. The error will be corrected because all Z errors will form a closed loop and become a stabilizer. It does not take our code outside the code space. The same analysis holds for X-type errors.

The error can be recovered if the error chain is less than half the linear distance of the surface. Otherwise, it may not be recovered because the standard recovery process may introduce a logical error, mapping the code to a different logical state. For an $L \times L$ lattice, since

$d = 2t + 1$, the code can correct upto any $t = (L - 1)/2$ errors. The toric code generally has a distance $d \sim \sqrt{n}$. A better error tolerance will be at the expense of more physical qubits.

Manifolds other than the torus can also be considered. Generally, for a surface code defined on a manifold of genus g and having h holes, $k = 2g + h$. There are some advantages of the surface code. (i) It has a distance $d \sim \sqrt{n}$, so it can correct $t = (d - 1)/2 = O(\sqrt{n})$ errors. (ii) It has local stabilizers (involve operators geometrically neighboring each other). An interesting feature of this kind of code is that the code space can be regarded as the ground space of a local Hamiltonian

$$H = -\sum S_X - \sum S_Z. \tag{2.2}$$

Since all stabilizers commute, the ground state of this Hamiltonian is the $+1$ co-eigenspace of the stabilizers; thus, the code space. This Hamiltonian is gapped because there is a finite energy difference between the ground state and the first excited states even in the thermodynamic limit ($n \to \infty$).

2.1.1.3 Error model and decoder

The error model is crucial for developing effective codes. Here, we consider the code capacity noise model, which means that the measurement operations are faultless. In the case of toric code, for simplicity, we only focus on the phase flip errors, as the bit flip error can be corrected in the same way as the phase flip error but with S_X stabilizer measurement. Thus, any single qubit has a probability p for a Z type error and a $1-p$ chance that the qubit remains unchanged.

$$|i\rangle \to pZ\rho Z + (1 - p)\rho \tag{2.3}$$

Intuitively, we would expect that increasing code distance could increase the robustness of toric code. In addition, increasing the code size would introduce more errors for an independent noise channel. From this perspective, we can define an error threshold for the code. If the error rate is less than the threshold, increasing the code size will improve the robustness of the code. If the error rate exceeds the error threshold, increasing the code size

will decrease the probability of successful error correction. In 2002, Dennis et al. gave an analytical lower bound of the error threshold for the toric code, which is 11% [52].

The measurements of stabilizer generators, which are called error syndrome, can be used to correct errors. However, in some cases, the syndrome and the correction operator are not one-to-one. In those cases, we either correct the error or introduce a logical error. The strategy for obtaining a correction operator for a given syndrome is called a decoder. The threshold depends on both the error channel and the decoder. For the toric code under the independent Pauli Z noise channel described above. We consider the minimal weight perfect matching (MWPM) decoder.

The underlying logic of the MWPM decoder is that the error rate of the system is considered to be sufficiently small, so the most likely error pattern that is consistent with the syndrome is the one that has a minimal number of errors. Thus, we would like to find a minimal weight for a given perfect match. To apply the MWPM decoder, the quantum code needs to have just one type of excitation for each kind of error (X or Z). This means that any pair of matching can give a valid correction operator. In the case of toric code, a simple simulation can be done to show that the threshold for the MWPM decoder is about 10.3% [124, 88] (as shown in Fig. 2.3).

2.1.2 Discrete time crystal

A discrete-time crystal (DTC) describes systems that spontaneously break the continuous-time translation symmetry, much like ordinary crystals break the continuous spatial translation symmetry (see [125] for a detailed discussion). Time crystals are typically realized in periodically driven systems, known as Floquet systems, where a Floquet operator encapsulates the driving mechanism. These systems are characterized by a periodic Hamiltonian with a period T with $H(t) = H(t + T)$, thus manifesting symmetry that is discrete and explicitly linked to the periodicity of the driving term. This periodic driving leads to dynamics that are not invariant under arbitrary time translations but only under multiples of the driving period, hence breaking the continuous-time symmetry.

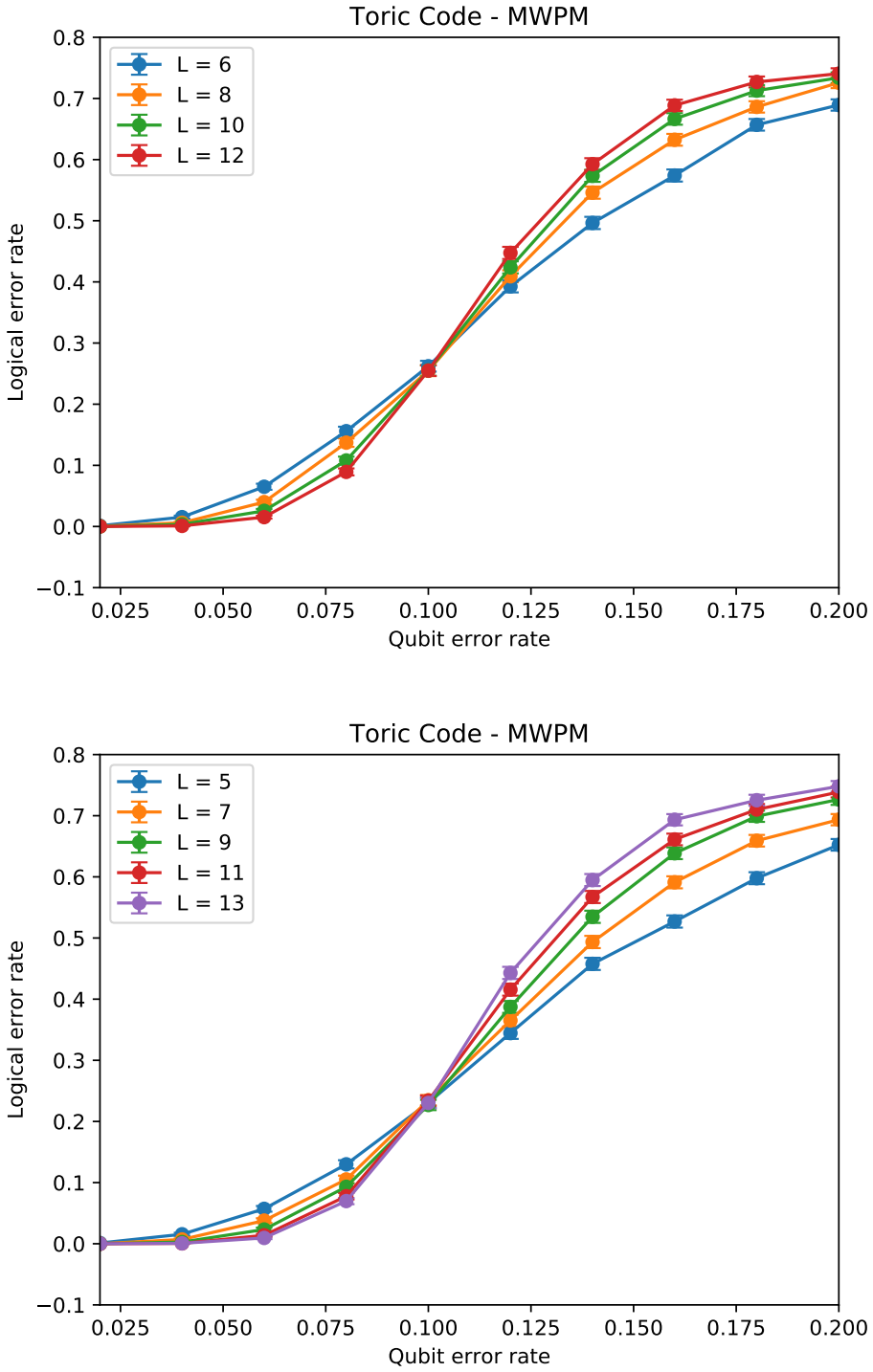


FIGURE 2.3: The performance of the MWPM decoder is depicted in two sections: the upper section illustrates its performance for even lattice sizes, while the lower section shows its performance for odd lattice sizes. The transverse axis is the physical error rate, while the vertical axis is the logical error rate after decoding. Different colors label different system sizes. The threshold is approximately 10%. When the physical error rate is less than the threshold, an increase in the system size leads to better performance (lower logical error rate), while this becomes not true when the error rate is larger than the threshold.

Raditya W. Bomantara has researched the impact of periodic driving in quantum codes, explicitly focusing on repetition codes[126] and surface codes[127]. In his study on surface codes, he suggests using a nonlocal generalization of specific metrics to characterize different phases of the surface code Hamiltonian under varying boundary conditions. His findings indicate that different boundaries exhibit DTC phases.

2.2 Periodical driving

In this section, we demonstrate how to construct a discrete-time crystal by periodically driving the toric code. The system setup is the following: the toric code is defined on a $L_x \times L_y$ lattice with periodic boundary conditions, with initial state $|\psi\rangle$. The Floquet operator U_T is expressed as

$$U_T = U_{T_1} U_{\text{error}} U_{T_2} = e^{i\sum_l a_l X_l} e^{i\sum_k c_k X_k} e^{i\sum_j b_j S_j}. \tag{2.4}$$

This operator is applied periodically with time step T. Now, let us break down the driving term and explain each part. First, the component U_{T_1} implements a logical X operator, so with a chosen logical X operator, l sums over all the single Pauli X that belong to that logical operator. The parameter a_l here is for disturbance. At $a_l = 0.5\pi$, U_{T_1} performs an ideal logical X operator. The error term U_{error} introduces X type errors across all qubits k with corresponding amplitudes c_k. Similarly, the term c_k can vary to simulate different error strengths. The last one, U_{T_2}, involves the stabilizers of the toric code, denoted by S_j. Thus, j is designed to sum over all plaquette and vertex operators, with perturbation parameters b_j.

This setup is a Floquet system with discrete time-steps, where evolution is governed by the Floquet operator U_T rather than continuous-time evolution e^{-iHt} from a static Hamiltonian. The system evolves under U_T, which comprises three components: U_{T_1} (a logical operator), U_{error} (simulating the error mechanism), and U_{T_2} (the Hamiltonian term with perturbations). In the absence of errors and the logical operator (which gives discrete time crystal behavior), this resembles evolution under a static Hamiltonian, effectively simulating a toric code Hamiltonian with errors.

The Floquet operator U_T allows the simulation of both logical operators and arbitrary error dynamics within the toric code. Given such a Floquet operator U_T, with an input as toric code state. The main output of this simulation is a time series of logical Z operator measurements. In the following, we show the whole simulation process.

- 1. Initialization: set all qubits on the $L_x \times L_y$ lattice in the state of $+1$ eigenstate of Pauli Z operator, i.e. $|0\rangle$.
- 2. Apply stabilizer operator U_{T_2}: Implement the stabilizer operations that involve all plaquette and vertex operators of the toric code.
- 3. Introduce error U_{error} : Introduce errors into the system based on a preset error rate (governed by c_k) that involves all qubits.
- 4. Measure the logical Z operator by first selecting a logical Z operator (e.g., a Z chain along the horizontal direction of the torus layout), then computing the expectation value by sandwiching the operator with the state. Note that since we perform state simulation with complete state information, we extract the logical information while preserving the pre-measurement state; the state does not collapse from the measurement. Furthermore, there are two different ways to do the measurement here. One directly gives the measurement outcome, while the other involves reading the value of the logical Z operator after the MWPM decoding. We will compare these two results.
- 5. Apply logical X operation U_{T_1}: Execute logical X operations that flip single qubits belonging to that logical operator.
- 6. Measure logical Z operator again: Same as step 4.
- 7. Repeat for multiple simulations: Cycle through steps 2 to 5 for N iterations to gather time series data from logical Z measurements.

It should be emphasized that, in the first step, we construct a $L_x \times L_y$ qubit grid, with all qubits in the $|0\rangle$ state. This is not a toric code state. During the full simulation, we only extract logical Z information. The all-$|0\rangle$ state is a stabilizer state for all Z-type stabilizers, which is sufficient to establish well-defined logical Z values (as it in the logical Z code state) and track their dynamics under the Floquet protocol.

The whole simulation is repeated multiple times, and the measurement results of the logical Z operator are averaged over these repetitions. As expected, the result of the logical Z operator falls within the range $[-1, 1]$. In the following section, we detail the various simulations we conducted and explain our choice of parameters. To enhance the realism of the simulation, we also show how to disturb those parameters.

2.3 Simulation results

2.3.1 Different stabilizer prefactor

We first show the results of varying the stabilizer prefactor b_j while keeping the other parameters fixed. The size of the system is chosen as a 3×3 lattice with 18 qubits. The logical operator prefactor $a_l = 0.4\pi$ with a perturbation uniformly selected from $(-0.1, 0.1)$ for each l. The error rate parameter is set to $c_k = 0.02\pi$, with a perturbation uniformly selected from $(-0.02, 0.02)$ for each k. Different values of b_j are chosen from the set $[0.00\pi, 0.05\pi, 0.15\pi, 0.25\pi, 0.35\pi, 0.45\pi, 0.50\pi]$. Note that we also add a perturbation uniformly sampled from $(-0.5, 0.5)$ for each j. The simulation results are shown in Fig. 2.4.

The oscillation in Fig. 2.4 shows an alternation between negative and positive values with a period of $2T$. During each $2T$ interval, we first obtain a positive value (corresponding to the logical Z information), followed by a negative value. This behavior is expected because a logical X operation is applied between consecutive measurements, causing the logical Z information to alternate sign and producing the $2T$ oscillation. After a certain time T, error accumulation degrades the logical Z information, causing the measurement signal to collapse to zero and the $2T$ oscillation to vanish.

Additionally, the oscillation is more robust when using a decoder, as it persists longer. Moreover, as shown in Fig. 2.4. In the case of the decoder, it is evident that the optimal choice is $b_j = 0.25\pi$, as this value maintains the oscillation for the longest duration. In the following, we will elucidate the reasons behind this assertion. This argument is attributed to Raditya W. Bomantara.

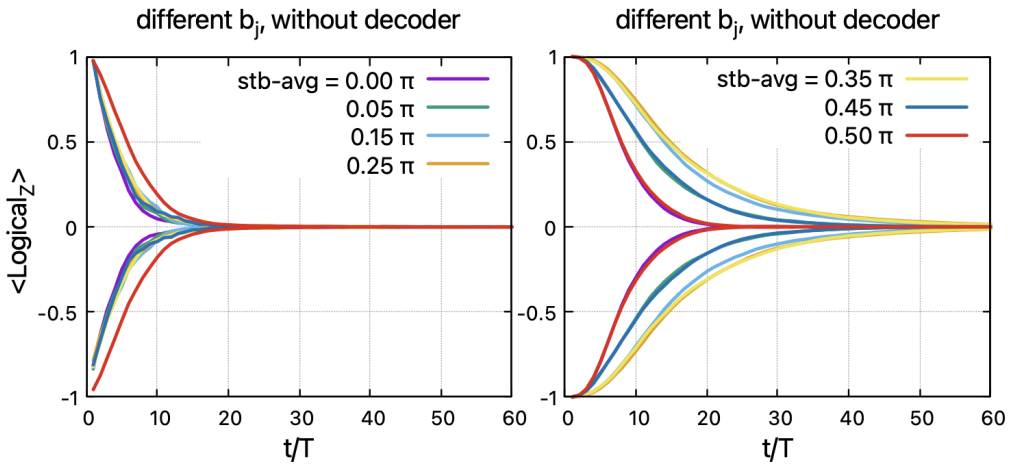


FIGURE 2.4: Simulation results with different b_j. System size is 3×3 with 18 qubits. $a_l = 0.4\pi$ with a perturbation uniformly selected from $(-0.1, 0.1)$ for each l. $c_k = 0.02\pi$, with a perturbation uniformly selected from $(-0.1, 0.1)$ for each k. Different color corresponding to different value of b_j, which is chosen from the set $[0.00\pi, 0.05\pi, 0.15\pi, 0.25\pi, 0.35\pi, 0.45\pi, 0.50\pi]$, with perturbations uniformly sampled from $(-0.5, 0.5)$ for each j. The transverse axis is the time step, while the vertical axis is the average measurement outcome of the logical Z operator. The plot on the left is the result without the decoder, while the plot on the right is the case with the decoder.

Consider a typical $2T$-crystal described by the unitary operator

$$U = U_{\log} U_{\text{stab}} = e^{-iH_{\log}} e^{-iH_{\text{stab}}} \ , \tag{2.5}$$

where

$$\begin{aligned} H_{\log} &= \sum_j h_j X_j \ , \\ H_{\text{stab}} &= \sum_j J_j \mathcal{S}_j \ , \end{aligned} \tag{2.6}$$

$\mathcal{S}_j$ are stabilizer operators. Note that U can be diagonalized with eigenvalues of the form $e^{-i\varepsilon}$. In the following, we will refer to the eigenvectors of U as the Floquet eigenstates and ε as quasienergy. Note that by construction, ε is only defined modulo 2π with ε and $\varepsilon + 2\pi k$ representing the same physics. As such, we can restrict $\varepsilon \in [-\pi, \pi]$.

Next, note that in a $2T$ time crystal, *all* eigenstates must come in pairs with π quasienergy separation, i.e., if $|\varepsilon\rangle$ is the Floquet eigenstate of U associated with quasienergy ε, then there

must exist $|\varepsilon'\rangle$ which is also a Floquet eigenstate of U but with quasienergy $\varepsilon + \pi \bmod 2\pi$. This can be understood from the fact that a generic state can be expanded as

$$|\psi\rangle \;=\; \sum_{\varepsilon} c_\varepsilon |\varepsilon\rangle \;. \tag{2.7}$$

In the special case that $|\psi\rangle = |\varepsilon\rangle + |\varepsilon+\pi\rangle$, it evolves to $U|\psi\rangle = e^{-i\varepsilon}(|\varepsilon\rangle - |\varepsilon+\pi\rangle)$ after one period, before it returns back to $U^2|\psi\rangle = e^{-2i\varepsilon}(|\varepsilon\rangle + |\varepsilon+\pi\rangle) = e^{-2i\varepsilon}|\psi\rangle$, thus establishing a $2T$-periodicity.

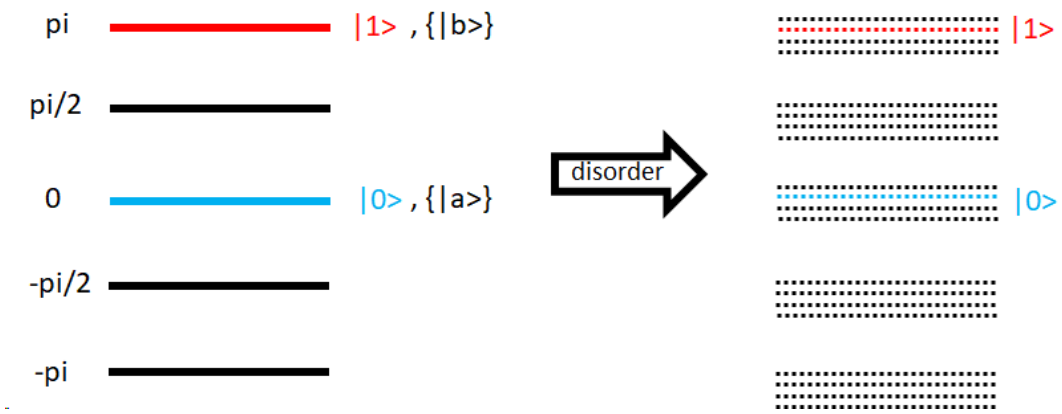
To understand how the above π spacing is established in the model, we first note that the eigenstates of H_{stab} are at least two-fold degenerate and form logical subspaces corresponding to different $\mathcal{S}_j = \pm 1$ combination. At appropriate parameter values, the unitary $U_{\log}$ realizes a logical X gate. It has the effect of splitting the degeneracy between two logical states in each logical subspace by π quasienergy. For a system to be called a time-crystal, this π quasienergy spacing, hence the associated $2T$-oscillation, must be robust against perturbation. In the following, we intuitively explain the role of disorder and quasienergy gap in ensuring this robustness.

First, let $J_j = \pi/4$ for simplicity. In the ideal case that $U_{\log}$ is a perfect logical X gate, the quasienergy spectrum can be easily obtained as Two eigenstates in a logical subspace

$|0\rangle, |1\rangle \in \mathcal{L}$ with π quasienergy separation are highlighted in red and blue for reference, along with a set of other eigenstates that are degenerate with them. Note that this large degeneracy is generally not desirable for the formation of time crystals. Indeed, let $|b_1\rangle$ and $|a_1\rangle$ be two eigenstates at quasienergy π and 0 respectively that belong to two different stabilizer

subspaces $\mathcal{L}'$ and $\mathcal{L}''$. Consider a perturbation of the form $P = p' + p''$, where p' couples $\mathcal{L}$ and $\mathcal{L}'$, while p'' couples $\mathcal{L}$ and $\mathcal{L}''$. According to the theory of degenerate perturbation, p' will resolve $|0\rangle$ into some superposition $|0'\rangle = \alpha|0\rangle + \beta|b_1\rangle$ with a quasienergy shift $\langle 0'|P|0'\rangle$, while p'' will resolve $|1\rangle$ into some superposition $|1'\rangle = A|1\rangle + B|a_1\rangle$ with quasienergy shift $\langle 0'|P|0'\rangle$. Since $\langle 0'|P|0'\rangle \neq \langle 1'|P|1'\rangle$ in general, the above π quasienergy separation no longer exists and, consequently, $2T$-oscillation is destroyed.

To avoid the above degeneracy problem, the disorder in J_j is required. Assuming that J_j is taken from the set $(\pi/4 - \delta, \pi/4 + \delta)$, the qualitative effect of this disorder on the quasienergy spectrum is illustrated below. That is, the disorder causes the “broadening" of



the clean quasienergy levels into bands centered at $k\pi/2$ quasienergy due to the $\pi/4$ mean of the disorder. In this case, quasienergy shift due to perturbation can generally be described by the second-order perturbation theory

$$\sum_{\varepsilon' \neq \varepsilon} \frac{\langle \varepsilon|P|\varepsilon'\rangle\langle \varepsilon'|P|\varepsilon\rangle}{e^{-i\varepsilon} - e^{-i\varepsilon'}} \,. \tag{2.8}$$

That is, such a shift is suppressed by the quasienergy gap $e^{-i\varepsilon} - e^{-i\varepsilon'}$. As such, the maximum robustness of the $2T$ oscillation is obtained when such a quasienergy gap is maximized. It is now easy to understand why the choice of J_j centered on $\pi/4$ is optimal; the quasienergy band at $\pi/2$ has the largest gap with respect to the two quasienergy bands at 0 and π. The following figure illustrates how other choices of J_j compare with the optimal case above. In the case of J_j centered around $\pi/2$, all quasienergy eigenstates are centered around either 0 or

$\pm\pi$ quasienergies; very large disorder strength is required to break any accidental degeneracy. This is, therefore, not a good parameter choice.

Further analysis of simulations with different values of b_j reveals that for the decoder-enhanced scenarios, $b_j = 0.25\pi$ emerges as optimal. This optimal setting can be attributed to the dynamics of quasienergy shifts resulting from perturbations, which can generally be described by second-order perturbation theory:

$$\sum_{\varepsilon' \neq \varepsilon} \frac{\langle \varepsilon | P | \varepsilon' \rangle \langle \varepsilon' | P | \varepsilon \rangle}{e^{-i\varepsilon} - e^{-i\varepsilon'}} \tag{2.9}$$

This quasienergy shift is mitigated by the quasienergy gap $e^{-i\varepsilon} - e^{-i\varepsilon'}$, which is instrumental in determining the robustness of the 2T-oscillation. Maximization of this gap is crucial to improving the stability of the system against perturbations. In particular, the quasienergy band at $\pi/2$ presents the largest gap relative to both bands at 0 and π, thereby offering maximal robustness to the $2\ \mathrm{T}$ oscillation under optimal conditions.

2.3.2 Different error rate

We also performed simulations varying the error rate, which is determined by c_k in the Floquet operator U_T. This parameter directly influences the strength and frequency of errors introduced

into the system, allowing us to assess the robustness of the toric code under different conditions of quantum noise. In this case, we fix the system size as 3×3 with 18 qubits. The prefactor of the logical operator $a_l = 0.4\pi$ with a perturbation uniformly selected from $(-0.1, 0.1)$ for each l. The term b_j is chosen as $b_j = 0.5\pi$ with a perturbation uniformly sampled from $(-0.5, 0.5)$ for each j. The error rate parameter is chosen from the set $[0.01\pi, 0.02\pi, 0.03\pi, 0.06\pi, 0.10\pi]$, with a perturbation uniformly selected from $(-0.02, 0.02)$ for each k. The result is shown in Fig. 2.5.

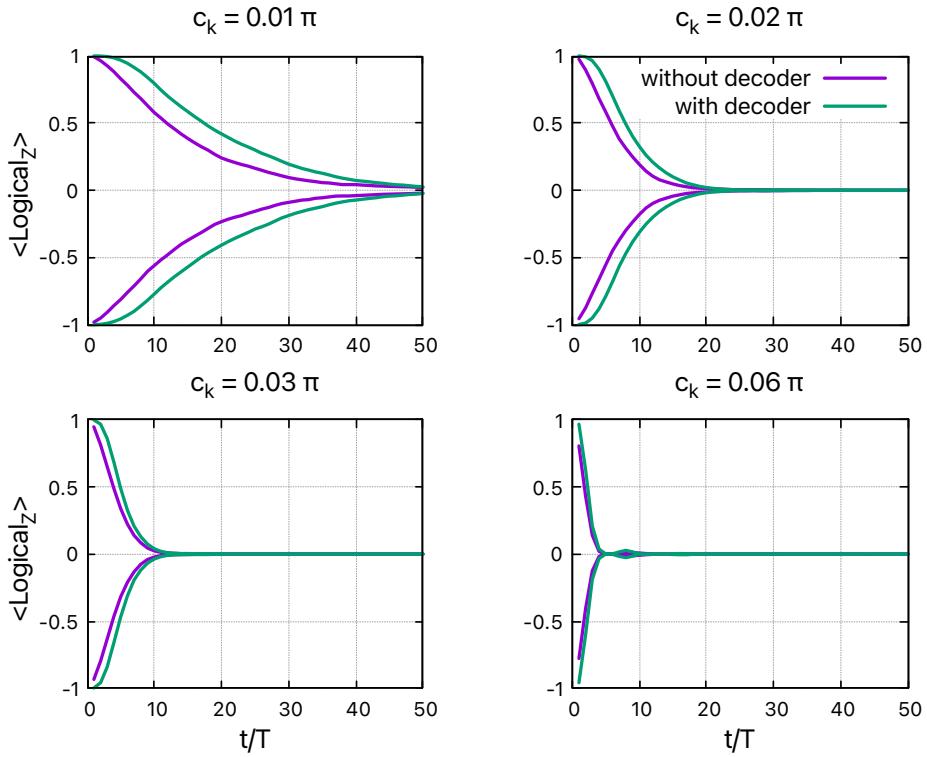


FIGURE 2.5: Simulation results across varying error rates. The logical operator coefficient $a_l = 0.4\pi$ with a perturbation uniformly selected from $(-0.1, 0.1)$. The system size is 3×3 with 18 qubits. $b_j = 0.5\pi$ with a perturbation uniformly sampled from $(-0.5, 0.5)$ for each j. Error rates c_k are selected from $[0.01\pi, 0.02\pi, 0.03\pi, 0.06\pi, 0.10\pi]$ with uniform perturbations in $(-0.1, 0.1)$. For different error rates, they compare decoder (green) and without-decoder (purple) results, with time steps on the x-axis and logical Z operator measurements on the y-axis

We observe that, at lower error rates, the oscillation persists longer. This is expected, as the collapse of oscillations is due to errors destroying the toric code. Furthermore, for each error

rate, using a decoder makes the oscillations more pronounced, as the absolute value of the averaged logical outcome of the measurement Z is higher.

2.3.3 Different system size

Finally, we consider the effect of the system size. Here, we fix $b_j = 0.5\pi$ with auniform perturbations in $(-0.5, 0.5)$, logical operator coefficient $a_l = 0.4\pi$ with perturbations in $(-0.1, 0.1)$, and error rate $c_k = 0.02\pi$ with perturbations in $(-0.02, 0.02)$. We consider three system sizes 3×2, 5×2, and 7×2, with 12, 20, and 28 qubits, respectively. Note that for these system sizes, one logical operator of the toric code has a distance of 2, which is insufficient for error correction. However, since the toric code encodes two logical qubits, the other logical operator has distances of 3, 5, and 7 in these cases. Therefore, we can use these long-distance logical operators for measurements and decoding. The reason we set one direction to have a distance of 2 is mainly because simulating larger systems is difficult, as the computational cost grows exponentially with the number of qubits. The simulation result is shown in Fig 2.6.

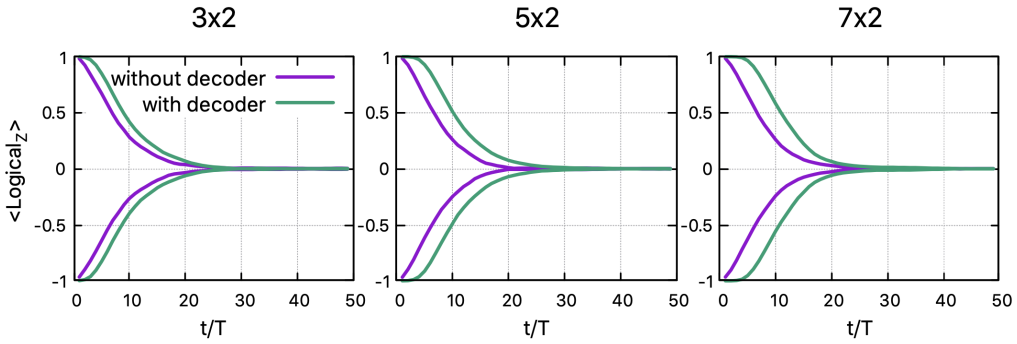


FIGURE 2.6: Simulation results with different system sizes. In U_T, $b_j = 0.5\pi$ with a perturbation uniformly sampled from $(-0.5, 0.5)$ for each j. $a_l = 0.4\pi$ with a perturbation uniformly selected from $(-0.1, 0.1)$ for each l. $c_k = 0.02\pi$, with a perturbation uniformly selected from $(-0.02, 0.02)$ for each k. We consider three system sizes 3×2, 5×2, and 7×2, with 12, 20, and 28 qubits, respectively.

In Fig. 2.6, we observe that for each size of the system, the oscillations last longer with the decoder. In addition, decoder effectiveness increases proportionally with system size, consistent with sub-threshold error rate behavior. Furthermore, we also plot the results of different system sizes with and without the decoder to directly analyze the effect of the system size, which is shown in Fig. 2.7. For the case without a decoder, the $2T$ oscillations are nearly identical between system sizes since the error rate is the same, and the average logical Z value depends primarily on this rate. However, with the decoder, the oscillations shift because of the decoder's performance. In particular, this enhancement diminishes at $T = 18$ when errors exceed the threshold, negating the advantages of larger system sizes.

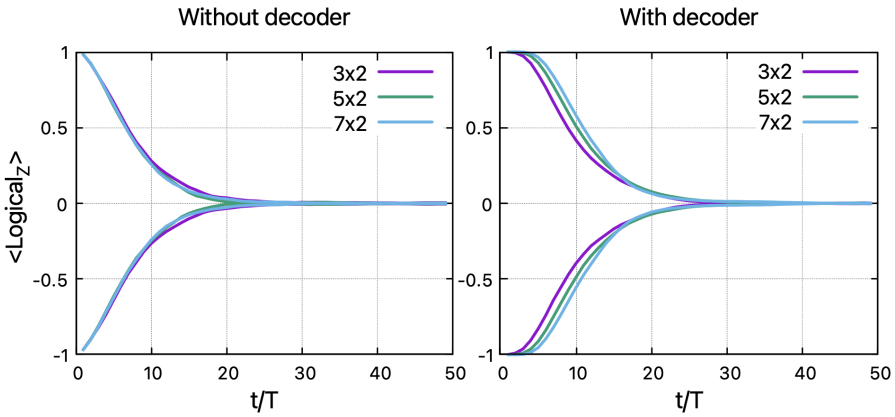


FIGURE 2.7: Simulation results for different system sizes are shown. The parameters are the same as before, but we separately plot the cases with and without the decoder to directly analyze the effect of system size.

2.4 Outlook

The simulation code can be found on Github [1]. To give more persuasive results, we consider scaling up the system size further. However, since we perform an exact simulation through the code state, a 7×2 size toric code system evolution requires approximately ten days to complete on a high-performance computing system. Therefore, the current simulation method

[1] https://github.com/Guangqi-Phys/DTC-toric/tree/master

is computationally constrained and is incapable of performing simulations on a larger system size.

Nevertheless, tensor network approaches may offer a viable alternative for larger systems. This approach is notably more efficient for simulating systems with mild entanglement. As a simplification method, the tensor network framework allows us to effectively control the entanglement between qubits. For example, in the case of 1D gapped systems, the entanglement is naturally bounded due to the area law [128], making MPS an adequate representation of the system. The toric code $n \times 2$ can be modeled using a one-dimensional matrix product state (MPS) and representing the Floquet operator as a matrix product operator (MPO). Future work will employ tensor networks to investigate size-dependent behaviors of discrete time crystals in the toric code.

CHAPTER 3

Energy barrier of hypergraph product codes

3.1 Introduction

Quantum computers are capable of solving problems that are likely intractable for classical computers, such as factoring of large integers [2] and simulation of quantum systems [1, 129]. However, realistic quantum computers are noisy. In order to build a useful quantum computer capable of carrying out such computational tasks, quantum error correction is likely necessary.

Since the discovery of the first quantum error-correcting code [18], much progress has been made towards finding better codes. While the leading approach to scalable quantum error correction has been the surface code [121] for nearly 20 years, lately there has been a surge of interest in using quantum low-density parity check (LDPC) codes. This recent interest is in part due to Gottesman, who showed that with quantum LDPC codes, one can achieve a constant overhead for fault-tolerant quantum computation [39]. A well-known approach to construct such codes is the hypergraph product construction [40]. More recent studies led to the development of other families of quantum LDPC codes with improved parameters [41, 42, 43, 44, 45, 46, 47, 48, 49]. Moreover, recent studies suggest that there are viable fault-tolerant quantum computing architectures that can host such codes [50, 5, 51].

One challenge in using these codes lies in the decoding. While there are general methods such as the BP+OSD decoder [71], it is a priori not obvious how well such a general-purpose decoder would work for a given code. One attractive approach is to use codes that have an extensive energy barrier. For such codes, a simple process that iteratively lowers

the energy (quantified in terms of the number of stabilizers violated) is a viable decoder candidate. Alternatively, such a model can be viewed as a candidate for self-correcting quantum memory, protecting quantum information by the macroscopic energy barrier. This has been demonstrated for codes with confinement using a thermal Gibbs “decoder" [130]. (We note that the energy barrier is not a sufficient condition for building a self-correcting quantum memory [131, 104, 105], though it is nevertheless a necessary condition.)

Such approaches work well for four-dimensional (4D) toric code [132] and the quantum expander code [102, 106], both of which have extensive energy barriers. Unfortunately, rigorous bounds on the energy barrier are hard to come by and often derived for specific codes (or family of codes) [133, 104, 102, 134, 135].

In this paper, we prove a tight bound on the energy barrier for the hypergraph product codes [40], defined in terms of the energy barrier of the underlying classical codes. The hypergraph product is defined in terms of the parity check matrices (or equivalently, the Tanner graphs) of two classical codes. While there are codes with better parameters [44, 46, 45, 47], the hypergraph product remains a flexible framework for constructing quantum LDPC codes with many advantages, such as the variety of decoders [102, 106, 71, 72, 136], logical gates [108, 137, 138], and distance-preserving syndrome extraction circuit [50, 112].

Now, we describe our main result. Without loss of generality, consider a hypergraph product of two classical codes, defined in terms of the parity check matrices H_1 and H_2. We denote the parity check matrix of the resulting quantum code as $H_{(H_1,H_2)}$. For both the quantum and the classical code, we denote the energy barrier as $\Delta(H)$, where H can be a parity check matrix of the quantum or the classical code. We prove that under a modest condition,

$$\Delta(H_{(H_1,H_2)}) = \min(\Delta(H_1), \Delta(H_2), \Delta(H_1^T), \Delta(H_2^T)), \tag{3.1}$$

where H^T is the transpose of H. Eq. (3.1) holds if the energy barriers of the classical codes are larger than or equal to a certain sparsity parameter of the code. Importantly, if the underlying codes are LDPC, then the sparsity parameter is $O(1)$. Therefore, Eq. (3.1) holds if the energy barriers of H_1, H_2, H_1^T, and H_2^T grow as the code size grows. If this condition is not satisfied,

the energy barrier is bounded by a constant, a fact that follows trivially from the definition of the hypergraph product code.

The proof of Eq. (3.1) is based on two results. First, if two logical operators of a quantum LDPC code are equivalent up to a stabilizer, their energy barriers differ only by a constant related to the code's sparsity parameters w_c, w_q [Theorem 3.3.1]. Due to this result, proving the energy barrier for a given code reduces to proving the energy barrier for any complete set of logical operators, which can be a much smaller set than the set of all logical operators. We then identify a set of logical operators for which the exact energy barrier can be determined in terms of the energy barriers of the underlying classical codes. Together, these two results imply Eq. (3.1).

As an application, we provide a simple proof that the quantum expander code [102, 106] has a macroscopic energy barrier. We remark that the energy bound for this code also follows from Lemma 11 of Ref.[106], though our technique can be applied more generally.

3.2 Energy barrier of codes

We first present a formal definition of the energy barrier for stabilizer codes [139]. Let $\mathcal{C}$ be the code subspace of a stabilizer code, defined in terms of the stabilizer group S. The code subspace can be viewed as the ground state subspace of a Hamiltonian of the form $\hat{H} = \sum_{i=1}^{m}(I - s_i)/2$, where $\{s_1, \ldots, s_m\} \subset S$ is a set of generators. Note that the energy barrier depends on the choice of the generating set, the energy barrier is a physical concept rather than an intrinsic mathematical property of codes. Specifically, the energy barrier depends on the interactions of a system and the terms in its Hamiltonian. For a stabilizer code, many different generating sets can define the same code space, but the choice of generating set determines the Hamiltonian and thus the energy landscape. In this sense, the energy barrier is dependent on the choice of generating set.. For an operator P in the Pauli group $\mathcal{P}$, the energy of the state $P|\psi\rangle$ is given by $\langle\psi| P^\dagger \hat{H} P |\psi\rangle = \epsilon(P)$. Here $|\psi\rangle$ is any ground state with $\langle\psi| \hat{H} |\psi\rangle = 0$, and $\epsilon(P)$ is the energy cost of P. This is the number of s_is that anticommute with P, i.e., $\epsilon(P) = |\{i : s_i P = -P s_i\}|$. Equivalently, one may define the energy barrier in

terms of the parity check matrix H of the stabilizer code (note that in the chapter, we use $\hat{H}$ to denote the Hamiltonian, and use H to denote parity check matrix). Let $v(P)$ be the binary (bit-string) representation of a Pauli P:

$$\epsilon(P) = \text{wt}(Hv(P)). \tag{3.2}$$

A sequence $P_0, P_1, \ldots, P_t$ from the Pauli group $\mathcal{P}$ forms a *path* from P_0 to P_t if for each index i, the operators P_i and P_{i+1} differ at no more than one qubit. The notation $w(P_0, P_t)$ represents the collection of all such paths from P_0 to P_t. For $r \in w(P_0, P_t)$, $\epsilon_{\max}(r)$ denotes the highest energy along path r, i.e., $\epsilon_{\max}(r) = \max_{P_i \in r} \epsilon(P_i)$, as the energy barrier of E along the path r.

The minimum energy associated with a Pauli P is the smallest value of ϵ_{max} across all possible paths from 0 to P. This is the energy barrier of P, denoted as $\Delta(P)$:

$$\Delta(P) = \min_{r \in w(I,P)} \epsilon_{\max}(r). \tag{3.3}$$

The energy barrier of the quantum code is the minimum energy barrier over the set of nontrivial logical operators.

DEFINITION 3.2.1. *Given a quantum code with stabilizer generating set S, let H be the parity check matrix corresponding to S, and let $L(S)$ be the set of nontrivial logical operators. Then the energy barrier is*

$$\Delta(H) := \min_{\ell \in L(S)} \Delta(\ell). \tag{3.4}$$

This is the smallest energy the environment has to overcome to enact a logical operation on the encoded qubit. We can similarly define the energy barrier of classical codes by only considering the path formed by Pauli-Xs. We shall denote this energy barrier also as $\Delta(H)$, where H in this case is the parity check matrix of the classical code.

In the following, as examples, we examine the energy barrier of some classical and quantum codes.

Case 1: 1D repetition code. The one-dimensional repetition code can be regarded as analogous to a one-dimensional Ising model:

$$H = -\sum_{0<j<L} Z_j Z_{j+1}. \tag{3.5}$$

The code space encompasses the states $|0\ldots0\rangle$ and $|1\ldots1\rangle$. It is readily verifiable that the energy barrier of this system remains constant, as the transition from $|0\ldots0\rangle$ to $|1\ldots1\rangle$ incurs a fixed, constant energy cost.

$$|000\ldots0\rangle \to |100\ldots0\rangle \to |110\ldots0\rangle \to \ldots \to |111\ldots1\rangle\,. \tag{3.6}$$

During this process, at each step, only two checks adjacent to the defect (which is a point at the location of $|10\rangle$) exhibit anticommutation with the error. Consequently, the energy barrier remains constant, independent of the system size.

Case 2: 2D Ising model. The 2D Ising model has the Hamiltonian

$$H = -\sum_{0<i,j<L} Z_{i,j} Z_{i+1,j} - \sum_{0<i,j<L} Z_{i,j} Z_{i,j+1}. \tag{3.7}$$

The system is defined on an $L \times L$ square lattice with periodic boundary conditions. The ground state of this Hamiltonian also has two distinct states: $|0\ldots0\rangle$ and $|1\ldots1\rangle$. In this case, it is a self-correcting classical memory, because the energy barrier is proportional to the linear size of the system. Intuitively, for a transition from state $|0\ldots0\rangle$ to $|1\ldots1\rangle$, all spins must be flipped. The defect then takes the form of a line corresponding to the linear dimensions of the system, ensuring that the energy barrier is $O(L)$. It has been shown to be a self-correcting memory at a sufficiently low temperature[140, 141].

Case 3: 2D Toric code. As discussed in Chapter 2, the 2D toric code encodes quantum information within a two-dimensional lattice with periodic boundary conditions. The logical operators of the toric code are defined as global loops possessing a topologically distinct structure from the local loops generated by the stabilizers, forming a 1D string. In addition,

any string errors cause stabilizers to anticommute only at the endpoints, thus the energy barrier, according to its definition, is the number of endpoints, which is 2 in this case, thus the energy barrier is a constant $O(1)$.

Case 4: 4D Toric code. In the 4D case, the logical operators have a two-dimensional membrane structure. To induce a logical error, for example, a logical X, it is necessary to flip all the qubits of the membrane. This implies that, throughout this process, the defect forms a line, as in the case of the 2d Ising model. Thus, similarly, the energy barrier is proportional to the linear size of the system. The 4D Toric code has been rigorously verified to exhibit exponential storage times $\tau \propto \exp(L)$, where L is the linear size. When operating below the critical temperature, it is a self-correcting quantum momery[132].

3.3 Quantum LDPC codes and their energy barriers

A quantum LDPC code is a stabilizer code with a sparse parity-check matrix; see [37, 38] for recent reviews. The sparsity parameters are w_c and w_q, which are the maximum row and column weights of the parity check matrix, respectively. These represent the maximum weight amongst all the checks and the maximum number of checks associated with a single bit. A code is LDPC if $w_c, w_q = O(1)$.

Here, we prove a property that holds true for any quantum LDPC code. It states that the energy barrier of two logical operators equivalent under stabilizers are equal, provided that at least one of their energy barriers is larger than or equal to $w_c w_q$. For quantum LDPC codes, $w_c w_q = O(1)$. Therefore, if the energy barrier is $\Omega(1)$ for any given non-trivial logical operator, it must also be the same energy barrier for any equivalent logical operator.

We first prove the following bound.

LEMMA 3.3.1. *Let $\mathcal{C}$ be a quantum code with sparsity parameters (w_c, w_q). For any stabilizer $s \in S$,*

$$\Delta(s) \leq w_c w_q. \tag{3.8}$$

PROOF. Without loss of generality, any stabilizer s can be expressed as a product $s = s_1 \cdots s_m$, where $s_1, \ldots, s_m$ are the stabilizer generators. Consider a path that applies s_i in sequence, from $i = 1$ to m. For each s_i, we envision applying a (sub)sequence of Paulis in the support of s_i. This subsequence has a length of at most w_c, and each single Pauli error in the sequence anti-commutes with at most ω_q stabilizers in the Hamiltonian of LDPC codes. Therefore, the highest energy attained within this subsequence is at most $w_c w_q$. Once s_i is applied, the energy cost becomes zero. Therefore, $\Delta(s) \leq w_c w_q$. □

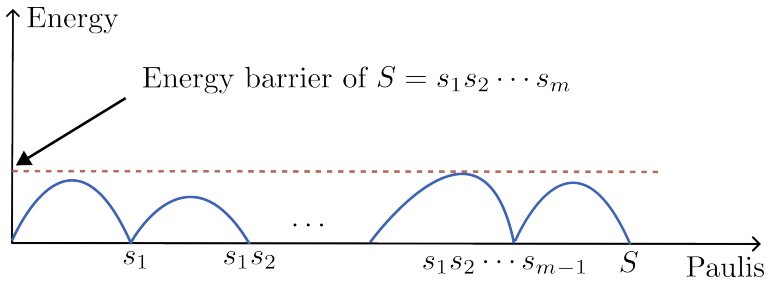


FIGURE 3.1: Graphical illustration of the energy barrier for any stabilizer S of quantum LDPC codes. Any stabilizer can be decomposed as $S = s_1 s_2 \cdots s_m$, where $s_1, s_2, \ldots, s_m$ are constant-weight stabilizers. This naturally defines a construction path for S by sequentially constructing $s_1, s_2, \ldots, s_m$. Consequently, the energy barrier of S is bounded by the energy barriers of these constant-weight stabilizers, which are constant.

THEOREM 3.3.1. *Let $\mathcal{C}$ be a (w_c, w_q) quantum LDPC code with stabilizer group S. For any logical X or Z operator L of $\mathcal{C}$ and any stabilizer $s \in S$,*

$$\Delta(Ls) \leq \max(\Delta(L), w_c w_q). \tag{3.9}$$

PROOF. Without loss of generality, consider a path $r = (\ell_1, \ldots, \ell_N) \in w(0, L)$. We can consider a new path $r' \in w(0, Ls)$ by appending r with a sequence $\delta_r \in w(0, s)$, forming $(\ell_1, \ldots, \ell_N, \ell'_1, \ldots, \ell'_M)$ where $\delta_r = (\ell'_1, \ldots, \ell'_M)$. By assumption,

$$\epsilon_{\max}(r') = \max(\epsilon_{\max}(r), \Delta(s)). \tag{3.10}$$

According to Lemma 3.3.1, $\Delta(s) \leq w_c w_q$. It implies that for any given path r, there exists a path r' such that $\epsilon_{\max}(r') \leq \max(\epsilon_{\max}(r), w_c w_q)$. Choose r such that $\Delta(L) = \epsilon_{\max}(r)$. Since $\Delta(Ls) \leq \epsilon_{\max}(r')$, we have $\Delta(Ls) \leq \max(\Delta(L), w_c w_q)$. □

We remark that using the same logic, one can deduce $\Delta(L) = \Delta(Lss) \leq \max(\Delta(Ls), w_c w_q)$. Consequently, if $\Delta(L) \geq w_c w_q$ or $\Delta(Ls) \geq w_c w_q$, $\Delta(L) = \Delta(Ls)$. Therefore, to determine the asymptotic scaling of the energy barrier of a quantum LDPC code, it suffices to consider the energy barrier of any *fixed* complete set of logical operators. Once an energy barrier is obtained for such a set, the energy of all the other logical operators is also essentially determined, thanks to Theorem 3.3.1.

However, it is a priori not obvious how to choose such a set. Below, we will solve this problem for a large family of quantum codes known as the hypergraph product code [40].

3.4 Hypergraph product code and its logical operators

Hypergraph product codes are CSS codes formed from two classical linear codes. Without loss of generality, let H_1 and H_2 be $r_1 \times n_1$ and $r_2 \times n_2$ parity check matrices, respectively. The parity-check matrix of the hypergraph product code becomes [40]:

$$\begin{aligned} H_X = &\quad \left(H_1 \otimes \mathbf{I}_{n_2}\, \mathbf{I}_{r_1} \otimes H_2^T\right), \\ H_Z = &\quad \left(\mathbf{I}_{n_1} \otimes H_2\, H_1^T \otimes \mathbf{I}_{r_2}\right). \end{aligned} \tag{3.11}$$

Because $H_X H_Z^T = 0$, these two parity check matrices define a CSS code, with a quantum parity check matrix

$$H_{(H_1,H_2)} = \begin{pmatrix} H_X & 0 \\ 0 & H_Z \end{pmatrix}. \tag{3.12}$$

The classical codes defined by $H_1, H_2, H_1^T,$ and H_2^T form the parent classical codes. Without loss of generality, we will assume that H_i and H_i^T define codes with parameters $[n_i, k_i, d_i]$ and $[r_i, k_i^T, d_i^T]$, for $i = 1, 2$. Under this assumption, the quantum code has parameter [40]

$$\left[[n_1 n_2 + r_1 r_2, k_1 k_2 + k_1^T k_2^T, \min(d_1, d_2, d_1^T, d_2^T)]\right]. \tag{3.13}$$

Hypergraph product codes can also be understood through the tensor product of chain complexes. The properties of the corresponding homology groups are used to derive code

parameters. See the Appendix for an overview of homology and the homological structure of hypergraph product codes.

3.4.1 The graph structure of hypergraph product codes

Given two classical linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$. Let $\mathcal{G}_1(V_1, C_1, E_1)$ and $\mathcal{G}_2(V_2, C_2, E_2)$ be the Tanner graphs of $\mathcal{C}_1$ and $\mathcal{C}_2$. The hypergraph product is defined by the product graph $\mathcal{G}_1 \times \mathcal{G}_2$, which is a bipartite graph with vertex set $V \cup C$, where

$$V = V_1 \times V_2 \cup C_1 \times C_2, \tag{3.14}$$

$$C = C_1 \times V_2 \cup V_1 \times C_2. \tag{3.15}$$

Then one can define two new classical codes, one is $\mathcal{G}_X = \mathcal{G}_X(\mathcal{G}_1 \times \mathcal{G}_2)$ as the subgroup of $\mathcal{G}_1 \times \mathcal{G}_2$ with bit set V and check set $C_1 \times V_2$, another is $\mathcal{G}_Z = \mathcal{G}_Z(\mathcal{G}_1 \times \mathcal{G}_2)$ as the subgroup of $\mathcal{G}_1 \times \mathcal{G}_2$ with bit set V and check set $V_1 \times C_2$, see Fig. 3.2. It can be verified [40] that

$$\mathcal{G}_X(\mathcal{G}_1 \times \mathcal{G}_2)^\perp \subset \mathcal{G}_Z(\mathcal{G}_1 \times \mathcal{G}_2). \tag{3.16}$$

Such pair $(\mathcal{G}_X, \mathcal{G}_Z)$ defines a CSS code, which is the desired hypergraph product code. Furthermore, suppose $\mathcal{G}_1$ have a parity check matrix $\mathbf{H}_1$ of size $r_1 \times n_1$, while $\mathcal{G}_2$ have a parity check matrix $\mathbf{H}_2$ of size $r_2 \times n_2$.

Just as in the classical case, each codeword is a binary string, with each bit regarded as a unit vector; the same applies to each check. In constructing hypergraph product codes, we have two classical codes. We can arrange one code horizontally (say $\mathcal{C}_1$) and another vertically ($\mathcal{C}_2$). Thus, the bits and checks of $\mathcal{C}_1$ can be regarded as horizontal unit vectors, while those of $\mathcal{C}_2$ are vertical unit vectors. Since the final qubit set comprises two parts, $V_1 \times V_2$ and $C_1 \times C_2$, we have two tensor product components (formed by the tensor product of the two sets, which is constructed from the tensor products of individual elements from each set), each with its own horizontal and vertical structure. Moreover, the logical operators of classical codes can also be regarded as vectors. In the next section, we will see that, for example, the

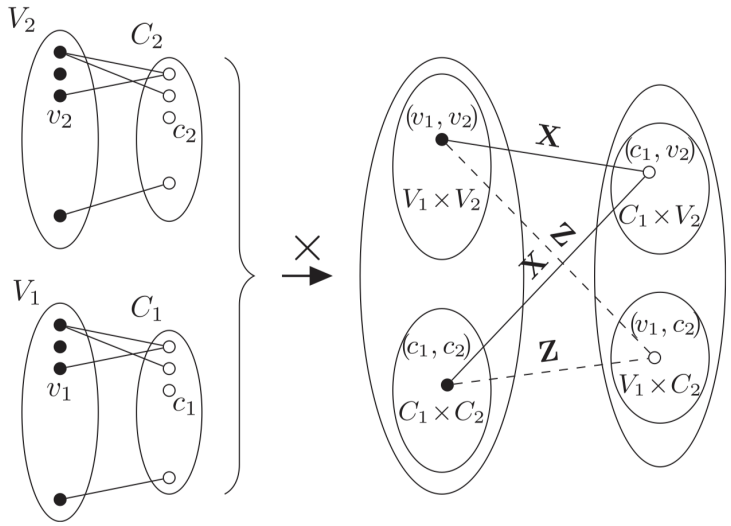


FIGURE 3.2: Graphical illustration of the Tanner graph of the hypergraph product codes. The Tanner graphs of $\mathcal{G}_X$ $(\mathcal{G}_1 \times \mathcal{G}_2)$ and $\mathcal{G}_Z$ $(\mathcal{G}_1 \times \mathcal{G}_2)$ are given by the edges of $\mathcal{G}_1 \times \mathcal{G}_2$ which join V to $C_1 \times V_2$ for $\mathcal{G}_X$ (the corresponding edges are denoted by solid lines) and which join V to $V_1 \times C_2$ for $\mathcal{G}_Z$ (the corresponding edges are indicated by dashed lines) [40].

tensor product (normal tensor product of two vector) of a logical operator of $\mathcal{C}_1$ and a unit bit vector of $\mathcal{C}_2$ yields a logical operator of the hypergraph product code. This logical operator is horizontal on the $V_1 \times V_2$ part; similarly, we have vertical logical operators.

We use the toric code as an example. The toric code is a hypergraph product of two 1D repetition codes. As shown in Fig. 3.4(a), both the horizontal and vertical codes are 1D repetition codes. The $V_1 \times V_2$ part is represented by green dots, while the $C_1 \times C_2$ part is represented by orange dots. Fig. 3.4(b) and (c) show the logical operators of the toric code: the logical Z operator is a vertical vector on $V_1 \times V_2$ but horizontal on $C_1 \times C_2$, while the logical X operator is a horizontal vector on $V_1 \times V_2$ but vertical on $C_1 \times C_2$.

3.4.2 Canonical logical operators

We now introduce a particularly useful set of logical operators, which we refer to as the *canonical* logical operators [142, 112]. For the Z-type logical operators, consider the following

operator

$$\begin{pmatrix} \sum_{k,j} \lambda_{kj} \overline{x}_k \otimes y_j \\ \sum_{\ell,m} \kappa_{\ell m} a_\ell \otimes \overline{b}_m \end{pmatrix}, \tag{3.17}$$

where (i) $H_1 \overline{x}_i = H_2^T \overline{b}_m = 0$ and (ii) $y_j \notin \mathrm{Im}(H_2^T)$ and $a_\ell \notin \mathrm{Im}(H_1)$ are unit vectors. We note that $j \in \{1, \ldots, k_2\}$ and $\ell \in \{1, \ldots, k_1^T\}$ and that the set of logical operators expressible in this form is complete [142, 112], which means all $k_1 k_2 + k_1^T k_2^T$ logical operators are provided. A canonical logical operator is *elementary* if only one of the coefficients, either λ_{kj} or $\kappa_{\ell m}$, equals one. A similar form of canonical X-type logical operators can also be constructed. Because our discussion below can be applied to such operators with little change, we omit the discussion about the X-type logical operators.

Let $\mathcal{G}_1(V_1, C_1)$ and $\mathcal{G}_2(V_2, C_2)$ be the Tanner graphs of codes defined by H_1 and H_2, respectively. Here, V_1 and V_2 represent the set of bits, and C_1 and C_2 denote the set of checks. We use $\{v_1^i : i \in \{1, 2, \cdots n_1\}\}$ (resp. $\{v_2^j : j \in \{1, 2, \cdots n_2\}\}$) to refer to bit vertices in V_1 (resp. V_2), and also as length n_1 (resp. n_2) unit vectors with the ith (resp. jth) entry as 1. The hypergraph product $\mathcal{G}_1 \times \mathcal{G}_2$ is a bipartite graph with vertex set $V \cup C$, where $V = V_1 \otimes V_2 \cup C_1 \otimes C_2$ is the qubit set and $C = C_1 \otimes V_2 \cup V_1 \otimes C_2$ is the stabilizer set.

The set of qubits can be partitioned into two subsets, $V_1 \otimes V_2$ and $C_1 \times C_2$. For H_X, $H_1 \otimes \mathbf{I}_{n_2}$ acts on $V_1 \otimes V_2$ and $\mathbf{I}_{r_1} \otimes H_2^T$ acts on $C_1 \otimes C_2$. Moreover, the subset $V_1 \otimes V_2$ can be further partitioned into n_2 subsets $\{V_1 \otimes v_2^1, V_1 \otimes v_2^2, \cdots, V_1 \otimes v_2^{n_2}\}$, where $V_1 \otimes v_2^k := \{v \otimes v_2^k : v \in V_1\}$ and $V_2 = \{v_2^1, \ldots, v_2^{n_2}\}$ [Fig. 3.3].

Thus a Z-type Pauli operator can be expressed as a bit-string $z = \left(z^{(1)}, z^{(2)}\right)^T$, where $z^{(1)}$ is supported on the qubit subset $V_1 \otimes V_2$ with vector space $\mathbb{F}_2^{n_1} \otimes \mathbb{F}_2^{n_2}$, and $z^{(2)}$ is supported on the qubit set $C_1 \otimes C_2$ with vector space $\mathbb{F}_2^{r_1} \otimes \mathbb{F}_2^{r_2}$. Because $z^{(1)}$ and $z^{(2)}$ act on a tensor product of two vector spaces, one can view them also as a matrix. (For instance, $v_i \otimes u_j$ for unit vectors v_i and u_j can be viewed as a matrix whose entry is 1 on the i'th row and the j'th column and zero elsewhere.) We call this procedure as vector reshaping, and explain a few basic facts in the Appendix. After the reshaping, $z^{(1)}$ and $z^{(2)}$ become $Z^{(1)}$ and $Z^{(2)}$ respectively. Here, $Z^{(1)}$ is an $n_1 \times n_2$ matrix, and the entry $Z_{i,j}^{(1)}$ is supported on qubit $v_1^i \otimes v_2^j$. Moreover, the jth column $Z_j^{(1)}$ is supported on qubits $\{v_1^1 \otimes v_2^j, v_1^2 \otimes v_2^j, \cdots, v_1^{n_1} \otimes v_2^j\}$, which

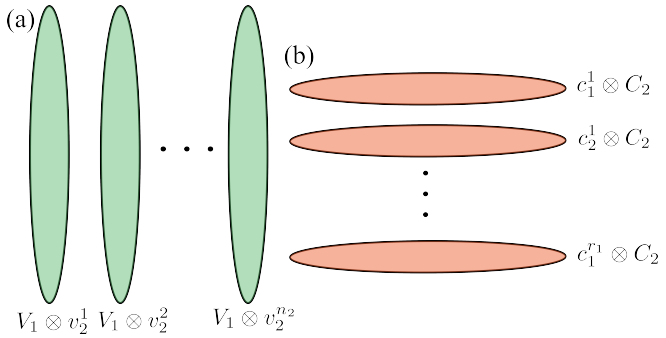


FIGURE 3.3: Graphical illustration of the hypergraph product structure: (a) The qubit subset $V_1 \otimes V_2$ is partitioned into $\{V_1 \otimes v_2^1, V_1 \otimes v_2^2, \ldots, V_1 \otimes v_2^{n_2}\}$, where $\{v_2^1, v_2^2, \ldots, v_2^{n_2}\} \in V_2$. (b) The qubit subset $C_1 \otimes C_2$ is divided into $\{c_1^1 \otimes C_2, c_1^2 \otimes C_2, \ldots, c_1^{r_1} \otimes C_2\}$, with $\{c_1^1, c_1^2, \cdots, c_1^{r_1}\} \in C_1$. An elementary canonical logical operator $(\bar{x}_k \otimes y_j, 0_{r_1 r_2})^T$ resides on the qubit subset $V_1 \otimes v_2^j$ if $y_j = v_2^j$. Similarly, the logical operator $(0_{n_1 n_2}, a_\ell \otimes \bar{b}_m)^T$ is localized on the subset $c_1^i \otimes C_2$ if $c_1^i = a_\ell$.

we refer to as $V_1 \otimes v_2^j$. Similarly, the subset $C_1 \otimes C_2$ can be partitioned into r_1 rows, and ith row is supported on qubit subset $c_1^i \otimes C_2$ (defined similarly).

An elementary canonical logical-Z operator, which is in the form $(\bar{x}_k \otimes y_j, 0_{r_1 r_2})^T$ (where $0_{r_1 r_2}$ is the $r_1 r_2$-dimensional zero vector), is supported on the subset $V_1 \otimes v_2^j$ if $y_j = v_2^j$. In the matrix form, it is supported on the jth column of $Z^{(1)}$.

It would be instructive to discuss an example of the canonical logical operators. It is well-known that Kitaev's toric code [121] can be viewed as two copies of 1D repetition code [40]. In this context, the canonical logical operators are the string operators of minimal lengths [Fig. 3.4].

3.5 Energy barrier of hypergraph product code.

We now study relationship between the energy barrier of hypergraph product codes and their underlying classical codes. We first upper bound the energy barrier of the hypergraph product code in terms of the energy barrier of the classical codes. We then prove a matching lower bound, establishing their equivalence.

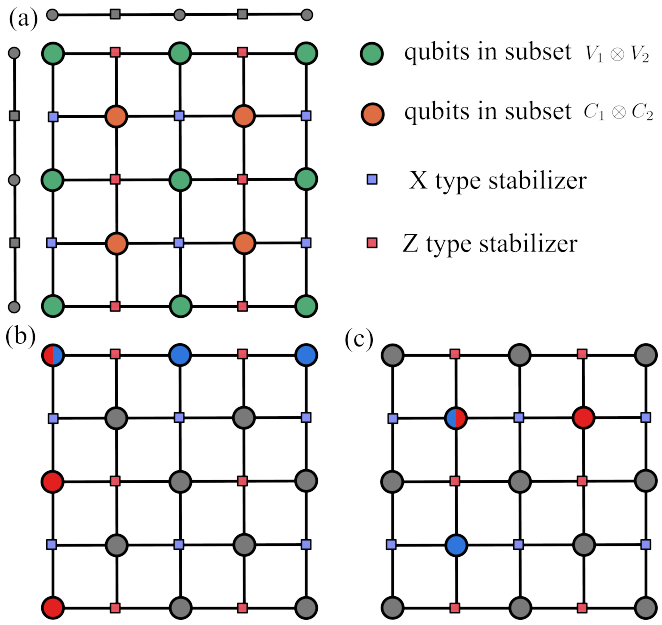


FIGURE 3.4: The hypergraph product construction of the Toric code from two repetition codes (periodic boundary conditions). (a) Delineates the two subsets of qubits, $V_1 \otimes V_2$ and $C_1 \otimes C_2$. (b) Show the logical X operator (blue circles) and the logical-Z operator (red circles) within the subset $V_1 \otimes V_2$. (c) Showcasing the logical X (blue circles) and Z (red circles) operators within the subset $C_1 \otimes C_2$.

3.5.1 Upper bound

The upper bound on the energy barrier can be obtained by considering a specific path from the identity to some logical operator. By choosing this logical operator to be a canonical logical operator, we obtain an upper bound. First, consider an elementary canonical logical operator of the form of $(\bar{x}_k \otimes y_j, 0_{r_1 r_2})^T$, where $0_{r_1 r_2}$ is the $r_1 r_2$-dimensional zero vector. We will consider a path consisting of the operators of the form of $(x_i \otimes y_j)^T$ from $i = 1$ to $i = N$, interpolating between $x_1 = (0, \ldots, 0)^T$ to $x_N = \overline{x}_k$. The energy at the ith step is precisely $\mathrm{wt}(H_1 x_i)$, which is the energy of the classical code H_1 evaluated with respect to x_i. Because $x_N = \overline{x}_k$ is a codeword of H_1, the energy barrier along this path is at most $\Delta(H_1)$.

Similarly, we can consider a path of the form of $(0_{n_1 n_2}, a_\ell \otimes b_i)$ from $i = 1$ to $i = N$, interpolating between $b_1 = (0, \ldots, 0)^T$ to $b_N = \overline{b}_m$, where $\overline{b}_m$ is a codeword of H_2^T. This yields an energy barrier upper bound of $\Delta(H_2^T)$. Together, we obtain an energy upper bound

of $\min(\Delta(H_1), \Delta(H_2^T))$ for the canonical logical-Z operators. A similar argument can be applied to the canonical logical X operators, yielding an upper bound of $\min(\Delta(H_2), \Delta(H_1^T))$.

3.5.2 Lower bound

We now prove a matching lower bound for the energy barrier of the canonical logical-Z operators.

PROPOSITION 3.5.1. *For any nontrivial canonical logical-Z operator L,*

$$\Delta(L) \geq \min(\Delta(H_1), \Delta(H_2^T)). \tag{3.18}$$

We introduce two lemmas (Lemma 3.5.1 and Lemma 3.5.3) to prove this proposition. We use the following convention in the proof. A path $r = \{P_0, P_1, \ldots, P_F\}$ is said to be supported on $U \subseteq V$ if the support of all P_i in r is included in U.

Consider an elementary canonical logical-Z operator L supported on $V_1 \otimes v_2^\alpha$. Suppose $\Delta(L)$ is given by a path r. The main idea behind the proof of Lemma 3.5.1 is to deform a general path r into a new path r', supported on $V_1 \otimes v_2^\alpha$. Importantly, we show that the energy barrier of r' is no greater than that of the original path r. A similar argument can be applied to prove Lemma 3.5.3.

LEMMA 3.5.1. *For any elementary canonical logical-Z operator L supported on $V_1 \otimes v_2^\alpha$ (resp. $c_1^\beta \otimes C_2$), there exists an optimal error path $\mathcal{P}$ such that: All errors along $\mathcal{P}$ are supported on $V_1 \otimes v_2^\alpha$ (resp. $c_1^\beta \otimes C_2$), and the energy barrier of L is achieved by this path $\mathcal{P}$.*

Now we prove this lemma. Recall that a Z-type Pauli error of the hypergraph product code can be expressed as a bit-string $z = \left(z^{(1)}, z^{(2)}\right)^T$. The corresponding energy $\epsilon(z) = \text{wt}(H_X z)$ is

$$\epsilon(z) = \text{wt}\left((H_1 \otimes I_{n_2}) z^{(1)} + (I_{r_1} \otimes H_2^T) z^{(2)}\right). \tag{3.19}$$

By applying vector reshaping, the energy can be written as

$$\epsilon(z) = \text{wt}\left(H_1 Z^{(1)} + Z^{(2)} H_2\right), \tag{3.20}$$

where $Z^{(1)}$ and $Z^{(2)}$ are the matrices reshaped from $z^{(1)}$ and $z^{(2)}$, respectively. The jth column of $Z^{(1)}$ is supported on qubit subset $V_1 \otimes v_2^j$.

Using Eq. (3.20), we aim to prove a lower bound on the energy $\epsilon(z)$ [Lemma 3.5.2]. To that end, we shall use the following convention. Let L_c be a nontrivial codeword of H_2. The set of columns of $Z^{(1)}$ associated with the nonzero entries of L_c will play an important role. We define the index set of such columns as $\mathcal{C}(L_c)$:

$$\mathcal{C}(L_c) := \{j : (L_c)_j = 1\}. \tag{3.21}$$

Given a codeword L_c of H_2, one can construct a vector $z^{1,s}$ from $Z^{(1)}$ by summing all the columns in the set $\mathcal{C}(L_c)$. More formally,

$$z_i^{1,s} = \sum_{j:j\in\mathcal{C}(L_c)} Z_{ij}^{(1)}, \tag{3.22}$$

where the addition is modulo 2. For example, if $L_c = 110100$, we would sum columns $1, 2,$ and 4. Using Eq. (3.22), we can deform an arbitrary path to a path consisting of Paulis only supported on subset $V_1 \otimes v_2^k$ for some k.[1] In particular, we can prove an inequality between the energy of the original Pauli and the deformed Pauli, proved in Lemma 3.5.2.

LEMMA 3.5.2.

$$\mathrm{wt}(H_1 z^{1,s}) \leqslant \mathrm{wt}\left(H_1 Z^{(1)} + Z^{(2)} H_2\right). \tag{3.23}$$

PROOF. We prove this by contradiction. Consider the row spaces of $H_1 z^{1,s}$ and $H_1 Z^{(1)} + Z^{(2)} H_2$. If $\mathrm{wt}(H_1 z^{1,s}) > \mathrm{wt}\left(H_1 Z^{(1)} + Z^{(2)} H_2\right)$, then there exists a row j such that

$$\mathrm{wt}((H_1 z^{1,s})_{\text{row } j}) > \mathrm{wt}\left((H_1 Z^{(1)} + Z^{(2)} H_2)_{\text{row } j}\right). \tag{3.24}$$

We will prove that Eq. (3.24) cannot be satisfied, thereby proving the claim.

Without loss of generality, consider the j'th row. Since $H_1 z^{1,s}$ is a $r_1 \times 1$ vector, $\mathrm{wt}((H_1 z^{1,s})_{\text{row } j})$ must be either 0 or 1. If $\mathrm{wt}((H_1 z^{1,s})_{\text{row } j}) = 0$, Eq. (3.24) cannot be satisfied. Therefore, we consider the $\mathrm{wt}((H_1 z^{1,s})_{\text{row } j}) = 1$ case.

[1] The precise choice of k does not matter; any $k \in \mathcal{C}(L_c)$ would suffice.

If $\mathrm{wt}((H_1 z^{1,s})_{\text{row } j}) = 1$, $(H_1 Z^{(1)})_{\text{row } j}$ must contain an odd number of ones on the columns in the set $\mathcal{C}(L_c)$. Otherwise, we would have had $\mathrm{wt}((H_1 z^{1,s})_{\text{row } j}) = 0$, which is a contradiction. On the other hand, we remark that $(Z^{(2)} H_2)_{\text{row } j}$ consists of an even number of ones on the column set $\mathcal{C}(L_c)$. To see why, note that each row of $Z^{(2)} H_2$ is a linear combination of the checks in H_2. Because L_c is a codeword of H_2, $(Z^{(2)} H_2)_{\text{row},j} L_c = 0$. Therefore, in the $\mathrm{wt}((H_1 z^{1,s})_{\text{row } j}) = 1$ case, the number of ones in $H_1 Z^{(1)} + Z^{(2)} H_2$ on the j'th row and the columns in $\mathcal{C}(L_c)$ is odd.

Thus we conclude $\mathrm{wt}\left((H_1 Z^{(1)} + Z^{(2)} H_2)_{\text{row } j}\right) \geq 1$. As such, Eq. (3.24) cannot be satisfied. This completes the proof. □

Now we are in a position to prove Lemma 3.5.1. We do so by identifying a path $r' = \{P'_0, P'_1, \cdots, P'_F\}$ that is only supported on $V_1 \otimes v_2^\alpha$ while ensuring that $\epsilon_{\max}(r') \leqslant \epsilon_{\max}(r)$. Lemma 3.5.2 suggests a way to deform the path r to the one supported on $V_1 \otimes v_2^\alpha$.

Without loss of generality, let $r = \{P_0, P_1, \cdots, P_F\}$ be a path that $\Delta(L) = \epsilon_{\max}(r)$, with $P_0 = I$ and $P_F = L$. We consider a Pauli P_i in the path r. It will be convenient to work in its binary representation, written as $(p^{(1)}, p^{(2)})^T$, where $p^{(1)}$ and $p^{(2)}$ represent the Paulis supported on $V_1 \otimes V_2$ and $C_1 \otimes C_2$, respectively. First, we remove all the Paulis supported on $C_1 \otimes C_2$ by setting $p^{(2)}$ as the zero vector. Next, we apply the following transformations to $p^{(1)}$. We reshape $p^{(1)}$ and refer to the reshaped matrix as $P^{(1)}$. Let L_c be a codeword of H_2 such that $(L_c)_\alpha = 1$. Consider a set of columns in $P^{(1)}$ corresponding to the index set $\mathcal{C}(L_c)$. Denoting each column as u_k, where $k \in \mathcal{C}(L_c)$, we update the column u_α in the following way:

$$u_\alpha \to u_\alpha + \sum_{k \in \mathcal{C}(L_c) \setminus \{\alpha\}} u_k. \tag{3.25}$$

Afterward, the other columns of $P^{(1)}$ are set to the zero vector. This yields the deformed Pauli operator P'_i.

By construction, the resulting P'_i is supported on $V_1 \otimes v_2^\alpha$. Note that r' is a valid path because $\mathrm{wt}(P'_i P'_{i+1}) \leq 1$ for every i. Also, because $P'_F = P_F = L$, r' is still a path for L. Moreover, because $\epsilon(P'_i) \leqslant \epsilon(P_i)$ for all i [Lemma 3.5.2], we have $\epsilon_{\max}(r') \leqslant \epsilon_{\max}(r)$. Both r' and r

are paths for L, by definition $\epsilon_{\max}(r') \geqslant \epsilon_{\max}(r)$, we conclude $\epsilon_{\max}(r') = \epsilon_{\max}(r)$. Thus, by deforming r, we obtained a new path r' supported on $V_1 \otimes v_2^\alpha$ that yields the energy barrier $\Delta(L)$.

This argument can be applied to prove similar lower bounds for logical operators on $c_1^\beta \otimes C_2$. To conclude, for any elementary canonical logical operator L supported on $V_1 \otimes v_2^\alpha$ (resp. $c_1^\beta \otimes C_2$), their energy barrier can be given by a path supported on $V_1 \otimes v_2^\alpha$ (resp. $c_1^\beta \otimes C_2$). Thus, we finish the proof of Lemma 3.5.1.

Another lemma we need to prove to establish the lower bound of the energy barrier is

LEMMA 3.5.3. *For any nontrivial canonical logical-Z operator L, the energy barrier $\Delta(L)$ is greater than or equal to the minimum energy barrier of the elementary canonical logical-Z operators.*

PROOF. Any nontrivial canonical logical-Z operator L belongs to one of the following categories:

- Case 1: L is supported solely on the qubit subset $V_1 \otimes V_2$.
- Case 2: L is supported solely on the qubit subset $C_1 \otimes C_2$.
- Case 3: L is supported on both subsets.

We will focus solely on Case 1. Case 2 can be analyzed similarly by considering subsets $C_1 \otimes C_2$, while case 3 can be treated as Case 1 or 2.

Without loss of generality, let the energy barrier of L be attained by a path $r = \{P_0, P_1, \cdots, P_F\}$, with $P_0 = I$ and $P_F = L$. Similar to the approach taken in Lemma 3.5.1, we aim to deform the path r to the one supported on $V_1 \otimes v_2^k$ for some k, such that the energy barrier of the deformed path lower bounds that of the r.

The deformation works in the same way as in the proof of Lemma 3.5.1. We describe this procedure again for the readers' convenience. Let L_c be a nontrivial codeword of H_2 and $\mathcal{C}(L_c)$ be its corresponding column index set. We consider a binary representation of a Pauli P_i, written as $(p^{(1)}, p^{(2)})^T$. As in the proof of Lemma 3.5.1, we remove the Paulis supported

on $C_1 \otimes C_2$ by setting $p^{(2)}$ as the zero vector. Next, reshape $p^{(1)}$ into a matrix $P^{(1)}$ and update its columns in the following way. Choose $\alpha \in \mathcal{C}(L_c)$. This column is updated as Eq. (3.25). The other columns of $P^{(1)}$ are converted to zero vectors.

Thanks to Lemma 3.5.2, we obtain a new path $r' = \{P'_0, P'_1, \cdots, P'_F\}$ supported on $V_1 \otimes v_\alpha^2$ with the property $\epsilon_{\max}(r') \leqslant \epsilon_{\max}(r)$. Note that P'_F, in the binary representation, is of the form $\bar{x} \otimes v_\alpha^2$, where $\bar{x}$ is a codeword of H_1. Therefore, P'_F is either a nontrivial elementary canonical logical operator or the identity. In the latter case, P'_F is the trivial codeword (zero vector) in the binary representation. Henceforth, we denote this as $L' = P'_F$.

If L' is nontrivial, we can use the relation between the energy barriers of L and L':

$$\Delta(L) = \epsilon_{\max}(r) \geqslant \epsilon_{\max}(r') \geqslant \Delta(L'). \tag{3.26}$$

Because L' is an elementary logical operator, $\Delta(L)$ is greater or equal to the minimum energy barrier of elementary canonical logical operators. Thus, if L' is nontrivial, the proof follows immediately.

If L' is an identity, the above argument does not work. Fortunately, it turns out that for any L', one can choose L_c (the codeword of H_2 used in the current proof) such that L' is not an identity.

Without loss of generality, consider a canonical logical-Z operator L, expressed as

$$L = \begin{pmatrix} \sum_{k,j} \lambda_{kj} \bar{x}_k \otimes y_j \\ 0_{r_1 r_2} \end{pmatrix}, \tag{3.27}$$

where (i) $H_1 \bar{x}_i = 0$ and (ii) $y_j \notin \mathrm{Im}\left(H_2^T\right)$ are unit vectors. If a given path ends with L, its deformation (using Eq. (3.25)) yields the following operator L':

$$L' = \begin{pmatrix} \sum_{k} c_k \bar{x}_k \otimes y_\alpha \\ 0_{r_1 r_2} \end{pmatrix}, \tag{3.28}$$

where $\alpha \in \mathcal{C}(L_c)$ and c_k is defined as

$$c_k := \sum_{j \in \mathcal{C}(L_c)} \lambda_{kj}. \tag{3.29}$$

Note that L' is trivial if and only if $c_k = 0$ for all k. Therefore, we aim to prove that there exists a choice of L_c such that $c_k = 1$ for at least one k.

Let us prove the contrapositive. Suppose $c_k = 0$ for all k, for any choice of L_c. Consider the following vector:

$$u_k := \sum_j \lambda_{kj} y_j. \tag{3.30}$$

Note that $c_k = u_k^T L_c$. By our assumption, $c_k = 0$ for any choice of L_c, and so the inner product of u_k with any codeword of H_2 must be zero. On the other hand, u_k, if it is nonzero, must lie outside of $\mathrm{Im}(H_2^T)$ by the definition of the y_j's. Thus, u_k is not an element of the row space of H_2. However, this is a contradiction for the following reasons. For a linear code, let H and G be the parity check matrix and the generator matrix. Then $v^T G = 0$ if and only if v is a vector in the row space of H. In our setup, if $c_k = 0$ for any L_c, then $u_k^T G_2 = 0$. This implies that u_k must be in the row space of H_2, which contradicts the fact that it lies outside of $\mathrm{Im}(H_2^T)$. To conclude, there must be at least one k such that $c_k = 1$. Thus, there is always a choice of L_c such that L' is not an identity, thereby proving the claim.

Case 2 can be analyzed similarly to Case 1 by considering rows in the subset $C_1 \otimes C_2$. For Case 3, one can treat it just as Case 1 or Case 2. For example, when treating it as Case 1, the logical operator L has a nontrivial part in the subset $V_1 \otimes V_2$. One can prove there exists a codeword L_c of H_2, such that after the deformation, the resulting L' is a nontrivial elementary canonical logical operator.

□

Let $L = (\bar{x}_k \otimes v_2^j, 0_{r_1 r_2})^T$ be an elementary canonical logical-Z operator, supported on $V_1 \otimes v_2^j$. Lemma 3.5.1 implies there is a path r supported on $V_1 \otimes v_2^j$ that attains the energy barrier of $\Delta(L)$.

Using such a path, we can prove a lower bound on the energy barrier of L. Without loss of generality, let $r = (P_0, P_1, \cdots, P_F)$ with $P_0 = I$ and $P_F = L$. Because this path is supported on $V_1 \otimes v_2^j$, in the binary representation, each P_i becomes $u_i \otimes v_2^j$ for some $u_i \in V_1$. In particular, at the i'th step, the energy is wt$(H_1 u_i)$, which is exactly the energy of the classical code with respect to $u_i \in V_1$. Because $u_1 = (0, \ldots, 0)$ and $u_F = \bar{x}_k$ for some codeword $\bar{x}_k$ of H_1, the path $(u_0, \ldots, u_F)$ must have an energy barrier of at least $\Delta(H_1)$. Similarly, for any elementary canonical logical-Z operator L on subset $c_1^\beta \otimes C_2$, one can show that $\Delta(L) \geq \Delta\left(H_2^T\right)$. Then Lemma 3.5.3 imeidately implies Proposition 3.5.1.

3.5.3 Energy barrier

Since the lower bound and the upper bound match, the energy barrier of the canonical logical-Z operators is precisely $\min(\Delta(H_1), \Delta(H_2^T))$. Similarly, the energy barrier of the canonical logical-X operators can be shown to be $\min(\Delta(H_2), \Delta(H_1^T))$. Moreover, due to Theorem 3.3.1, the energy barrier of any logical operator can be bounded in terms of the canonical logical operators' energy barrier and the code's sparsity parameter. Note that this conclusion applies to the hypergraph product of *any* two classical codes.

If the code is LDPC and the energy barrier is $\Omega(1)$, we conclude that the energy barrier of the quantum code is exactly equal to $\min(\Delta(H_1), \Delta(H_2), \Delta(H_1^T), \Delta(H_2^T))$.

THEOREM 3.5.1. *Let $\Delta(H)$ be the energy barrier of the hypergraph product code obtained from parity check matrices H_1 and H_2 of $O(1)$ row and column weight. If*

$$\Delta(H_1), \Delta(H_2), \Delta(H_1^T), \Delta(H_2^T) = \Omega(1), \tag{3.31}$$

then

$$\Delta(H) = \min(\Delta(H_1), \Delta(H_2), \Delta(H_1^T), \Delta(H_2^T)). \tag{3.32}$$

More precisely, for any (w_c, w_q) quantum code derived from a hypergraph product, if the parent classical codes' energy barriers exceed $w_c w_q$, the quantum code's energy barrier matches the minimum energy barrier of these parent codes. This can be used to prove tight bounds on the energy barrier. For instance, a classical code whose Tanner graph is a bipartite

expander graph has an extensive macroscopic energy barrier. Thus, our result implies that the quantum expander code [102, 106] also has an extensive energy barrier. While this is already known [106, 130], note that this argument does not rely on the expansion property of quantum expander code.

3.6 Expander codes

Let $G = (V \cup C, E)$ be a bipartite graph. For a vertex subset $S \subseteq V$, a vertex $g \in C$ is a neighbor of S if it is connected to at least one vertex in S, $N(S)$ is the set of neighbors of S. We say $g \in C$ is a unique neighbor of S if it is only connected to a single vertex in S. We use $U(S)$ to denote the collection of unique neighbors of S. The expander graph can be defined as follows:

DEFINITION 3.6.1 (Expander graph). *Let $G = (V \cup C, E)$ be a bipartite graph with left and right degrees bounded by w. Let $|V| = n$ and $|C| = m$. We say G is (γ_1, δ_1)-left-expanding with constants $\gamma_1, \delta_1 > 0$, if for any subset $S \subseteq V$ with $|S| \leq \gamma_1 n$, the neighbor $N(S)$ of S in the subset C satisfies:*

$$|N(S)| \geqslant w(1-\delta_1)|S|. \tag{3.33}$$

Similarly, We say G is (γ_2, δ_2)-right-expanding with constants $\gamma_2, \delta_2 > 0$, if for any subset $T \subseteq C$ with $|T| \leq \gamma_2 n$, the neighbor $N(T)$ of T in the subset V satisfies $|N(T)| \geqslant w(1-\delta_2)|T|$. We call G an $(n, m, w, \gamma_1, \delta_1, \gamma_2, \delta_2)$ expander graph if it is both left and right expanding.

Let H_G be the parity check matrix of an expander code defined by this $(n, m, w, \gamma_1, \delta_1, \gamma_2, \delta_2)$ expander graph (choosing V and C as the variable and check nodes of the tanner graph) with $\delta_1 < \frac{1}{2}$. Then for any $S \subset V$ such that $|S| \leq \gamma n, \quad U(S) \geq w(1-2\delta_1)|S|$. Furthermore, the distance d_C of the code $\mathcal{C}(G)$ is greater than $\gamma_1 n$ [143].

LEMMA 3.6.1. *let $\Delta(H_G)$ be the energy barrier of classical expander code defined by H_G with parameters $(n, m, w, \gamma_1, \delta_1, \gamma_2, \delta_2)$ and $\delta_1 < \frac{1}{2}$, then*

$$\Delta(H_G) \geq cn, \tag{3.34}$$

where $c > 0$ is a constant.

PROOF. Without loss of generality, suppose that energy barrier $\Delta(H_G)$ is equal to an energy barrier of a logical operator L, with path r_L. Since $|L| \geq \gamma n$, there is a point along the path (denoted as P) such that $|P| = \gamma n$. Because $U(P) \geq w(1-2\delta_1)|P|$, we get $\Delta(H_G) \geq U(P) \geq w(1-2\delta_1)|P| = w(1-2\delta_1)\gamma_1 n$. Thus we have $\Delta(H_G) \geq cn$ for $c = w(1-2\delta_1)\gamma_1 > 0$. □

From Lemma 3.6.1, it follows that the classical expander code has an energy barrier scaling linearly with the number of bits. A similar conclusion follows the transpose code in which the variable and the check nodes are exchanged. Therefore, the quantum expander code defined by parity check matrix $H_{(H_G,H_G)}$ has $\Omega(\sqrt{n})$ energy barrier, where n here is the number of qubits. Note that our argument does not rely on the expansion property of quantum expander code.

3.7 Outlook

We proved a tight bound on the energy barrier of the hypergraph product code, determined in terms of the energy barriers of the underlying classical codes. While it was expected that the energy barrier of the quantum code is related to its counterpart [144], we provided a first rigorous proof of this statement [Eq. (3.1)] to our knowledge. Looking forward, it would be interesting to study the energy barrier of the codes such as the homological product [5], balanced product codes [46] and lifted product codes [45]. The generalized bicycle code [145, 45], due to its simple structure, is another natural candidate to explore. Understanding how our proof technique can be generalized to these setups and how to design efficient decoders that can leverage the energy barrier is left for future work.

CHAPTER 4

Improved energy barrier of higher-dimensional hypergraph product codes

---

In the previous chapter, we established a lower bound on the energy barrier of hypergraph product codes. This bound matches the energy barriers of the underlying classical codes and their duals. In that case, constructing a quantum code with a macroscopic energy barrier requires classical codes and their duals that also feature a significant energy barrier. However, determining energy barriers for classical codes still remains difficult. Thus, one might wonder if quantum codes with a macroscopic energy barrier can be constructed when the underlying classical codes lack that property.

Here, we propose that higher-dimensional hypergraph product [110] may offer such a route. Unlike the original hypergraph product, this advanced construction uses more than two classical codes. These extra codes serve as meta-checks, which means checks of checks. These meta-checks can be used to validate stabilizer measurement outcomes.

The intuition of considering higher-dimensional hypergraph products stems from two examples: the 3D toric code and the 4D toric code. The 3D toric code is a 3D hypergraph product code built from three 1D repetition codes. One type of logical operator has an energy barrier proportional to the linear size of the system [146], while the other remains constant, as in the case of the 1D repetition code. In contrast, the 4D toric code, a 4D hypergraph product code constructed from four 1D repetition codes, achieves energy barriers proportional to the linear system size for both types (X and Z) of logical operators[52, 147, 132].

Higher-dimensional hypergraph product codes have many good properties. In 2019, Campbell demonstrated that by applying the hypergraph product twice, the resulting quantum codes have

good soundness [148]. We will show later that applying a hypergraph product twice gives a 4D hypergraph product code. Moreover, higher-dimensional codes tend to be more fault-tolerant, offering better protection against errors. They have an effective distance matching their code-theoretic distance [111]. This extends the result that stabilizer measurement schedules preserve the distance of hypergraph product codes [112]. Additionally, higher-dimensional hypergraph product codes have efficient logical gates [113] and single-shot decoding [114, 59].

Soundness is closely related to single-shot quantum error correction, a concept introduced by Bombin [149]. In quantum error correction, usually, the number of syndrome measurement rounds increases with code size due to measurement errors. However, single-shot error correction offers a novel solution by enabling error correction with only a single round of syndrome measurements, even in the presence of measurement noise. This approach significantly accelerates the error correction process and demonstrates its practical advantages. Later, Quintavalle et al. [59] demonstrated that confinement more effectively captures the general requirements for single-shot error correction than soundness.

In this chapter, we will show that the energy barrier of LDPC HHGP codes is lower bounded not only by the energy barriers of the underlying classical codes (as in standard hypergraph products), but also by the distances of these underlying classical codes. Thus to construct quantum codes with macroscopic energy barrier, one can just use classical codes that have large distances.

Notably, previous research has demonstrated that HHGP codes exhibit good soundness and confinement, making them single-shot correctable against adversarial errors [148, 59]. These soundness and confinement properties induce a lower bound on the energy barrier. However, we demonstrate that this bound is not optimal. Our approach, based on analyzing logical operator structures, provides a tighter characterization of the energy landscape for HHGP codes.

To guide the reader, our paper is structured as follows. In Sec. 4.1, we review key concepts including soundness, confinement, energy barrier and HHGP codes. In Sec. 4.2, we formally

demonstrate how confinement induces a lower bound on energy barriers and use the confinement results from Ref. [59] to establish an energy barrier lower bound for 3D hypergraph product codes. Then, in Sec. 4.3, we provide an energy barrier lower bound for tensor product codes, as formally stated in Lemma 4.3.2. This lower bound serves as a key insight for proving the energy barrier lower bound of LDPC HHGP codes. Finally, by analyzing the structure of logical operators of HHGP codes and applying Lemma 4.3.2, we derive energy barrier lower bounds for 3D and 4D hypergraph product codes in Sec. 4.4 and Sec. 4.5, respectively. These bounds, presented in Theorem 4.4.1 and Theorem 4.5.1, relate directly to the distances of the underlying classical codes.

4.1 Preliminaries

4.1.1 Quantum LDPC codes

Stabilizer codes, conceptualized by Daniel Gottesman in 1996 [35], encode quantum information in a protected subspace of a larger Hilbert space. Specifically, these codes store quantum states within the $+1$ co-eigenspace of a set of m commuting Pauli operators, $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$, defined on an n-qubit Hilbert space $\mathcal{H}_2^{\otimes n}$. Each operator $S_i \in \{I, X, Y, Z\}^{\otimes n}$ consists of a tensor product of the Pauli operators, with the requirement that all these operators commute with each other. $\mathcal{S}$ excludes the operator $-I$ to ensure a non-trivial code space. Logical operators are defined as the elements of $\mathcal{C}(S)\backslash S$.

Calderbank-Shor-Steane (CSS) codes [36, 34] are an important family of stabilizer codes with special structural properties. Their stabilizer generators are exclusively either X-type or Z-type Pauli operators. This separation allows all X-type stabilizers to be represented by a parity check matrix H_X, while all Z-type stabilizers are represented by a parity check matrix H_Z. Consequently, the parity check matrix for the quantum CSS code takes the form

$$H = \begin{pmatrix} 0 & H_Z \\ H_X & 0 \end{pmatrix}, \tag{4.1}$$

with the condition $H_X H_Z^T = 0$ ensuring that all stabilizers commute with each other.

Quantum LDPC codes are stabilizer codes with sparse parity check matrices (see [37, 38] for good reviews). The sparsity is defined by two parameters: w_c, the maximum weight amongst all stabilizer generators, and w_q, the maximum number of stabilizer generators associated with any single qubit. Formally, a code is classified as LDPC if both w_c and w_q remain constant as the code size increases, specifically $w_c, w_q = O(1)$.

Quantum LDPC codes are promising because they often provide lower overhead and can be designed for fault-tolerance [39]. Recent research has led to several families of quantum LDPC codes with enhanced parameters [41, 42, 43, 44, 45, 46, 150], with some efforts even achieving good quantum LDPC codes [47, 48, 49].

Various methods exist for constructing quantum LDPC codes. From a mathematical perspective, the hypergraph product construction [40, 110] represents a well-established approach, providing a systematic way to construct quantum codes from any classical codes. Alternative methods such as two-block group-algebra (2BGA) codes [151, 152] offer different advantages, like suitability for two-dimensional layouts [153], making them useful for practical quantum computing architectures.

4.1.2 Energy barrier of codes

Please see the definition in previous chapter. Here, we emphasize that that there are two main obstacles for computing the energy barrier of quantum codes. First, numerous paths exist for implementing any logical operator, creating a vast solution space to explore. Second, in quantum codes, any stabilizer can be multiplied with a logical operator to produce an equivalent logical operator, while for those equivalent logical operators, the energy barriers may differ. That is to say, for a logical operator L and stabilizer S, usually $\Delta(L) \neq \Delta(LS)$. This complicates the analysis as one must consider the full equivalence class of logical operators.

Quantum LDPC codes offer significant advantages for energy barrier computation by reducing analytical complexity. As shown in Ref. [109], in LDPC codes, two logical operators that are equivalent under stabilizers have identical energy barriers, provided at least one of their energy

barriers equals or exceeds $w_c w_q$, the product of sparsity parameters. With $w_c w_q = O(1)$, this property yields a powerful simplification: if any non-trivial logical operator has an energy barrier of $\Omega(1)$, all equivalent logical operators share the same energy barrier.

Therefore, to determine the energy barriers of quantum LDPC codes, one needs only analyze a fixed complete set of logical operators. Once the energy barrier is established for this complete set, the energy barrier of the code follows directly from these results.

4.1.3 Soundness

Soundness originated in the study of locally testable codes (LTCs) [154, 155, 156, 157]. LTCs are a class of error-correcting codes that allow efficient verification of whether a string is a valid codeword or is far from the code space. The soundness of a code, denoted by $R(\delta)$, measures the likelihood that a randomly selected local constraint is violated by a string that is at Hamming distance at least δn from the code space, where n is the codeword length. The concept of locally testability has deeply influenced fields like PCP, combinatorial optimization, property testing, program verification, and cryptography.

Aharonov and Eldar introduced quantum locally testable codes (QLTCs) [158], with interest surging after Eldar and Harrow showed [159] that QLTCs with constant soundness, locality, and relative distance could construct Hamiltonians lacking low-energy trivial states—addressing the NLTS conjecture [160]. QLTCs are closely related to the quantum PCP conjecture [161], a fundamental problem in quantum complexity theory related to the quantum analog of the classical PCP theorem.

Roughly speaking, the soundness property ensures that high-weight errors produce high-weight syndromes. The following is the formal definition of soundness provided by Aharonov and Eldar [158].

DEFINITION 4.1.1 (Quantum locally testable code, Definition 14 of [158]). *Let $R = R(\delta)$ be some function $R(\delta) : [0,1] \mapsto [0,1]$, this is called the soundness function. Let C be a quantum code on n d-dimensional qudits, defined as the groundspace of $H = \sum_{i=1}^{m} \Pi_C^i$, where Π_C^i are*

m k-local projections for some constant k. We say that C is quantum locally testable with soundness $R(\delta)$, if:

$$\forall \delta > 0, |\Psi\rangle : \quad \operatorname{dist}(|\Psi\rangle, C) \geq \delta n \mapsto \frac{1}{m}\langle\Psi|H|\Psi\rangle \geq R(\delta), \tag{4.2}$$

where $\operatorname{dist}(\cdot)$ denotes the distance between the state $|\Psi\rangle$ and the code space C, defined as the minimal Hamming distance between $|\Psi\rangle$ and any state in C. The query complexity of the code is defined to be k.

For stabilizer codes, soundness can be characterized as follows. Consider a code C defined on n qubits, determined by a generating set $\mathcal{G}$ where each generator has support over w qubits (representing the stabilizer size). For any error E acting on the code with a syndrome weight of at least δn, a randomly chosen generator $g \in \mathcal{G}$ fails to commute with E with probability at least $R(\delta)$. This property ensures that errors can be detected probabilistically by examining only a limited number of stabilizers. Additionally, this soundness property induces a macroscopic energy barrier [162].

To analyze single-shot property of quantum codes, Campbell provided the following alternative definition of soundness [148].

DEFINITION 4.1.2 (Soundness, Definition 3 in [148]). *Let t be an integer and $f : \mathbb{Z} \to \mathbb{R}$ be a function termed the soundness function, where $f(0) = 0$. A set of Pauli stabilizers $\mathcal{M}$ is called (t, f)-sound if, for every Pauli error E with syndrome weight $|\sigma(E)| = x < t$, there exists an $E^\star$ such that $\sigma(E^\star) = \sigma(E)$ and the weight wt $(E^\star) \leqslant f(x)$.*

In essence, this definition requires that low-weight syndromes are caused by low-weight errors. This formulation represents the contrapositive of definition of locally testable codes, with the key distinction being its focus on scenarios where $|\sigma(E)| = x < t$. Under this definition, good soundness is characterized by function $f(x)$, which is a monotonically increasing polynomial function of x that remains independent of the size of the check set.

Campbell demonstrated that codes possessing good soundness naturally exhibit single-shot error correction capabilities, with the specific performance determined by the soundness

parameters t, f and the code parameters $[n, k, d]$ [148]. Moreover, good soundness in LDPC codes implies the existence of a macroscopic energy barrier [158, 148], as formalized in the following lemma:

LEMMA 4.1.1 (Lemma 3 in [148]). *Consider a $[[n, k, d]]$ quantum code with checks $\mathcal{M}$ that is (t, f)-sound and where all qubits are involved in no more than w_c checks. It follows that the energy barrier is at least $f^{-1}(c)$ where $c = \min\left[(t-1)/w_c, (d-1)/2\right]$ and f^{-1} is the inverse of the soundness function.*

It is important to note that the converse statement, any LDPC check family with a macroscopic energy barrier necessarily possesses good soundness, does not hold. The expander code provides a clear counterexample: despite exhibiting bad soundness due to its lack of check redundancy [163], it has been proven to support single-shot error correction with the small-set flip decoder [106, 164]. In this context, check redundancy refers to the ratio between the check set size (encompassing all checks, including dependent ones) and the number of independent checks.

The soundness of a stabilizer code depends specifically on the selection of the stabilizer set, not merely on the stabilizer generators. Campbell also demonstrated that for any stabilizer code, it is possible to strategically choose a stabilizer set that endows the code with good soundness properties [148]. The price is that the size of some checks could be excessively large. LDPC property may be lost due to the existence of high-weight checks.

To maintain the LDPC characteristic, Campbell discovered an elegant solution for achieving single-shot properties: applying the hypergraph product construction twice [148]. The resulting structures naturally possess meta-checks, a direct consequence of the iterative hypergraph product process. Check redundancy first induces meta-checks. With proper design of the check redundancy, these meta-checks can be used to verify whether the checks (or stabilizer measurement results) are correct. This potentially provides single-shot capability, meaning decoding can be performed using a single round of measurement results. This is possible because the meta-checks can first be used to correct the checks themselves, ensuring sufficient fidelity such that the error after decoding remains bounded. To make the thesis

more clear, I have added this discussion to the main text. Through this approach, Campbell effectively combined soundness and LDPC properties by strategically leveraging check redundancy. The following lemma shows this formally.

LEMMA 4.1.2 (Lemma 5 and Lemma 6 in [148]). *Let $C_0 \xleftarrow{\partial_0} C_1$ be a chain complex. Applying the hypergraph product, we obtain a new chain complex $\tilde{C}_{-1} \xleftarrow{\tilde{\delta}_{-1}} \tilde{C}_0 \xleftarrow{\tilde{\delta}_0} \tilde{C}_1$, where the maps $\tilde{\delta}_0^T$ and $\tilde{\delta}_{-1}$ are (t, f)-sound with $f(x) = x^2/4$ and $t = \min\left[d_0, d_0^T\right]$, where d_0, d_0^T are the distances of codes defined by δ_0, δ_0^T, respectively. Applying the hypergraph products again, we obtain a new chain complex $\breve{C}_{-2} \xleftarrow{\breve{\delta}_{-2}} \breve{C}_{-1} \xleftarrow{\breve{\delta}_{-1}} \breve{C}_0 \xleftarrow{\breve{\delta}_0} \breve{C}_1 \xleftarrow{\breve{\delta}_1} \breve{C}_2$, where the maps $\breve{\delta}_0$ and $\breve{\delta}_{-1}^T$ are (t, g)-sound with soundness function $g(x) = x^3/4$, and $t = \min\left[d_0, d_0^T\right]$.*

For a 4D hypergraph product code that constructed with four identical classical codes defined by the same parity check matrix δ with distance d (δ^T with distance d^T), according to Lemma 4.1.1, the energy barrier for logical operators of the resulting quantum code then has a lower bound of $\Omega\left((\min[d, d^T])^{\frac{1}{3}}\right)$. Later, we will demonstrate that the optimal lower bound for this energy barrier scales as $\Omega(\min[d, d^T])$ in the given scenario.

4.1.4 Confinement

Confinement is another robustness measure of quantum codes. Bombín originally introduced the concept of confinement as a mechanism that allows quantum codes to achieve single-shot error correction [149]. However, in this discussion, we use a more developed definition provided in [59].

DEFINITION 4.1.3 (Confinement, Definition 1 in [59]). *Let t be an integer and $f : \mathbb{Z} \to \mathbb{Z}$ some increasing function with $f(0) = 0$. We say that a stabilizer code has (t, f)-confinement if, for all errors e with $\mathrm{wt}_{\mathrm{red}}(e) \leq t$, it holds*

$$f(|\sigma(e)|) \geq \mathrm{wt}_{\mathrm{red}}(e), \tag{4.3}$$

where $\sigma(e)$ is the syndrome of the error e.

In this definition, $\mathrm{wt}_{\mathrm{red}}(e)$ denotes the reduced weight of the error e, which represents the smallest weight of any error e' that produces the same error syndrome as e.

Unlike Bombín's more restrictive formulation (Definition 16 in [149]), the above definition permits nonlinear functions $f(x)$. 2D repetition code and 3D hypergraph product codes both exhibit superlinear confinement functions while supporting single-shot correction. These important cases would be excluded under Bombín's original framework.

A code family is said to have good confinement if each code within the family satisfies (t, f)-confinement with the following criteria: (1) t increases with the length of the code n, specifically $t \in \Omega\left(n^b\right)$ for some $b > 0$, and (2) the confinement function $f(\cdot)$ increases monotonically and does not depend on n. Additionally, a code family is said to exhibit good X-confinement if this property applies only to Pauli Z errors [59].

Confinement is a weaker requirement than soundness. While good soundness requires low-weight syndromes to correspond to low-weight errors, good confinement demands that low-weight errors produce low-weight syndromes. For an LDPC code that is (t, f)-sound, if the maximum qubit degree is ω_q (the sparsity parameter), then the code has $(t/\omega_q, f)$ confinement (see Lemma 2 in [59]). This follows from a direct analysis: if e is an error set with $\mathrm{wt}_{\mathrm{red}}(e) \leq t/\omega_q$, then its syndrome satisfies $|\sigma(e)| \leq t/\omega_q \cdot \omega_q = t$. By the soundness of the code, we have $f(|\sigma(e)|) \geq \mathrm{wt}_{\mathrm{red}}(e)$.

Importantly, one can show that codes with good confinement possess single-shot error correction capabilities [59].

LEMMA 4.1.3 (Theorem 1 in [59]). *Consider a family of $[[n, k, d]]$ quantum-LDPC codes with good confinement such that $d \geq an^b$ with $a, b > 0$. This code family is single-shot for the adversarial noise model. If the code family only has good X-confinement then it is single-shot with respect to Pauli Z noise.*

Confinement is also related to the energy barrier of a code [149, 130, 165]. In Sec. 4.2, we demonstrate formally that confinement naturally induces an energy barrier lower bound.

Researches have shown that good soundness requires check redundancy in LDPC codes [163, 166], the expander code exhibits poor soundness due to its lack of check redundancy. However, despite this limitation, the quantum expander code has been proven to support single-shot error correction when implemented with the small-set flip decoder [106, 164]. This makes confinement a more general and inclusive property compared to soundness, in relation to single-shot error correction.

4.1.5 Higher-dimensional hypergraph product

The hypergraph products provide a way to construct quantum codes with any two classical linear codes [40]. It enables the application of extensive insights from classical coding theory to quantum coding. Those codes have good properties, such as circuit-level distance preservation [112, 167].

Given two classical codes C_1 and C_2, with respective parity check matrices H_1 and H_2, the hypergraph product defines a quantum code with parity check matrices

$$\begin{aligned} H_X &= \begin{pmatrix} H_1 \otimes I & I \otimes H_2^T \end{pmatrix}, \\ H_Z &= \begin{pmatrix} I \otimes H_2 & H_1^T \otimes I \end{pmatrix}. \end{aligned} \tag{4.4}$$

One can verify that $H_X H_Z^T = 0$ within the field $\mathbb{F}_2$. Thus these two parity check matrices define a CSS code, with a quantum parity check matrix of the form

$$H_{(H_1,H_2)} = \begin{pmatrix} H_X & 0 \\ 0 & H_Z \end{pmatrix}. \tag{4.5}$$

Hypergraph product codes can be interpreted through chain complexes. Classical codes correspond to length-2 chain complexes, where boundary maps define parity check matrices. A hypergraph product code arises from the tensor product of two such chain complexes, yielding a length-3 chain complex. This structure provides two boundary maps that serve as parity checks. The fundamental property that "a boundary has no boundary" ensures these checks commute, allowing the construction of CSS quantum codes directly from the boundary maps of the resulting length-3 chain complex.

HHGP codes [110] extends this concept by generalizing the construction to create a length-n chain complex. This generalization has important connections to topological codes: n-dimensional toric codes on hypercubic lattices constitute a length-n chain complex [168, 52], and can be perfectly described by higher-dimensional hypergraph products.

Following Ref. [110], we give a short introduction to the chain complex interpretation of hypergraph product. A chain complex is a sequence of finite-dimensional vector spaces $\{\ldots, \mathcal{A}_{j-1}, \mathcal{A}_j, \ldots\}$ connected by boundary operators $\partial_j : \mathcal{A}_{j-1} \leftarrow \mathcal{A}_j$, satisfying $\partial_j \partial_{j+1} = 0$ for all $j \in \mathbb{Z}$.

For qubit systems, we consider vector spaces $\mathcal{A}_j = \mathbb{F}_2^{n_j}$, consisting of binary vectors of length n_j. $\mathcal{A} \equiv \mathcal{K}(A_1, \ldots, A_m)$ can be defined as a length-$(m+1)$ chain complex with boundary map ∂_j represented by $n_{j-1} \times n_j$ binary matrices A_j:

$$\mathcal{A} : \{0\} \overset{\partial_0}{\leftarrow} \mathcal{A}_0 \overset{A_1}{\leftarrow} \mathcal{A}_1 \ldots \overset{A_m}{\leftarrow} \mathcal{A}_m \overset{\partial_{m+1}}{\leftarrow} \{0\}, \tag{4.6}$$

where adjacent matrices satisfy the orthogonality condition $A_{j-1}A_j = 0$ for $j \in \{1, \ldots, m\}$. The trivial operators $\partial_0 : \{0\} \leftarrow \mathcal{A}_0$ and $\partial_{m+1} : \mathcal{A}_m \leftarrow \{0\}$ are treated as zero matrices of dimensions $0 \times n_0$ and $n_m \times 0$, respectively.

The subspace $\mathrm{Im}(A_{j+1}) \subseteq \mathcal{A}_j$ consists of boundaries, which can be represented as linear combinations of the columns of A_{j+1}. This subspace defines a binary linear code with generator matrix A_{j+1}^T, denoted as $\mathrm{Im}(A_{j+1}) = \mathcal{C}_{A_{j+1}^T}$. For the special case where $j = m$, we have $\mathrm{Im}(\partial_{m+1}) = 0$, which forms a trivial vector space.

The subspace $\mathrm{Ker}(A_j) \subseteq \mathcal{A}_j$ consists of cycles, which are vectors $x \in \mathcal{A}_j$ orthogonal to the rows of A_j, satisfying $A_j x^T = 0$. This subspace defines a binary linear code with parity check matrix A_j, expressed as $\mathrm{Ker}(A_j) = \mathcal{C}_{A_j}^{\perp}$. In the special case where $j = 0$, we have $\mathrm{Ker}(\partial_0) = \mathcal{A}_0$, encompassing the entire vector space.

Within the binary vector space $\mathcal{A}_j$, we have defined two key subspaces: $\mathrm{Im}(A_{j+1})$, consisting of boundaries, and $\mathrm{Ker}(A_j)$, comprising cycles. A fundamental property of chain complexes is that all boundaries are cycles, though not all cycles are boundaries. This relationship is mathematically expressed through the orthogonality condition $A_j A_{j+1} = 0$.

Based on these subspaces, we can define the j-th homology group as

$$H_j(\mathcal{A}) \equiv \mathrm{Ker}\,(A_j) \,/\, \mathrm{Im}\,(A_{j+1})\,. \tag{4.7}$$

This quotient space contains all elements that are cycles but not boundaries, which we refer to as homologically nontrivial cycles.

The rank of the j-th homology group, representing the dimension of the associated vector space, is given by

$$k_j \equiv \mathrm{rank}\, H_j(\mathcal{A}) = n_j - \mathrm{rank}\, A_j - \mathrm{rank}\, A_{j+1}. \tag{4.8}$$

Additionally, the homological distance d_j is defined as the minimum Hamming weight of a nontrivial element in $H_j(\mathcal{A})$, expressed as:

$$d_j = \min\{\mathrm{wt}(x) : x \in H_j(\mathcal{A}), x \neq 0\}. \tag{4.9}$$

A quantum CSS code can be derived from a length-3 chain complex $\mathcal{A}_0 \overset{A_1}{\leftarrow} \mathcal{A}_1 \overset{A_2}{\leftarrow} \mathcal{A}_2$ by setting $H_Z^T = A_1$ and $H_X = A_2$. In chain complex terminology, the code length equals $\dim \mathcal{A}_1$, while its dimension corresponds to the first homology group $\mathcal{H}_1 = \ker A_1 / \mathrm{Im}\, A_2$, or equivalently, the first cohomology group $\mathcal{H}_1^* = \ker A_1^T / \mathrm{Im}\, A_2^T$. The X and Z distances are determined by the minimum weights of non-zero vectors in $\mathcal{H}_1^*$ and $\mathcal{H}_1$ respectively.

Classical codes can be described by length-2 chain complexes. To construct quantum codes from classical codes, it is just a question of how to build a length-3 chain complex from two length-2 chain complexes. For this purpose, we employ the tensor product of chain complexes.

The tensor product $\mathcal{A} \times \mathcal{B}$ of two chain complexes $\mathcal{A}$ and $\mathcal{B}$ is defined as a new chain complex in the following way

$$(\mathcal{A} \times \mathcal{B})_l = \bigoplus_{i+j=l} \mathcal{A}_i \otimes \mathcal{B}_j. \tag{4.10}$$

The vector spaces in the new chain complex are direct sums of Kronecker products of vector spaces from the original chain complexes. Moreover, the boundary operators in this tensor product construction act as

$$\partial_{i+j}(a \otimes b) = \partial'_i a \otimes b + (-1)^i a \otimes \partial''_j b, \tag{4.11}$$

where $a \in \mathcal{A}_i, b \in \mathcal{B}_j$, and ∂'_i and ∂''_j are the boundary operators in the corresponding space $\mathcal{A}$ and $\mathcal{B}$. If both $\mathcal{A}$ and $\mathcal{B}$ have finite dimension, then the dimension of $\mathcal{C} = \mathcal{A} \times \mathcal{B}$ is

$$n_j(\mathcal{C}) = \sum_i n_i(\mathcal{A}) n_{j-i}(\mathcal{B}). \tag{4.12}$$

By the Künneth theorem, the homology groups of the product complex $\mathcal{C} = \mathcal{A} \times \mathcal{B}$ are given by

$$H_j(\mathcal{C}) \cong \bigoplus_i H_i(\mathcal{A}) \otimes H_{j-i}(\mathcal{B}). \tag{4.13}$$

Consequently, the rank $k_j(\mathcal{C})$ of the j-th homology group $H_j(\mathcal{C})$ can be expressed as

$$k_j(\mathcal{C}) = \sum_i k_i(\mathcal{A}) k_{j-i}(\mathcal{B}). \tag{4.14}$$

The hypergraph product code is defined as the tensor product of two length-2 chain complexes. The higher-dimensional hypergraph product extends this concept to the tensor product of two chain complexes of arbitrary lengths. As demonstrated in Ref. [110], for HHGP codes formed from a length-m chain complex $\mathcal{A}$ and a length-l chain complex $\mathcal{B}$ with $l = 2$, there exists a lower bound on the distance of logical operators in the resulting quantum codes:

$$d_j(\mathcal{A} \times \mathcal{B}) = \min\left[d_{j-1}(\mathcal{A}) d_1(\mathcal{B}), d_j(\mathcal{A}) d_0(\mathcal{B})\right]. \tag{4.15}$$

In this work, we focus specifically on those 3D and 4D hypergraph product codes that can be regarded as tensor products of three or four classical codes, respectively. This allows us to apply the distance bounds given by above equation.

4.2 Confinement and energy barrier

The relation between confinement and energy barrier is well-known [149, 130, 165]. In this section, we formalize this connection through the following lemma.

LEMMA 4.2.1. *For a $[[n, k, d]]$ quantum code $\mathcal{C}$ with (t, f) confinement where $t \leqslant d$ and $f(\cdot)$ is increasing monotonically, the energy barrier of the code is at least $f^{-1}(w)$, where $w = \min[t, (d-1)/2]$.*

PROOF. Without loss of generality, suppose that the code energy barrier $\Delta(\mathcal{C})$ is given by logical operator L, with a path r_L. Since $|L| \geqslant d \geqslant t$, we first show that there always exist a step P_c along the path r_L, such that $\mathrm{wt}_{\mathrm{red}}(P_c) \geqslant (d-1)/2$. This can be argued as follows [148]:

Suppose $r_L = \{P_0, P_1, \cdots, P_L\}$, and each $E_j = P_j P_{j-1}$ is single qubit Pauli operator. For each P_j, we define the reduced weight as:

$$\mathrm{wt}_{\mathrm{red}}(P_j) := \min_{V} \{\mathrm{wt}(P_j V) : V \in \mathcal{P}, \sigma(V) = 0\}, \tag{4.16}$$

where the minimization is over Pauli operators V with trivial syndrome. Let V_j denote the Pauli operators achieving this minimum for P_j. Since $\sigma(V_j) = 0$, V_j is either a stabilizer or a nontrivial logical operator.

Trivially, we find $V_0 = I$ and $V_L = P_L$, indicating that the sequence starts with a stabilizer and ends with a logical operator. Therefore, there exists an index j^* such that V_{j^*} is a stabilizer, while V_{j^*+1} is a nontrivial logical operator. Then $V_{j^*} V_{j^*+1}$ is also a nontrivial logical operator, leading to:

$$d \leq \mathrm{wt}\left(V_{j^*} V_{j^*+1}\right). \tag{4.17}$$

Notice that

$$\begin{aligned} \mathrm{wt}\left(V_{j^*}V_{j^*+1}\right) &= \mathrm{wt}\left(V_{j^*}V_{j^*+1}P_{j^*}P_{j^*}^{\dagger}P_{j^*+1}P_{j^*+1}^{\dagger}\right) \\ &= \mathrm{wt}\left(V_{j^*}P_{j^*}V_{j^*+1}P_{j^*+1}P_{j^*}^{\dagger}P_{j^*+1}^{\dagger}\right). \end{aligned} \tag{4.18}$$

Applying the triangle inequality twice gives:

$$\mathrm{wt}\left(V_{j^*}V_{j^*+1}\right) \leq \mathrm{wt}_{\mathrm{red}}\left(P_{j^*}\right) + \mathrm{wt}_{\mathrm{red}}\left(P_{j^*+1}\right) + 1. \tag{4.19}$$

Thus, combining this result, we obtain:

$$d \leq 2 \cdot \max\left[\mathrm{wt}_{\mathrm{red}}\left(P_{j^*}\right), \mathrm{wt}_{\mathrm{red}}\left(P_{j^*+1}\right)\right] + 1, \tag{4.20}$$

which implies

$$\frac{d-1}{2} \leq \max\left[\mathrm{wt}_{\mathrm{red}}\left(P_{j^*}\right), \mathrm{wt}_{\mathrm{red}}\left(P_{j^*+1}\right)\right]. \tag{4.21}$$

Thus, the sequence of reduced weights $\left\{\mathrm{wt}_{\mathrm{red}}\left(P_0\right), \mathrm{wt}_{\mathrm{red}}\left(P_1\right), \cdots, \mathrm{wt}_{\mathrm{red}}\left(P_L\right)\right\}$ starts and ends with zero, and it reach a value of at least $(d-1)/2$ in the middle. Moreover, the local error condition ensures that $\left|\mathrm{wt}_{\mathrm{red}}\left(P_{j+1}\right) - \mathrm{wt}_{\mathrm{red}}\left(P_j\right)\right|$ is either 0 or 1, then the sequence must include every integer from 0 to $(d-1)/2$.

Recall that (t, f) confinement means that for all errors e with $\mathrm{wt}_{\mathrm{red}}(e) \leq t$, $f(|\sigma(e)|) \geq \mathrm{wt}_{\mathrm{red}}(e)$. Then for $w = \min[t, (d-1)/2]$, there must exist an P_w with $\mathrm{wt}_{\mathrm{red}}(P_w) = w$, such that $f(|\sigma(P_w)|) \geqslant \mathrm{wt}_{\mathrm{red}}(P_w) = w$. Given $f(\cdot)$ is monotonically increasing, $|\sigma(P_c)| \geqslant f^{-1}(w)$. This implies that the energy barrier of code $\mathcal{C}$ is at least $f^{-1}(w)$. □

The converse statement of Lemma 4.2.1—that any quantum code with a macroscopic energy barrier necessarily exhibits confinement—does not generally hold. This can be demonstrated with a straightforward counterexample. Consider a quantum LDPC code $\mathcal{C}$ constructed as a composite system comprising two regions: Region A, based on a 2D quantum repetition code $[[n, 1, n]]$ with energy barrier $O(\sqrt{n})$, and Region B, based on a 1D quantum repetition code $[[n, 1, n]]$ with constant energy barrier $O(1)$, see Fig. 4.1. The overall code is a $[[2n, 1, 2n]]$

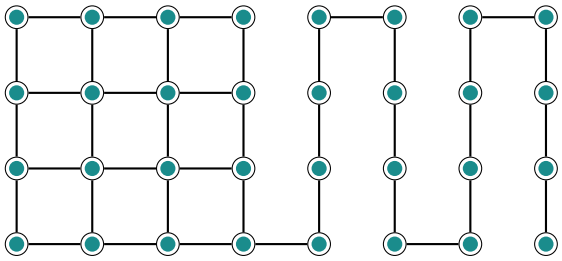


FIGURE 4.1: A composite system consisting of a 2D quantum repetition code (Region A) and a 1D quantum repetition code arranged in a snake pattern (Region B). The overall code exhibits an energy barrier determined by the 2D quantum repetition code, which scales as the linear dimension of Region A. However, confinement properties are influenced by the 1D quantum repetition component, resulting in the absence of confinement for the composite system.

quantum repetition code with energy barrier $O(\sqrt{n})$ determined by Region A. However, for an error e contained entirely within Region B, we can have $\mathrm{wt}_{\mathrm{red}}(e) = |e|$ that grows up to $(d-1)/2$ with $|\sigma(e)| \leq c$, where c is constant. Consequently, the overall code fails to satisfy the confinement criterion.

Quintavalle et al. proved that all 3D hypergraph product codes exhibit (t, f) X-confinement (according to their convention) where t equals the minimum distance of classical codes and $f(x) = x^3/4$ (see Lemma 10 of Ref [59]). For a 3D hypergraph product code constructed with three identical classical codes defined by the same parity check matrix δ with distance d, the result in Ref [59] implies a lower bound for the energy barrier of Z-type logical operators as $\Omega\left(d^{\frac{1}{3}}\right)$. In section 4.4, we will demonstrate that the optimal lower bound for this energy barrier actually scales as $\Omega(d)$ in this scenario.

4.3 Tensor product code and its energy barrier

In this section, we analyze the energy barrier of tensor product codes — a specific classical code structure that builds new classical codes from two existing ones. We will demonstrate that the energy barrier of the resulting code is related to the distance of the underlying classical codes.

This result is essential for our subsequent analysis of HHGP codes. Specifically, we establish that the logical operators of HHGP codes admit a tensor product decomposition. This structural property enables us to derive a lower bound on the energy barrier of HHGP codes in terms of the distances of their constituent classical codes.

4.3.1 Code construction

Given two classical codes $\mathcal{C}_a$ and $\mathcal{C}_b$ with parity check matrix δ_a and δ_b, the tensor product code $\mathcal{C}_c$ is a classical code defined to have the parity check matrix

$$\delta_c = \begin{pmatrix} \delta_a \otimes I_{n_b} \\ I_{n_a} \otimes \delta_b \end{pmatrix}. \tag{4.22}$$

This tensor product structure can be also understood from the perspective of code space. Let $\mathcal{C}_a$ with parameters $[n_a, k_a, d_a]$ and $\mathcal{C}_b$ with parameters $[n_b, k_b, d_b]$ be two classical codes. The tensor product of these two codes, denoted as $\mathcal{C}_c = \mathcal{C}_a \otimes \mathcal{C}_b$, encodes messages as follows: The message of the tensor product code $\mathcal{C}_a \otimes \mathcal{C}_b$ can be represented as a $k_a \times k_b$ matrix M. The encoding proceeds in two stages: first, each column of M is encoded using $\mathcal{C}_a$, yielding a $n_a \times k_b$ intermediate matrix; second, each row of this intermediate matrix is encoded using $\mathcal{C}_b$, producing the final $n_a \times n_b$ codeword.

Let L_a be the logical operator of δ_a, and L_b be the logical operator of δ_b, one can verify that $L_c = L_a \otimes L_b$ is the logical operator of δ_c because

$$\begin{aligned} \delta_c L_c &= \begin{pmatrix} \delta_a \otimes I_{n_b} \\ I_{n_a} \otimes \delta_b \end{pmatrix} (L_a \otimes L_b) \\ &= \begin{pmatrix} (\delta_a \otimes I_{n_b})(L_a \otimes L_b) \\ (I_{n_a} \otimes \delta_b)(L_a \otimes L_b) \end{pmatrix} \\ &= 0. \end{aligned} \tag{4.23}$$

Here we used the relation $\delta_a L_a = \delta_b L_b = 0$.

Given k_a distinct logical operators L_a and k_b distinct logical operators L_b, one can construct $k_a k_b$ distinct logical operators L_c through this tensor product construction.

Moreover, the distance of the tensor product code is at least $d_a d_b$. A simple argument demonstrates this: consider two different message matrices of code $\mathcal{C}_c$, denoted as M_a and M_b. After the first encoding step, we obtain intermediate matrices N_a and N_b, and the second encoding step produces the final codewords C_a and C_b. If M_a and M_b differ in the i-th row, then N_a and N_b differ in at least d_a positions in that row, corresponding to d_a columns. Let the indices of these differing columns be $\{j_1, \ldots, j_{d_a}\}$. During the second encoding step, each of these columns generates differences in at least d_b positions. Consequently, C_a and C_b differ in at least $d_a d_b$ positions. In summary, the tensor product code δ_c has parameters $[n_a n_b, k_a k_b, d_a d_b]$.

4.3.2 Energy barrier of tensor product code

Let L_a and L_b be the logical operators of δ_a and δ_b. Our goal is to determine the energy barrier of the operator $L_c = L_a \otimes L_b$, which serves as a logical operator for δ_c.

To compute the energy barrier of L_c, we consider a path $r = \{P_0, P_1, \cdots, P_F\}$ with $P_0 = I$ and $P_F = L_c$. Then

$$\Delta(L_c) = \min\{\epsilon_{\max}(r) : r \in w(0, L_c)\}. \tag{4.24}$$

Note that r is a path on the bit of code δ_c. Given a $P_\ell \in r$, the energy (the number of violated checks) of P_ℓ is

$$\begin{aligned} \epsilon(P_\ell) &= \text{wt}(\delta_c P_\ell) \\ &= \text{wt}\left(\begin{pmatrix} \delta_a \otimes I_{n_b} \\ I_{n_a} \otimes \delta_b \end{pmatrix} P_\ell\right) \\ &= \text{wt}((\delta_a \otimes I_{n_b})P_\ell) + \text{wt}((I_{n_a} \otimes \delta_b)P_\ell). \end{aligned} \tag{4.25}$$

Because $\text{wt}(\cdot) \geqslant 0$, then we have

$$\epsilon(P_\ell) \geqslant \text{wt}((\delta_a \otimes I_{n_b})P_\ell), \tag{4.26}$$

or

$$\epsilon(P_\ell) \geqslant \text{wt}((I_{n_a} \otimes \delta_b)P_\ell). \tag{4.27}$$

Because the path r contains paths for L_a and L_b, the above inequalities indicate that the energy barrier of L_c is lower bounded by the energy barrier of L_a or L_b. Furthermore, the energy barrier of code δ_c is then lower bounded by energy barrier of codes δ_a and δ_b. We conclude this as the following Lemma

LEMMA 4.3.1. *Given two classical codes δ_a and δ_b with parameters $[n_a, k_a, d_a, E_a]$ and $[n_b, k_b, d_b, E_b]$, where E_a and E_b are energy barriers of codes. Then the energy barrier of the tensor product code δ_c is lower bounded by*

$$\Delta(\delta_c) \geqslant \min[E_a, E_b]. \tag{4.28}$$

However, this may not be the tightest bound, particularly when codes δ_a and δ_b have constant energy barriers. In the following, we establish a relationship between the energy barrier of L_c and the distance of L_a and L_b.

4.3.2.1 Lower bound

Given two classical codes δ_a and δ_b with energy barriers $\Delta(\delta_a) = E_a$ and $\Delta(\delta_b) = E_b$. Here, we show that the energy barrier of the tensor product code δ_c can be lower bounded by the distances of the two constituent codes. Formally, we state the following lemma.

LEMMA 4.3.2. *Given two classical codes δ_a and δ_b with parameters $[n_a, k_a, d_a, E_a]$ and $[n_b, k_b, d_b, E_b]$. The energy barrier of the tensor product code δ_c is lower bounded by*

$$\Delta(\delta_c) \geqslant \min[d_a, d_b]. \tag{4.29}$$

PROOF. Let us first establish a clear representation of the tensor product code structure. Given the parity check matrices $\delta_a \in \{0,1\}^{m_a \times n_a}$ and $\delta_b \in \{0,1\}^{m_b \times n_b}$, we can arrange the code words of δ_c on a two-dimensional grid of size $n_a \times n_b$. Then, each row of the 2D grid corresponds to a copy of code δ_b, and each column corresponds to a copy of code δ_a.

The logical operator $L_c = L_a \otimes L_b$ can be visualized as a pattern on this grid, where L_a specifies which rows are involved and L_b specifies which columns are involved. Specifically, the support of L_c, denoted as $\mathrm{supp}(L_c)$, consists of positions (i, j) where $i \in \mathrm{supp}(L_a)$ and $j \in \mathrm{supp}(L_b)$, forming a subgrid of size $|L_a| \cdot |L_b| = d_a d_b$.

Any valid path $r = \{P_r^0, P_r^1, P_r^2, \ldots, P_r^L\}$ from the identity operator I to the logical operator L_c, by definition, must flip all the bits in the subset $\mathrm{supp}(L_c)$. For any intermediate state P_r^i along this path, the syndrome weight $\mathrm{wt}(\delta_c P_r^i)$ measures the energy cost at that step. To finish the proof, We show that for any valid path, there must exist a intermediate state with syndrome weight at least $\min[d_a, d_b]$.

To flip all bits in $\mathrm{supp}(L_c)$, the path must eventually form a logical operator in each of the d_a rows indexed by $\mathrm{supp}(L_a)$, as well as in each of the d_b columns indexed by $\mathrm{supp}(L_b)$. More precisely, in each row $i \in \mathrm{supp}(L_a)$, the final configuration must contain a codeword of δ_b. Similarly, each column $j \in \mathrm{supp}(L_b)$ must contain a codeword of δ_a.

Since the path proceeds by single bit flips, there must exist a critical intermediate state P^{i^*}, which represents the first moment when a logical operator of either δ_a or δ_b appears. More precisely, P^{i^*} could be one of the following configurations:

(1) Exactly one row $i \in \mathrm{supp}(L_a)$ contains a non-trivial logical operator L_b' of code δ_b (not necessarily to be L_b), while other rows and columns contain none, or
(2) Exactly one column $j \in \mathrm{supp}(L_b)$ contains a non-trivial logical operator L_a' of code δ_a (not necessarily to be L_a), while other rows and columns contain none, or
(3) In rare cases, exactly one row $i \in \mathrm{supp}(L_a)$ contains a non-trivial logical operator L_b' of code δ_b (not necessarily to be L_b), while simultaneously, exactly one column $j \in \mathrm{supp}(L_b)$ contains a non-trivial logical operator L_a' of code δ_a (not necessarily to be L_a), with all other rows and columns containing none.

Case 1: The row containing the logical operator has d_b flipped bits at positions $(i, j_1), \ldots, (i, j_{d_b'})$ where $\{j_1, \ldots, j_{d_b'}\} \subseteq \mathrm{supp}(L_b')$. Each of these bits lies in a distinct column. Since these

columns do not contain complete codewords of δ_a, each such incomplete column contributes at least 1 to the syndrome weight. With $d_b' \geqslant d_b$, the total syndrome weight is at least d_b.

Case 2: Similarly, the column containing the logical operator L_a' has d_a' flipped bits, each in a distinct row that does not form a complete codeword of δ_b. Thus each incomplete row contributes at least 1 to the syndrome weight, yielding a total contribution of at least d_a.

Case 3: The row containing a logical operator of δ_b contributes at least d_b flipped bits, and the column containing a logical operator of δ_a contributes at least d_a flipped bits. With out loss of generality, suppose these two logical operators intersect at position (i, j). The row induces syndrome violations in $d_b - 1$ columns (excluding the column that contains a logical operator), and the column induces violations in $d_a - 1$ rows (excluding the row that contains a logical operator). Thus, the syndrome weight is at least $(d_a-1)+(d_b-1) = d_a+d_b-2 \geq \min[d_a, d_b]$, where the final inequality holds when $d_a, d_b \geq 2$.

In all cases, provided $d_a, d_b \geq 2$ (which is standard for codes with non-trivial distance), the syndrome weight at the critical state P^{i^*} is at least $\min[d_a, d_b]$. Since the energy barrier $\Delta(L_c)$ is defined as the minimum over all paths of the maximum syndrome weight along each path, we conclude $\Delta(L_c) \geq \min[d_a, d_b]$.

The above result holds for any logical operator L_c of δ_c. Let us denote the set of logical operators of δ_c as $L(\delta_c)$, then we have

$$\Delta(\delta) = \min\{\Delta(L_c) : L_c \in L(\delta_c)\} \geq \min[d_a, d_b]. \tag{4.30}$$

□

The intuition of Lemma 4.3.2 is from the 2D repetition code, which can be interpreted as the tensor product of two 1D repetition code. The 1D repetition code has parameters $[L, 1, L, \Omega(1)]$, where the energy barrier scales as $O(1)$. In contrast, the 2D repetition code exhibits parameters $[L^2, 1, L^2, \Omega(L)]$, with an energy barrier scaling as $\Omega(L)$.

The energy barrier of the 2D repetition code can be rigorously analyzed through its tensor product structure. Consider a 2D repetition code on an $L \times L$ square lattice, where the

logical operator corresponds to flipping all bits on the lattice. For any valid path (meets local error condition) implementing this logical operator, there necessarily exists an intermediate configuration where precisely a single row or a single column of bits has been flipped (or, a single row and a single column simultaneously). At this configuration, the number of violated checks (i.e., the energy cost) is at least L, as verified by analyzing the violated checks on each column and row. Consequently, the energy barrier of the 2D repetition code is lower-bounded by L.

4.3.2.2 Upper bound - "strip" argument

To establish an upper bound, one can just consider any specific path from the identity to a logical operator. We analyze two distinct implementation strategies: row-wise and column-wise implementation of logical operators. Same to the "strip" argument in Ref. [144].

We still view the tensor product code δ_c as operating on a 2D grid of size $n_a \times n_b$. In this representation, each row corresponds to a copy of code δ_b, each column corresponds to a copy of code δ_a.

For the row-wise implementation, let $r_a = \{P_a^0, P_a^1, \ldots, P_a^F\}$ represent an optimal path for implementing logical operator L_a, where $P_a^0 = I$ and $P_a^F = L_a$. By definition, the maximum syndrome weight along the path r_a is E_a. Because $P_i P_{i+1} = E_i$ is a single Pauli for all i, we denote $q_a = \{j_1, j_2, \ldots, j_{L_a}\}$ as the position of single Pauli in each step of r_a. These positions correspond to rows on the 2D grid of the tensor product code.

We consider a path for implementing L_c by applying L_b sequentially according to the positions in q_a. Specifically, Our path r_c for implementing L_c begins with $P_c^0 = I$, for each position $j_k \in q_a$, we implement the logical operator L_b on the corresponding row. After finishing all rows in q_a, we have $P_c^F = L_c$.

Now consider the energy barrier given by this specific r_c. Given any $P_c^i \in r_c$, the energy of P_c^i is

$$\epsilon(P_c^i) = \text{wt}((\delta_a \otimes I_{n_b})P_c^i) + \text{wt}((I_{n_a} \otimes \delta_b)P_c^i). \tag{4.31}$$

The maximum syndrome weight along this path is less than $d_b E_a + E_b$. This follows because for the part $\text{wt}((\delta_a \otimes I_{n_b})P_c^i)$, each column contributes a syndrome weight of at most E_a, and there are d_b columns where L_b has support, which gives $d_b E_a$. While for the part $\text{wt}((I_{n_a} \otimes \delta_b)P_c^i)$, the maximum syndrome weight is E_b because we implement L_b sequentially.

Analogously, for column-wise implementation, let $r_b = \{P_b^0, P_b^1, \ldots, P_b^F\}$ represent an optimal path for implementing L_b, with maximum syndrome weight E_b. One can construct another path for L_c by implementing L_a on columns according to the positions provided by path r_b. The maximum syndrome weight along this path is less than $d_a E_b + E_a$.

By selecting the implementation strategy with the lower maximum syndrome weight, we establish the upper bound:

$$\Delta(L_c) \leqslant \min[d_b E_a + E_b, d_a E_b + E_a]. \tag{4.32}$$

In addition, we assert that (although we cannot provide a formal proof) there is no path with an energy barrier lower than $\min[d_b E_a, d_a E_b]$, which we formally state in the following conjecture.

CONJECTURE 4.3.1. *Given two classical codes δ_a and δ_b with parameters $[n_a, k_a, d_a, E_a]$ and $[n_b, k_b, d_b, E_b]$, the energy barrier of the tensor product code δ_c is*

$$\Delta(\delta_c) \geqslant \min[d_a E_b, E_a d_b]. \tag{4.33}$$

We conjecture this result and anticipate that an elegant, rigorous proof will be found in the future (See more discussion in the Appendix). Meanwhile, the lower bound provided in Lemma 4.3.2 is optimal with respect to the order of the distance, although the value E_a and E_b may yield a non-negligible difference in the absolute value of the energy barrier. Thus the lower bound in Lemma 4.3.2 is tight if underlying codes exhibit system size-dependent distances but constant energy barriers.

4.4 Energy barrier of 3D hypergraph product codes

With all the necessary preparations in place, we now turn our attention to the energy barrier of HHGP codes. In this section, we consider the simplest instance, 3D hypergraph product codes that are constructed from three classical codes.

We begin by reviewing the structure of 3D hypergraph product codes and analyzing their logical operators, defining a complete set as elementary canonical logical operators. We first address Z logical operators. Using the strip argument from Ref. [144], we establish an upper bound for the energy barrier. We then derive a lower bound by examining the structure of those elementary canonical Z logical operators. Based on energy barrier of tensor product codes (Lemma 4.3.2), we demonstrate that this lower bound relates to the distances of the underlying classical codes. Finally, we discuss the energy barrier of the logical X operators.

4.4.1 Code construction of 3D hypergraph product codes

Consider three classical codes δ_a, δ_b, and δ_c with respective parameters $[n_\ell, k_\ell, d_\ell]$, where $\ell \in \{a, b, c\}$, and the parameters of their transposes δ_ℓ^T are denoted by $[n_\ell^T, k_\ell^T, d_\ell^T]$. For convenience, we denote $r_\ell \equiv n_\ell^T$. Applying the 3D hypergraph product to these classical codes generates a length-4 chain complex [59]

$$C_0 \xleftarrow{\partial_0} C_1 \xleftarrow{\partial_1} C_2 \xleftarrow{\partial_2} C_3, \tag{4.34}$$

where ∂_0, ∂_1 and ∂_2 are

$$\partial_0 = \begin{pmatrix} \delta_a \otimes I_{n_b} \otimes I_{n_c} \\ I_{n_a} \otimes \delta_b \otimes I_{n_c} \\ I_{n_a} \otimes I_{n_b} \otimes \delta_c \end{pmatrix}, \tag{4.35}$$

$$\partial_1 = \begin{pmatrix} I_{r_a} \otimes \delta_b \otimes I_{n_c} & \delta_a \otimes I_{r_b} \otimes I_{n_c} & 0_{r_a r_b n_c \times n_a n_b r_c} \\ I_{r_a} \otimes I_{n_b} \otimes \delta_c & 0_{r_a n_b r_c \times n_a r_b n_c} & \delta_a \otimes I_{n_b} \otimes I_{r_c} \\ 0_{n_a r_b r_c \times r_a n_b n_c} & I_{n_a} \otimes I_{r_b} \otimes \delta_c & I_{n_a} \otimes \delta_b \otimes I_{r_c} \end{pmatrix}, \tag{4.36}$$

$$\partial_2 = \left(I_{r_a} \otimes I_{r_b} \otimes \delta_c \quad I_{r_a} \otimes \delta_b \otimes I_{r_c} \quad \delta_a \otimes I_{r_b} \otimes I_{r_c}\right). \tag{4.37}$$

Here I_n denotes an identity matrix of size $n \times n$.

The relation $\partial_1 \partial_0 = 0$ allows us to construct a quantum CSS code $\mathcal{C}(\delta_a, \delta_b, \delta_c)$ with parity check matrix $H_Z = \partial_0^T$ and $H_X = \partial_1$. The commutation relation follows as $H_X H_Z^T = \partial_1 \partial_0 = 0$.

Additionally, since $\partial_2 \partial_1 = 0$, the matrix $M = \partial_2$ serves as the meta-check for H_X. This means any valid X-syndrome must satisfy the constraints imposed by M.

The parameters of the code $\mathcal{C}(\delta_a, \delta_b, \delta_c)$ can be determined from the classical code δ_a, δ_b, and δ_c. As demonstrated in Ref. [110], the resulting quantum code $\mathcal{C}(\delta_a, \delta_b, \delta_c)$ is an $[[n, k, d_x, d_z]]$ code with

$$\begin{aligned} n &= n_a^T n_b n_c + n_a n_b^T n_c + n_a n_b n_c^T, \\ k &= k_a^T k_b k_c + k_a k_b^T k_c + k_a k_b k_c^T, \\ d_x &= \min\left[d_a^T, d_b^T, d_c^T\right], \\ d_z &= \min\left[d_b d_c, d_a d_c, d_a d_b\right]. \end{aligned} \tag{4.38}$$

4.4.2 Logical operators of 3D hypergraph product codes

The logical operators of stabilizer codes commute with all elements in the stabilizer group while remaining distinct from it. We first examine the Z-logical operators, which are operators in $\ker(H_X) \backslash \mathrm{im}(H_Z^T)$.

Denote the binary form of logical operators of classical codes δ_a, δ_b, and δ_c as $L_A = \{L_A^1, L_A^2, \ldots, L_A^{k_a}\}$, $L_B = \{L_B^1, L_B^2, \ldots, L_B^{k_b}\}$, and $L_C = \{L_C^1, L_C^2, \ldots, L_C^{k_c}\}$, respectively. We first show that the following three sets of binary vectors give a complete set of Z logical

operators.

$$L_Z^1 = \sum_{i,j,k} \alpha_{ijk} \begin{pmatrix} v_i \otimes L_B^j \otimes L_C^k \\ 0_{n_a r_b n_c} \\ 0_{n_a n_b r_c} \end{pmatrix},$$
$$L_Z^2 = \sum_{i,j,k} \beta_{ijk} \begin{pmatrix} 0_{r_a n_b n_c} \\ L_A^i \otimes w_j \otimes L_C^k \\ 0_{n_a n_b r_c} \end{pmatrix}, \tag{4.39}$$
$$L_Z^3 = \sum_{i,j,k} \gamma_{ijk} \begin{pmatrix} 0_{r_a n_b n_c} \\ 0_{n_a r_b n_c} \\ L_A^i \otimes L_B^j \otimes u_k \end{pmatrix},$$

where v_i, w_j, and u_k are unit vectors with length r_a, r_b, and r_c, respectively. 0_n is a zero vector with length n, and $\alpha_{ijk}, \beta_{ijk}, \gamma_{ijk} \in \{0, 1\}$. Given that $\delta_a L_A^i = \delta_b L_B^j = \delta_c L_C^k = 0$, one can confirm that $H_X L_Z^l = 0$ for $l \in \{1, 2, 3\}$.

For L_Z^1, the condition $H_X L_Z^1 = 0$ ensures that any X-type stabilizer from H_X commutes with L_Z^1. Similarly, any Z-type stabilizer from H_Z naturally commutes with L_Z^1 as they both consist solely of Pauli Z operators. To prove that L_Z^1 are Z logical operators, we must prove that L_Z^1 are not in the image of H_Z^T. To proceed with this analysis, we first denote:

$$L_Z^{1,i,j,k} = \begin{pmatrix} v_i \otimes L_B^j \otimes L_C^k \\ 0_{n_a r_b n_c} \\ 0_{n_a n_b r_c} \end{pmatrix}. \tag{4.40}$$

For $a_i \in \{0, 1\}$, suppose that the linear combination of these operators

$$\sum_i a_i L_Z^{1,i,j,k} = \begin{pmatrix} (\sum_i a_i v_i) \otimes L_B^j \otimes L_C^k \\ 0_{n_a r_b n_c} \\ 0_{n_a n_b r_c} \end{pmatrix} \tag{4.41}$$

is in the image of matrix H_Z^T. Then, there exists a vector x such that

$$\begin{aligned}\sum_i a_i L_Z^{1,i,j,k} &= \begin{pmatrix} (\sum_i a_i v_i) \otimes L_B^j \otimes L_C^k \\ 0_{n_a r_b n_c} \\ 0_{n_a n_b r_c} \end{pmatrix} \\ &= H_Z^T (x \otimes L_B^j \otimes L_C^k) \\ &= \begin{pmatrix} (\delta_a \otimes I_{n_b} \otimes I_{n_c})(x \otimes L_B^j \otimes L_C^k) \\ (I_{n_a} \otimes \delta_b \otimes I_{n_c})(x \otimes L_B^j \otimes L_C^k) \\ (I_{n_a} \otimes I_{n_b} \otimes \delta_c)(x \otimes L_B^j \otimes L_C^k) \end{pmatrix} \\ &= \begin{pmatrix} \delta_a x \otimes L_B^j \otimes L_C^k \\ 0_{n_a r_b n_c} \\ 0_{n_a n_b r_c} \end{pmatrix}. \end{aligned} \tag{4.42}$$

Recall the number of encoded bits for δ_a^T is k_a^T, then $\dim(\text{im}(\delta_a)) = n_a - k_a^T$, it is feasible to select a minimum of k_a^T unit vectors v_i, ensuring that any combination of these v_i remains external to the image of δ_a. Specifically, for these k_a^T unit vectors v_i, there does not exist any vector x such that

$$\sum_{i=1}^{k_a^T} a_i v_i = \delta_a x. \tag{4.43}$$

For fixed L_B^j and L_C^k, these k_a^T unit vectors generate k_a^T operators $L_Z^{1,i,j,k}$ that belong to $\ker(H_X) \backslash \text{im}(H_Z^T)$. Since δ_b has k_b logical bits and δ_c has k_c logical bits, there are k_b distinct L_B and k_c distinct L_C operators. This yields $k_a^T k_b k_c$ unique instances of $L_Z^{1,i,j,k}$, which function as logical Z operators of the quantum code.

Applying the same logic to operators L_Z^2 and L_Z^3 defined in Eq. (4.39), we can demonstrate that there exist $k_a k_b^T k_c$ distinct $L_Z^{2,i,j,k}$ and $k_a k_b k_c^T$ distinct $L_Z^{3,i,j,k}$ that serve as logical Z operators of the quantum code.

This completely characterizes all $k_a^T k_b k_c + k_a k_b^T k_c + k_a k_b k_c^T$ logical Z operators of the 3D hypergraph product codes. We refer to these as *elementary canonical* logical Z operators.

We now examine the logical X operators, which are in the set $\ker(H_Z)\backslash\text{im}(H_X^T)$. Consider the following three types of Pauli X operators,

$$L_X^1 = \sum_{i,j,k} \alpha'_{ijk} \begin{pmatrix} L_A^{i'} \otimes w_j \otimes u_k \\ 0_{n_a r_b n_c} \\ 0_{n_a n_b r_c} \end{pmatrix}, \tag{4.44}$$

$$L_X^2 = \sum_{i,j,k} \beta'_{ijk} \begin{pmatrix} 0_{r_a n_b n_c} \\ v_i \otimes L_B^{j'} \otimes u_k \\ 0_{n_a n_b n_c} \end{pmatrix}, \tag{4.45}$$

$$L_X^3 = \sum_{i,j,k} \gamma'_{ijk} \begin{pmatrix} 0_{r_a n_b n_c} \\ 0_{n_a r_b n_c} \\ v_i \otimes w_j \otimes L_C^{k'} \end{pmatrix}. \tag{4.46}$$

$$\tag{4.47}$$

Here v_i, w_j, and u_k are unit vectors of length n_a, n_b and n_c respectively. Moreover, $L_A^{i'}, L_B^{j'}$ and $L_C^{k'}$ are logical operators of the code defined by the parity check matrix δ_a^T, δ_b^T, and δ_c^T respectively.

For L_X^1, $H_Z L_X^1 = 0$ implies that any Z-type stabilizer from H_Z commutes with L_X^1. Meanwhile, any X-type stabilizer in H_X also commutes with L_X^1 as they consist only of Pauli X operators. To identify logical X operators, we need to find Pauli X operators in the set L_X^1 that are not present in the image of H_X^T. We denote:

$$L_X^{1,i,j,k} = \begin{pmatrix} L_A^{i'} \otimes w_j \otimes u_k \\ 0_{n_a r_b n_c} \\ 0_{n_a n_b r_c} \end{pmatrix}. \tag{4.48}$$

For $b_j, c_k \in \{0, 1\}$, suppose that the linear combination of these operators

$$\sum_{jk} b_j c_k L_X^{1,i,j,k} = \begin{pmatrix} L_A^{i'} \otimes \sum_j b_j w_j \otimes \sum_k c_k u_k \\ 0_{n_a r_b n_c} \\ 0_{n_a n_b r_c} \end{pmatrix} \tag{4.49}$$

is in the image of matrix H_X^T. Then, there exists vectors y, z such that

$$\begin{aligned}\sum_{jk} b_j c_k L_X^{1,i,j,k} &= \begin{pmatrix} L_A^{i'} \otimes \sum_j b_j w_j \otimes \sum_k c_k u_k \\ 0_{n_a r_b n_c} \\ 0_{n_a n_b r_c} \end{pmatrix} \\ &= H_X^T (L_A^{i'} \otimes y \otimes z) \end{aligned} \tag{4.50}$$

Recall that k_b^T and k_c^T represent the number of encoded bits for δ_b^T and δ_c^T respectively. Then with $\dim(\text{im}(\delta_b^T)) = n_b - k_b$ and $\dim(\text{im}(\delta_c^T)) = n_c - k_c$, one can select at least k_b unit vectors w_j and k_c unit vectors u_k such that any combination of these vectors remains outside the image of δ_b^T and δ_c^T respectively.

Given that there are k_a^T different $L_A^{i'}$, and paralleling our analysis of logical Z operators, we can verify that there exist $k_a^T k_b k_c$ distinct $L_X^{1,i,j,k}$, $k_a k_b^T k_c$ distinct $L_X^{2,i,j,k}$, and $k_a k_b k_c^T$ distinct $L_X^{3,i,j,k}$ that serve as logical X operators in $\ker(H_Z) \backslash \text{im}(H_X^T)$, we call those logical opertors as elementary canonical logical X operators. This accounts for all $k_a^T k_b k_c + k_a k_b^T k_c + k_a k_b k_c^T$ logical X operators of the code.

4.4.3 Energy barrier of Z logical operators

We now analyze the energy barrier for logical Z operators in 3D hypergraph product codes. First, we employ a strip argument to establish an upper bound. Then, using Lemma 4.3.2, we derive a lower bound for the energy barrier for logical Z operators.

4.4.3.1 Upper bound: "strip" argument

The upper bound on the energy barrier can be obtained by considering a specific path from the identity to any logical operator. By choosing this logical operator to be an elementary canonical logical operator, and considering the "strip" argument mentioned in Sec. 4.3.2, one can obtain an upper bound.

For elementary canonical logical operator in the form of $L_Z^{1,i,j,k}$, with fixed i, j, and k, the logical operator takes the form as in the Eq. (4.40). Since v_i is a unit vector, the effective logical operator simplifies to $L_Z^{1',j,k} = L_B^j \otimes L_C^k$. This can be interpreted exactly as a logical operator of a tensor product code. One can construct the same kind of strip configuration when implementing these logical operators. Thus, the energy barrier of this logical operator is upper bounded by the strip argument as $\Delta(L_Z^{1',j,k}) \leqslant \min[E_b d_c, E_c d_b]$, where E_b and E_c are the energy barriers of the logical operators L_B^j and L_C^k, respectively, and d_b and d_c are their corresponding distances.

With the same argument, the energy barrier of the canonical logical Z operator $L_Z^{2,i,j,k}$ is upper bounded by $\min[E_a d_c, E_c d_a]$, and the energy barrier of $L_Z^{3,i,j,k}$ is upper bounded by $\min[E_a d_b, E_b d_a]$.

In summary, consider a 3D hypergraph product code $\mathcal{C}$ constructed from classical codes δ_a, δ_b, and δ_c with parameters $[n_\ell, k_\ell, d_\ell, E_\ell]$, where $\ell \in \{a, b, c\}$. Based on the structure of the canonical logical operators we have derived, the energy barrier of the logical Z operators can be shown to have the following upper bound.

$$\Delta(L_Z) \leqslant \min\{d_\ell E_m : \ell, m \in \{a, b, c\}, \ell \neq m\}. \tag{4.51}$$

4.4.3.2 Lower bound

We now establish a lower bound for the energy barrier of Z-type logical operators by analyzing the structure of these canonical logical operators. Consider a 3D hypergraph product code $\mathcal{C}$ constructed from classical codes δ_a, δ_b, and δ_c with parameters $[n_\ell, k_\ell, d_\ell, E_\ell]$, where $\ell \in \{a, b, c\}$. The parity check matrices of this quantum code are $H_Z = \partial_0^T$ and $H_X = \partial_1$, as defined in Eq. (4.35) and Eq. (4.36). Recall that the Z-type canonical logical operators L_Z

fall into three categories.

$$\begin{aligned} L_Z^1 &= \sum_{i,j,k} \alpha_{ijk}((v_i \otimes L_B^j \otimes L_C^k)^T, 0^T, 0^T)^T, \\ L_Z^2 &= \sum_{i,j,k} \beta_{ijk}(0^T, (L_A^i \otimes w_j \otimes L_C^k)^T, 0^T)^T, \\ L_Z^3 &= \sum_{i,j,k} \gamma_{ijk}(0^T, 0^T, (L_A^i \otimes L_B^j \otimes u_k)^T)^T. \end{aligned} \tag{4.52}$$

where (i) $\delta_a L_A^i = \delta_b L_B^j = \delta_c L_C^k = 0$ and (ii) $v_i \notin \mathrm{Im}\,(\delta_a)$, $w_j \notin \mathrm{Im}\,(\delta_b)$ and $u_k \notin \mathrm{Im}\,(\delta_c)$ are unit vectors. We note that $i \in \{1, \ldots, k_a^T\}$, $j \in \{1, \ldots, k_b^T\}$ and $k \in \{1, \ldots, k_c^T\}$ and that the set of logical operators expressible in this form is complete. A canonical logical operator is elementary if only one of the coefficients, α_{ijk} (also β_{ijk} or γ_{ijk}), is equal to one.

The Tanner graph representation provides an intuitive framework that is well-suited for our analysis. Let $\mathcal{G}_1(V_1, C_1)$, $\mathcal{G}_2(V_2, C_2)$ and $\mathcal{G}_3(V_3, C_3)$ be the Tanner graphs of codes defined by δ_a, δ_b and δ_c, respectively. Here, V_1, V_2 and V_3 represent the set of bits, and C_1, C_2 and C_3 denote the set of checks. We use $\{v_1^i : i \in \{1, 2, \cdots n_a\}\}$ (resp. $\{v_2^j : j \in \{1, 2, \cdots n_b\}\}$, $\{v_3^k : j \in \{1, 2, \cdots n_c\}\}$) to refer to bit vertices in V_1 (resp. V_2, V_3), and also as length n_a (resp. n_b, n_c) unit vectors with the ith (resp. jth, kth) entry as 1.

The 3D hypergraph product $\mathcal{G}_1 \times \mathcal{G}_2 \times \mathcal{G}_3$ is a bipartite graph with vertex set $V \cup C$, where

$$V = V_1 \otimes V_2 \otimes C_3 \cup V_1 \otimes C_2 \otimes V_3 \cup C_1 \otimes V_2 \otimes V_3 \tag{4.53}$$

is the qubit set, and

$$\begin{aligned} C \;=\; & V_1 \otimes V_2 \otimes V_3 \cup C_1 \otimes C_2 \otimes V_3 \cup C_1 \otimes V_2 \otimes C_3 \\ & \cup V_1 \otimes C_2 \otimes C_3 \cup C_1 \otimes C_2 \otimes C_3 \end{aligned} \tag{4.54}$$

is the check and meta-check set. More specifically, $V_1 \otimes V_2 \otimes V_3$ is the set the Z type checks, $C_1 \otimes C_2 \otimes V_3 \cup C_1 \otimes V_2 \otimes C_3 \cup V_1 \otimes C_2 \otimes C_3$ represents the set of X checks, and $C_1 \otimes C_2 \otimes C_3$ is meta-checks on X checks.

The set of qubits can be partitioned into three subsets, $V_1 \otimes V_2 \otimes C_3$, $V_1 \otimes C_2 \otimes V_3$, and $C_1 \otimes V_2 \otimes V_3$. For $H_X = \partial_1$, there are three column blocks (see Eq. (4.36)). The first column

block of H_X acts on $C_1 \otimes V_2 \otimes V_3$, the second column block of H_X acts on $V_1 \otimes C_2 \otimes V_3$, and the third column block acts on $V_1 \otimes V_2 \otimes C_3$. Moreover, the subset $C_1 \otimes V_2 \otimes V_3$ can be further partitioned into $n_a^T \equiv r_a$ subsets $\{c_1^1 \otimes V_2 \otimes V_3, c_1^2 \otimes V_2 \otimes V_3, \cdots, c_1^{r_a} \otimes V_2 \otimes V_3\}$, where $c_1^k \otimes V_2 \otimes V_3 := \{c_1^k \otimes y \otimes z : y \in V_2, z \in V_3\}$ and $V_2 = \{v_2^1, \ldots, v_2^{n_b}\}$, $V_3 = \{v_3^1, \ldots, v_3^{n_c}\}$.

Thus, a Z-type Pauli operator can be expressed as a bit-string $z = \left({z^{(1)}}^T, {z^{(2)}}^T, {z^{(3)}}^T\right)^T$, where $z^{(1)}$ is supported on the qubit subset $C_1 \otimes V_2 \otimes V_3$ with vector space $\mathbb{F}_2^{r_a} \otimes \mathbb{F}_2^{n_b} \otimes \mathbb{F}_2^{n_c}$, $z^{(2)}$ is supported on the qubit set $V_1 \otimes C_2 \otimes V_3$ with vector space $\mathbb{F}_2^{n_a} \otimes \mathbb{F}_2^{r_b} \otimes \mathbb{F}_2^{n_c}$ and $z^{(3)}$ is supported on the qubit set $V_1 \otimes V_2 \otimes C_3$ with vector space $\mathbb{F}_2^{n_a} \otimes \mathbb{F}_2^{n_b} \otimes \mathbb{F}_2^{r_c}$. Because $z^{(1)}$, $z^{(2)}$, and $z^{(3)}$ act as a tensor product of three vector spaces, one can view them also as 3D matrices. (For instance, $v_i \otimes u_j \otimes w_k$ for unit vectors v_i, u_j and w_k can be viewed as a 3D matrix whose entry is 1 on the i'th a-slice, the j'th b-slice and the k'th c-slice and zero elsewhere. We call this procedure vector reshaping. After reshaping, $z^{(1)}$, $z^{(2)}$ and $z^{(3)}$ become $Z^{(1)}$, $Z^{(2)}$ and $Z^{(3)}$ respectively. Here, $Z^{(1)}$ is an $r_a \times n_b \times n_c$ matrix, and the entry $Z^{(1)}_{i,j,k}$ is supported on the qubit $r_1^i \otimes v_2^j \otimes v_3^k$. Moreover, the ith a-slice $Z_i^{(1)}$ is supported on qubits $\{r_1^i \otimes v_2^1 \otimes v_3^1, r_1^i \otimes v_2^2 \otimes v_3^1, \cdots, r_1^i \otimes v_2^{n_b} \otimes v_3^{n_c}\}$, which we refer to as $r_1^i \otimes V_2 \otimes V_3$. Similarly, the subset $V_1 \otimes C_2 \otimes V_3$ can be partitioned into r_b b-slices, and jth b-slice is supported on the qubit subset $V_1 \otimes r_2^j \otimes V_3$. Also, the subset $V_1 \otimes V_2 \otimes C_3$ can be partitioned into r_c c-slices, and kth c-slice is supported on the qubit subset $V_1 \otimes V_2 \otimes r_3^k$ (defined similarly).

An elementary canonical logical-Z operator, for example, in the form $(v_i \otimes L_B^j \otimes L_C^k, 0, 0)^T$, is supported on the subset $r_1^i \otimes V_2 \otimes V_3$ if $v_i = r_1^i$. In matrix form, it is supported on the ith a-slice of $Z^{(1)}$. Fig. 4.2 provides an illustrative example of the 3D Toric code structure.

We now prove a lower bound for the energy barrier of the canonical logical-Z operators, as shown in the following proposition.

PROPOSITION 4.4.1. *For any nontrivial canonical logical-Z operator L,*

$$\Delta(L) \geq \min[d_a, d_b, d_c] \tag{4.55}$$

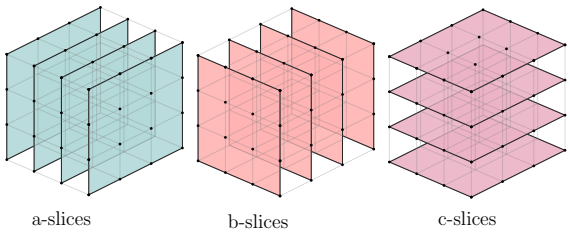


FIGURE 4.2: The 3D toric code can be structured as a cubic lattice (with periodic boundary conditions). From left to right, the figure illustrates its decomposition into a-slices, b-slices, and c-slices. The 3D Toric code encodes 3 logical qubits, with corresponding 3 logical Z operators. Each Z elementary canonical logical operator is supported on a single slice, with slices of the same color representing logical operators equivalent up to stabilizers. Generally, given any 3D hypergraph product code, one can create such a decomposition, although the structure within each slice will be more complicated and depends on the underlying classical codes.

We introduce two lemmas to prove this proposition. We use the following convention in the proof. For the subsequent proof, we adopt the following notation: a path $r = \{P_0, P_1, \ldots, P_\ell\}$ is said to be *supported on* a subset $U \subseteq V$ if $\mathrm{supp}(P_i) \subseteq U$ for all $0 \leq i \leq \ell$.

LEMMA 4.4.1. *For any elementary canonical logical-Z operator L supported on $r_1^\alpha \otimes V_2 \otimes V_3$ (resp. $V_1 \otimes r_2^\beta \otimes V_3$, $V_1 \otimes V_2 \otimes r_3^\gamma$), its energy barrier is attained by a path supported on $r_1^\alpha \otimes V_2 \otimes V_3$ (resp. $V_1 \otimes r_2^\beta \otimes V_3$, $V_1 \otimes V_2 \otimes r_3^\gamma$).*

LEMMA 4.4.2. *For any nontrivial canonical logical-Z operator L, the energy barrier $\Delta(L)$ is greater than or equal to the minimum energy barrier of the elementary canonical logical-Z operators.*

Consider an elementary canonical logical-Z operator L supported on $r_1^\alpha \otimes V_2 \otimes V_3$. Suppose $\Delta(L)$ is given by a path r. The main idea behind the proof of Lemma 4.4.1 is to deform a general path r into a new path r', supported only on $r_1^\alpha \otimes V_2 \otimes V_3$, and the energy barrier of r' is not greater than that of the original path r. A similar argument can be applied to prove Lemma 4.4.2. The proofs closely extended from the arguments in the hypergraph product case [109]. The details are provided in the appendix.

For the elementary canonical logical operator $L_Z^{1,\alpha} = ((r_1^\alpha \otimes L_B^j \otimes L_C^k)^T, 0^T, 0^T)^T$, with fixed α, j, and k. $L_Z^{1,\alpha}$ is supported on $r_1^\alpha \otimes V_2 \otimes V_3$. To compute the energy barrier of $L_Z^{1,\alpha}$, consider

a path $r = \{P_0, P_1, \cdots, P_F\}$. Then

$$\Delta(L_Z^{1,\alpha}) = \min\{\epsilon_{\max}(r) : r \in w(0, L_Z^{1,\alpha})\}. \tag{4.56}$$

Here, $P_0 = (0,0,0)^T$ is a zero vector, and $P_F = ((r_1^\alpha \otimes L_B^j \otimes L_C^k)^T, 0^T, 0^T)^T$. Note that r is a path that walks throughout the set of qubits. Therefore, $P_\ell \in r$ could be represented as $P_\ell = ((P_\ell^1)^T, (P_\ell^2)^T, (P_\ell^3)^T)^T$, where each component, P_ℓ^1, P_ℓ^2, and P_ℓ^3, could be nontrivial.

Lemma 4.4.1 implies that there is a path r supported on $r_1^\alpha \otimes V_2 \otimes V_3$ that attains the energy barrier of $\Delta(L)$. A path restricted to the first component means that for any $P_\ell \in r$, one can set $P_\ell^2 = P_\ell^3 = 0$, the only nontrivial Paulis (nonzero binaries) are in P_ℓ^1 with fixed r_1^α. Furthermore, r_1^α is fixed means every P_ℓ^1 has the form $r_1^\alpha \otimes y \otimes z$, where $y \in V_2$ and $z \in V_3$ are vectors of length n_b and n_c respectively.

Given a $P_\ell \in r$, the energy (the number of violated checks) of P_ℓ is given by $\epsilon(P_\ell) = \mathrm{wt}\,(H_X P_\ell)$, has the form

$$\mathrm{wt}\left(\begin{pmatrix} I_{n_a}\otimes\delta_b\otimes I_{n_c} & \delta_a\otimes I_{n_b}\otimes I_{n_c} & 0 \\ I_{n_a}\otimes I_{n_b}\otimes\delta_c & 0 & \delta_a\otimes I_{n_b}\otimes I_{n_c} \\ 0 & I_{n_a}\otimes I_{n_b}\otimes\delta_c & I_{n_a}\otimes\delta_b\otimes I_{n_c} \end{pmatrix} \begin{pmatrix} P_\ell^1 \\ P_\ell^2 \\ P_\ell^3 \end{pmatrix}\right). \tag{4.57}$$

By setting $P_\ell^2 = P_\ell^3 = 0$, then

$$\begin{aligned} \epsilon(P_\ell) &= \mathrm{wt}\left(\begin{pmatrix} I_{n_a}\otimes\delta_b\otimes I_{n_c} \\ I_{n_a}\otimes I_{n_b}\otimes\delta_c \end{pmatrix} P_\ell^1\right) \\ &= \mathrm{wt}\left(I_{n_a}\otimes\delta_b\otimes I_{n_c} P_\ell^1\right) + \mathrm{wt}\left(I_{n_a}\otimes I_{n_b}\otimes\delta_c P_\ell^1\right). \end{aligned} \tag{4.58}$$

Furthermore, for fixed r_1^α, because r_1^α is a unit vector, the part r_1^α can be discarded by applying the map $(r_1^\alpha)^T \otimes I_{n_b} \otimes I_{n_c}$ to P_ℓ^1. Thus

$$\epsilon(P_\ell) = \mathrm{wt}\left(\delta_b \otimes I_{n_c} P_\ell^{1'}\right) + \mathrm{wt}\left(I_{n_b} \otimes \delta_c P_\ell^{1'}\right), \tag{4.59}$$

with $P_\ell^{1'} = ((r_1^\alpha)^T \otimes I_{n_b} \otimes I_{n_c}) P_\ell^1$.

In summary, to calculate the energy barrier of the Z-type logical operator $L_Z^{1,\alpha,j,k} = ((r_1^\alpha \otimes L_B^j \otimes L_C^k)^T, 0^T, 0^T)^T$, one can consider a path $r \in \{Q_0, Q_1, \cdots, Q_L\}$ supported on qubit subset $r_1^\alpha \otimes V_2 \otimes V_3$, then

$$\Delta(L_Z^{1,\alpha,j,k}) = \min\{\epsilon_{\max}(r) : r \in w(0, L_B^j \otimes L_C^k)\}. \tag{4.60}$$

where for each r,

$$\epsilon_{\max}(r) = \max[\epsilon(Q_0), \epsilon(Q_1), \cdots, \epsilon(Q_L)], \tag{4.61}$$

and the energy cost of the step Q_ℓ is

$$\epsilon(Q_\ell) = \mathrm{wt}\left(\delta_b \otimes I_{n_c} Q_\ell\right) + \mathrm{wt}\left(I_{n_b} \otimes \delta_c Q_\ell\right). \tag{4.62}$$

This scenario corresponds exactly to computing the energy barrier of a classical tensor product code (Eq. (4.25)), where the underlying classical codes are δ_b and δ_c. The subset $r_1^\alpha \otimes V_2 \otimes V_3$ is the bit set of this classical tensor product code. The path for applying logical operator $L_Z^{1,\alpha,j,k}$ effectively forms a path for the logical operator $L_B \otimes L_C$ in the tensor product code constructed by δ_b and δ_c.

According to Lemma 4.3.2, the energy barrier of $L_B \otimes L_C$ is lower bounded by $\min[d_b, d_c]$. Thus the energy barrier of the Z-type logical operator $L_Z^{1,\alpha,j,k}$ satisfies $\Delta\left(L_Z^{1,\alpha,j,k}\right) \geq \min[d_b, d_c]$.

Following the same reasoning, the energy barrier for logical Z-type operators of the form $L_Z^{2,i,j,k} = \left(0^T, (L_A^i \otimes w_j \otimes L_C^k)^T, 0^T\right)^T$ is lower bounded by $\min[d_a, d_c]$, and the logical operators of the form $L_Z^{3,i,j,k} = \left(0^T, 0^T, (L_A^i \otimes L_B^j \otimes u_k)^T\right)^T$ have an energy barrier lower bounded by $\min[d_a, d_b]$.

In summary, for any elementary canonical Z logical operator L in the 3D hypergraph product code, the energy barrier is lower bounded by

$$\Delta(L) \geqslant \min[d_a, d_b, d_c]. \tag{4.63}$$

With Lemma 4.4.2, we can establish that for any canonical Z-type logical operator (not just elementary ones), the energy barrier is lower bounded by $\min[d_a, d_b, d_c]$. Moreover, results from Ref. [109] demonstrate that for any logical operator L and stabilizer S with $\Delta(L) \geqslant \Delta(S)$, we have $\Delta(L) = \Delta(LS)$. Therefore, with the LDPC property yielding $\Delta(S) = O(1)$ for any S, and in cases where $d_a, d_b, d_c \geqslant O(1)$, this lower bound applies to all logical Z operators, not just the canonical ones.

Formally, we have the following theorem.

THEOREM 4.4.1. *Let $\Delta(L_Z)$ be the energy barrier of the Z logical operators of the 3D hypergraph product code constructed from classical codes δ_a, δ_b and δ_c with parameters $[n_\ell, k_\ell, d_\ell, E_\ell]$ for $\ell \in \{a, b, c\}$. If the resulting quantum code is LDPC with sparsity parameters w_c, w_q, then in cases where $d_a, d_b, d_c \geqslant w_c w_q$, we have*

$$\Delta(L_Z) \geqslant \min[d_a, d_b, d_c]. \tag{4.64}$$

Similarly, with Lemma 4.3.1, we can conclude that the energy barrier of 3D LDPC hypergraph product codes is lower bounded by the energy barriers of the underlying classical codes:

$$\Delta(L_Z) \geqslant \min[E_a, E_b, E_c]. \tag{4.65}$$

Moreover, if Conjecture 4.3.1 is true, we have

$$\Delta(L_Z) \geqslant \min\{d_\ell E_m : \ell, m \in \{a, b, c\}, \ell \neq m\}. \tag{4.66}$$

This lower bound matches the upper bound obtained using the strip argument, effectively providing a tight bound.

4.4.4 Energy barrier of X logical operators

The canonical X logical operators have the form

$$\begin{aligned} L_X^1 &= \textstyle\sum_{i,j,k} \alpha'_{ijk}((L_A^{i'} \otimes w_j \otimes u_k)^T, 0^T_{n_a r_b n_c}, 0^T_{n_a n_b r_c})^T, \\ L_X^2 &= \textstyle\sum_{i,j,k} \beta'_{ijk}(0^T_{r_a n_b n_c}, (v_i \otimes L_B^{j'} \otimes u_k)^T, 0^T_{n_a n_b n_c})^T, \\ L_X^3 &= \textstyle\sum_{i,j,k} \gamma'_{ijk}(0^T_{r_a n_b n_c}, 0^T_{n_a r_b n_c}, (v_i \otimes w_j \otimes L_C^{k'})^T)^T, \end{aligned} \tag{4.67}$$

where (i) $\delta_a^T L_A^{i'} = \delta_b^T L_B^{j'} = \delta_c^T L_C^{k'} = 0$ and (ii) $v_i \notin \mathrm{Im}\left(\delta_a^T\right)$, $w_j \notin \mathrm{Im}\left(\delta_b^T\right)$ and $u_k \notin \mathrm{Im}\left(\delta_c^T\right)$ are unit vectors. We note that $i \in \{1, \ldots, k_a^T\}$, $j \in \{1, \ldots, k_b\}$ and $k \in \{1, \ldots, k_c\}$ and that the set of logical operators expressible in this form is complete. A canonical logical operator is elementary if only one of the coefficients, α'_{ijk}, β'_{ijk} or γ'_{ijk}, is equal to one.

These elementary canonical logical X operators take the form of tensor products between one classical logical operator and two unit vectors, which resembles that of logical operators in hypergraph product codes, though in a higher dimension. By extending the reasoning previously applied to hypergraph product codes [109] to this higher-dimensional case, in the LDPC regime, one can establish a lower bound for the energy barrier of logical X operators as

$$\Delta(L_X) \geqslant \min\left[E_a^T, E_b^T, E_c^T\right]. \tag{4.68}$$

The upper bound can be also derived from the structure of these elementary canonical logical X operators [109], which is also $\min[E_a^T, E_b^T, E_c^T]$. In the case where $\min\left[E_a^T, E_b^T, E_c^T\right] \geqslant w_c w_q$, with w_c, w_q are the sparsity parameters, we have

$$\Delta(L_X) = \min\left[E_a^T, E_b^T, E_c^T\right]. \tag{4.69}$$

4.5 Energy barrier of 4D hypergraph product codes

In 3D hypergraph product codes, the energy barrier of Z-type logical operators is lower bounded by the distance of the underlying classical codes. While X logical operators retain

the energy barrier of these classical codes. When classical codes possess large distances but small energy barriers, only Z-type logical operators have improved energy barrier, potentially enabling single-shot. The X-type logical operators, lacking this enhanced protection, remain vulnerable. In this section, we show that 4D hypergraph product codes offer improved energy barrier for both X and Z logical operators.

The distinction between the 3D and 4D cases stems from structural differences in their elementary canonical logical operators. In 3D codes, Z logical operators take the form $L_Z = L_{c_1} \otimes L_{c_2} \otimes u_k$, while X logical operators are simply $L_X = v_i \otimes w_j \otimes L_{c_3}^T$, where v_i, w_j, and u_k are unit vectors, and $L_{c_1}, L_{c_2}, L_{c_3}$ are classical logical operators. In contrast, 4D codes exhibit a more balanced structure where both X and Z logical operators are tensor products of two classical logical operators (and two unit vectors, see Eq. 4.77 and Eq. 4.78), resulting in enhanced energy barriers for both types of logical operators.

Given four classical codes represented by parity check matrices $\delta_a, \delta_b, \delta_c, \delta_d$, let $[n_\ell, k_\ell, d_\ell, E_\ell]$ and $\left[n_\ell^T, k_\ell^T, d_\ell^T, E_\ell^T\right]$ represent the parameters of the classical linear code defined by the parity check matrix δ_ℓ and its transpose δ_ℓ^T. For convenience, we denote $r_\ell \equiv n_\ell^T$, where $\ell = \{a, b, c, d\}$. The 4D hypergraph product is defined by a length-5 chain complex

$$C_0 \xleftarrow{\partial_0} C_1 \xleftarrow{\partial_1} C_2 \xleftarrow{\partial_2} C_3 \xleftarrow{\partial_3} C_4, \tag{4.70}$$

Boundary maps $\partial_0, \partial_1, \partial_2$ and ∂_3 are

$$\partial_0 = \begin{pmatrix} \delta_a \otimes I_{n_b} \otimes I_{n_c} \otimes I_{n_d} \\ I_{n_a} \otimes \delta_b \otimes I_{n_c} \otimes I_{n_d} \\ I_{n_a} \otimes I_{n_b} \otimes \delta_c \otimes I_{n_d} \\ I_{n_a} \otimes I_{n_b} \otimes I_{n_c} \otimes \delta_d \end{pmatrix}, \tag{4.71}$$

$$\partial_1 = \begin{pmatrix} I_{r_a} \otimes \delta_b \otimes I_{n_c} \otimes I_{n_d} & \delta_a \otimes I_{r_b} \otimes I_{n_c} \otimes I_{n_d} & 0 & 0 \\ I_{r_a} \otimes I_{n_b} \otimes \delta_c \otimes I_{n_d} & 0 & \delta_a \otimes I_{n_b} \otimes I_{r_c} \otimes I_{n_d} & 0 \\ I_{r_a} \otimes I_{n_b} \otimes I_{n_c} \otimes \delta_d & 0 & 0 & \delta_a \otimes I_{n_b} \otimes I_{n_c} \otimes I_{r_d} \\ 0 & I_{n_a} \otimes I_{r_b} \otimes \delta_c \otimes I_{n_d} & I_{n_a} \otimes \delta_b \otimes I_{r_c} \otimes I_{n_d} & 0 \\ 0 & I_{n_a} \otimes I_{r_b} \otimes I_{n_c} \otimes \delta_d & 0 & I_{n_a} \otimes \delta_b \otimes I_{n_c} \otimes I_{r_d} \\ 0 & 0 & I_{n_a} \otimes I_{n_b} \otimes I_{r_c} \otimes \delta_d & I_{n_a} \otimes I_{n_b} \otimes \delta_c \otimes I_{r_d} \end{pmatrix}, \tag{4.72}$$

$$\partial_2 = \begin{pmatrix} I_{r_a} \otimes I_{r_b} \otimes \delta_c \otimes I_{n_d} & I_{r_a} \otimes \delta_b \otimes I_{r_c} \otimes I_{n_d} & 0 & \delta_a \otimes I_{r_b} \otimes I_{r_c} \otimes I_{n_d} & 0 & 0 \\ I_{r_a} \otimes I_{r_b} \otimes I_{n_c} \otimes \delta_d & 0 & I_{r_a} \otimes \delta_b \otimes I_{n_c} \otimes I_{r_d} & 0 & \delta_a \otimes I_{r_b} \otimes I_{n_c} \otimes I_{r_d} & 0 \\ 0 & I_{r_a} \otimes I_{n_b} \otimes I_{r_c} \otimes \delta_d & I_{r_a} \otimes I_{n_b} \otimes \delta_c \otimes I_{r_d} & 0 & 0 & \delta_a \otimes I_{n_b} \otimes I_{r_c} \otimes I_{r_d} \\ 0 & 0 & 0 & I_{n_a} \otimes I_{r_b} \otimes I_{r_c} \otimes \delta_d & I_{n_a} \otimes I_{r_b} \otimes \delta_c \otimes I_{r_d} & I_{n_a} \otimes \delta_b \otimes I_{r_c} \otimes I_{r_d} \end{pmatrix}, \tag{4.73}$$

$$\partial_3 = \begin{pmatrix} I_{r_a} \otimes I_{r_b} \otimes I_{r_c} \otimes \delta_d & I_{r_a} \otimes I_{r_b} \otimes \delta_c \otimes I_{r_d} & I_{r_a} \otimes \delta_b \otimes I_{r_c} \otimes I_{r_d} & \delta_a \otimes I_{r_b} \otimes I_{r_c} \otimes I_{r_d} \end{pmatrix}. \tag{4.74}$$

One can confirm that $\partial_i \partial_{i-1} = 0 \mod 2$ for $i \in 1, 2, 3$. In this case, $H_Z = \partial_1^T, H_X = \partial_2$ is used to construct a quantum CSS code $H_{(\delta_a, \delta_b, \delta_c, \delta_d)}$ with parity check matrix

$$H_{(\delta_a, \delta_b, \delta_c, \delta_d)} = \begin{pmatrix} H_X & 0 \\ 0 & H_Z \end{pmatrix}. \tag{4.75}$$

Moreover, as $\partial_3 \partial_2 = \partial_1 \partial_0 = 0$, the remaining matrices $M_1 = \partial_3$, $M_2 = \partial_0^T$ work as a meta-check for H_X and H_Z, respectively. This means that any valid X-syndrome adheres to the constraints defined by M_1, and any valid Z-syndrome satisfies the constraints defined by M_2.

The parameters of the resulting quantum code $\mathcal{C}\left(\delta_a, \delta_b, \delta_c, \delta_d\right)$ depend on the parameters of classical codes δ_a, δ_b, δ_c, and δ_d, where

$$\begin{aligned} n &= n_a^T n_b^T n_c n_d + n_a^T n_b n_c^T n_d + n_a^T n_b n_c n_d^T + n_a n_b^T n_c^T n_d + n_a n_b^T n_c n_d^T + n_a n_b n_c^T n_d^T, \\ k &= k_a^T k_b^T k_c k_d + k_a^T k_b k_c^T k_d + k_a^T k_b k_c k_d^T + k_a k_b^T k_c^T k_d + k_a k_b^T k_c k_d^T + k_a k_b k_c^T k_d^T, \\ d_x &= \min\left[d_a^T d_b^T, d_c^T d_a^T, d_a^T d_d^T, d_c^T d_b^T, d_b^T d_d^T, d_c^T d_d^T\right], \\ d_z &= \min\left[d_b d_c, d_b d_d, d_a d_d, d_a d_c, d_a d_b\right]. \end{aligned} \tag{4.76}$$

The Z-type canonical logical operators L_Z can be categorized into the following six sets.

$$\begin{aligned} L_Z^1 &= \sum_{i,j,k,l} a_{ijkl}^1((v_i \otimes w_j \otimes L_C^k \otimes L_D^l)^T, 0^T, 0^T, 0^T, 0^T, 0^T)^T, \\ L_Z^2 &= \sum_{i,j,k,l} a_{ijkl}^2(0^T, (v_i \otimes L_B^j \otimes u_k \otimes L_D^l)^T, 0^T, 0^T, 0^T, 0^T)^T, \\ L_Z^3 &= \sum_{i,j,k,l} a_{ijkl}^3(0^T, 0^T, (v_i \otimes L_B^j \otimes L_C^k \otimes t_l)^T, 0^T, 0^T, 0^T)^T, \\ L_Z^4 &= \sum_{i,j,k,l} a_{ijkl}^4(0^T, 0^T, 0^T, (L_A^i \otimes w_j \otimes u_k \otimes L_D^l)^T, 0^T, 0^T)^T, \\ L_Z^5 &= \sum_{i,j,k,l} a_{ijkl}^5(0^T, 0^T, 0^T, 0^T, (L_A^i \otimes w_j \otimes L_C^k \otimes t_l)^T, 0^T)^T, \\ L_Z^6 &= \sum_{i,j,k,l} a_{ijkl}^6(0^T, 0^T, 0^T, 0^T, 0^T, (L_A^i \otimes L_B^j \otimes u_k \otimes t_l)^T)^T, \end{aligned} \tag{4.77}$$

where v_i, w_j, u_k, and t_l are unit vectors with length r_a, r_b, r_c and r_d, respectively. L_A^i, L_B^j, L_C^k and L_D^l are logical operators of the code defined by the parity check matrix δ_a, δ_b, δ_c and δ_d, respectively. Given that $\delta_a L_A^i = \delta_b L_B^j = \delta_c L_C^k = \delta_d L_D^l = 0$, one can confirm that $H_X L_Z^l = 0$ for $l \in \{1, 2, 3, 4, 5, 6\}$.

Similarly, the X-type canonical logical operators L_X can be categorized into six sets.

$$\begin{aligned}
L_X^1 &= \sum_{i,j,k,l} b_{ijkl}^1((L_A^i{}' \otimes L_B^j{}' \otimes u_k \otimes t_l)^T, 0^T, 0^T, 0^T, 0^T, 0^T)^T, \\
L_X^2 &= \sum_{i,j,k,l} b_{ijkl}^2(0^T, (L_A^i{}' \otimes w^j \otimes L_C^k{}' \otimes t_l)^T, 0^T, 0^T, 0^T, 0^T)^T, \\
L_X^3 &= \sum_{i,j,k,l} b_{ijkl}^3(0^T, 0^T, (L_A^i{}' \otimes w_j \otimes u_k \otimes L_D^l{}')^T, 0^T, 0^T, 0^T)^T, \\
L_X^4 &= \sum_{i,j,k,l} b_{ijkl}^4(0^T, 0^T, 0^T, (v_i \otimes L_B^j{}' \otimes L_C^k{}' \otimes t_l)^T, 0^T, 0^T)^T, \\
L_X^5 &= \sum_{i,j,k,l} b_{ijkl}^5(0^T, 0^T, 0^T, 0^T, (v_i \otimes L_B^j{}' \otimes u_k \otimes L_D^l{}')^T, 0^T)^T, \\
L_X^6 &= \sum_{i,j,k,l} b_{ijkl}^6(0^T, 0^T, 0^T, 0^T, 0^T, (v_i \otimes w_j \otimes L_C^k{}' \otimes L_D^l{}')^T)^T.
\end{aligned} \tag{4.78}$$

where v_i, w_j, u_k and t_l are unit vectors of length n_a, n_b, n_c and n_d, respectively. $L_A^i{}'$, $L_B^j{}'$, $L_C^k{}'$ and $L_D^l{}'$ are logical operators of the code defined by the parity check matrix δ_a^T, δ_b^T, δ_c^T and δ_d^T, respectively.

Both X and Z elementary canonical logical operators have the structure of tensor product of two classical logical operators (and two unit vectors). Thus, in the LDPC regime, with Lemma 4.3.2, by applying the same reasoning used for logical Z operators in 3D hypergraph product codes but extended to four dimension, we can establish the following lower bounds on the energy barriers for logical Z and X operators in the 4D hypergraph product code.

$$\begin{aligned}
\Delta(L_Z) &\geqslant \min[d_a, d_b, d_c, d_d], \\
\Delta(L_X) &\geqslant \min[d_a^T, d_b^T, d_c^T, d_d^T].
\end{aligned} \tag{4.79}$$

We state it formally in the following Theorem

THEOREM 4.5.1. *Let $\Delta(L_Z)$, $\Delta(L_X)$ be the energy barrier of the Z, X logical operators of the 4D hypergraph product code constructed from classical codes δ_a, δ_b, δ_c and δ_d, with parameters $[n_\ell, k_\ell, d_\ell, E_\ell]$ for $\ell \in \{a, b, c, d\}$. If the resulting quantum code is LDPC with*

sparsity parameters w_c, w_q*. When* $d_\ell, d_\ell^T \geqslant w_c w_q$ *for* $\ell \in \{a, b, c, d\}$*, then we have*

$$\Delta(L_Z) \geqslant \min[d_a, d_b, d_c, d_d], \tag{4.80}$$

$$\Delta(L_X) \geqslant \min[d_a^T, d_b^T, d_c^T, d_d^T]. \tag{4.81}$$

Similarly, with Lemma 4.3.1, we have

$$\Delta(L_Z) \geqslant \min[E_a, E_b, E_c, E_d], \tag{4.82}$$

$$\Delta(L_X) \geqslant \min[E_a^T, E_b^T, E_c^T, E_d^T]. \tag{4.83}$$

Moreover, if Conjecture 4.3.1 is true, we have

$$\Delta(L_Z) \geqslant \min\{d_\ell E_m : \ell, m \in \{a, b, c, d\}, \ell \neq m\}, \tag{4.84}$$

$$\Delta(L_X) \geqslant \min\{d_\ell^T E_m^T : \ell, m \in \{a, b, c, d\}, \ell \neq m\}, \tag{4.85}$$

which is tight.

4.6 Conclusion and outlook

In this work, we first gave a introduction to the confinement, soundness, and expansion properties of codes. We also discussed the relationships among these properties. Specifically, we showed the equivalence between left-expansion and linear confinement, and demonstrated formally that confinement property inherently induces a lower bound on energy barrier.

We then established a lower bound for the energy barrier of tensor product codes in Lemma 4.3.2, showing its relation to the distance of the underlying codes. Using this lemma, we proved that for 3D LDPC hypergraph product codes constructed from classical codes δ_a, δ_b, and δ_c with parameters $[n_\ell, k_\ell, d_\ell, E_\ell]$, where $\ell \in a, b, c$, the energy barrier of logical Z operators is lower-bounded by $\Omega(d_\ell)$, which improves upon the $\Omega(d_\ell^{\frac{1}{3}})$ bound derived from the soundness and confinement property. For logical X operators, the energy barrier corresponds to that of the transpose codes of these classical codes, paralleling the behavior in standard hypergraph

products. In the 4D case, we showed that X and Z logical operators have energy barriers lower-bounded by $\Omega(d_\ell)$ and $\Omega(d_\ell^T)$, respectively.

Several questions remain for further investigation. An immediate one concerns the mechanism behind energy barrier enhancement in HHGP codes. A possible explanation lies in the increased check redundancy introduced by these higher-dimensional structures. This raises follow-up questions: Is there a quantitative relationship between redundancy and energy barriers? What amount or structure of redundancy is sufficient to achieve a specified energy barrier?

The redundancy provided by higher-dimensional hypergraph products is sufficient for achieving soundness and confinement [148, 59]. Several works have also demonstrated the effectiveness of redundant parity check sets for simultaneously handling measurement and qubit errors [169, 170, 171]. However, establishing a general framework for introducing redundancy that ensures both confinement and macroscopic energy barriers remains an open question.

In contrast, quantum expander codes exhibit confinement even without check redundancy. This inspires the thought to explore alternative methods of providing confinement beyond redundancy or expansion, such as symmetries [172, 173]. Furthermore, quantum expander codes display unique statistical properties that may intrigue researchers, as demonstrated in recent studies [130, 174, 165, 175].

Finally, quantum codes with macroscopic energy barriers are conjectured to support local decoders [102, 176, 177, 178, 179, 52, 180, 181, 182, 183, 184], based on intuition from statistical mechanics. In such systems, local cooling processes can reduce system energy, suggesting that for codes with macroscopic energy barriers, a simple iterative local process could function effectively as a decoder. However, the design of local decoders for higher-dimensional hypergraph product codes remains an open research question.

CHAPTER 5

Conclusion and outlook

The main results of this thesis are on the energy barrier of hypergraph product codes and higher-dimensional hypergraph product codes. For the first time, we establish tight bounds on the energy barrier of a family of quantum low-density parity-check (LDPC) codes constructed using hypergraph products, which is a flexible framework for constructing quantum codes and offers many advantages. We accomplish this by first proving a general fact applicable to any quantum LDPC codes: equivalent logical operators up to stabilizers have the same energy barrier. Subsequently, we propose an approach that rigorously connects the energy barriers of quantum codes with the underlying classical codes. We present a general framework that establishes tight bounds on the energy barrier of LDPC hypergraph product codes, offering a way to discover self-correcting quantum memory and guiding the design of new decoders. As demonstrated in Chapter 3, the energy barrier property of LDPC codes has broader significance and potential applications.

To construct quantum codes with a macroscopic energy barrier using classical codes that lack this property, we demonstrate that higher-dimensional hypergraph product codes can achieve this goal. Generally speaking, the energy barrier of logical operators is shown to be lower bounded by $O(d)$, where d represents the distance of the classical codes.

Several questions remain for further investigation. An immediate question concerns the origin of the energy barrier enhancement in higher-dimensional hypergraph product codes. A likely explanation is the increased check redundancy introduced by the higher-dimensional structure. However, this raises additional questions: What is the precise relationship between redundancy and the energy barrier? How much redundancy is sufficient to achieve a given energy barrier?

Another issue, from a practical perspective, is to construct self-correcting quantum memories in lower dimensions, specifically 2D or 3D. Unfortunately, there is a no-go theorem for the 2D stabilizer code [139], and 3D stabilizer codes with translational and scale symmetries have been proven to not self-correct [146]. These no-go theorems arise because, in 2D, both logical operators are sting-like, and in 3D, only one type of logical operator has a membrane-like structure, but the other is still string-like. There are other approaches that can help bypass these no-go theorems. For instance, Haah's cubic code [131] breaks scale symmetries as the number of encoded qubits increases with the size of the system. The 3D Haah's cubic code was shown to achieve an $O(\log L)$ energy barrier, where L is the linear system size. Unluckily, it is only partially self-correcting due to the entropy effects [133, 103]. However, there is still an interesting question of whether the polynomials associated with Haah's cubic code are related to its energy barrier properties. Another approach to constructing self-correcting quantum memories is using subsystem codes. In particular, the 3D subsystem color code [149] and the 3D subsystem toric code [185] have been shown to have single-shot error correction properties and thus are very likely to be self-correcting.

It is believed that quantum codes with large energy barriers can support local decoders based on intuition from statistical mechanics. In such systems, a local cooling process can reduce the energy of the system. For codes with macroscopic energy barriers, it is expected that a simple local process could iteratively lower energy, acting as a decoder. However, designing a practical local decoder for such quantum codes remains an open problem. An example is the small-set flip decoder for quantum expander codes [106, 164]. Although not strictly local, it is considered 'local' in a broader sense because each step involves a constant number of qubits.

Quantum expander codes present a puzzling case because of their unusual thermodynamic properties. In typical models of self-correcting memory, thermodynamic phase transitions lead to a low-temperature phase where logical qubits are protected by thermally stable topological order. In contrast, quantum expander codes exhibit no thermodynamic phase transitions at nonzero temperatures because there is no check redundancy in quantum expander code [130]. Furthermore, many constant-rate quantum LDPC expander codes have "check-soundness," defining stable phases of matter with a constant zero-temperature entropy density, thereby

violating the third law of thermodynamics [186]. These special features inspire the further need to investigate the thermodynamic properties of quantum expander codes.

Bibliography

[1] Richard P Feynman. 'Simulating physics with computers'. In: *Feynman and computation*. CRC Press, 1982, pp. 133–153. DOI: 10.1007/BF02650179.

[2] Peter W Shor. 'Algorithms for quantum computation: discrete logarithms and factoring'. In: *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, pp. 124–134.

[3] David Deutsch and Richard Jozsa. 'Rapid solution of problems by quantum computation'. In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (1992), pp. 553–558.

[4] Lov K Grover. 'A fast quantum mechanical algorithm for database search'. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 212–219.

[5] Sergey Bravyi et al. 'High-threshold and low-overhead fault-tolerant quantum memory'. In: *Nature* 627.8005 (2024), pp. 778–782.

[6] Rajeev Acharya et al. *Quantum error correction below the surface code threshold*. 2024. arXiv: 2408.13687 [quant-ph]. URL: https://arxiv.org/abs/2408.13687.

[7] Ciaran Ryan-Anderson et al. 'Realization of real-time fault-tolerant quantum error correction'. In: *Physical Review X* 11.4 (2021), p. 041058.

[8] Ben W Reichardt et al. 'Demonstration of quantum computation and error correction with a tesseract code'. In: *arXiv preprint arXiv:2409.04628* (2024).

[9] Ben W. Reichardt et al. *Logical computation demonstrated with a neutral atom quantum processor*. 2024. arXiv: 2411.11822 [quant-ph]. URL: https://arxiv.org/abs/2411.11822.

[10] Frank Arute et al. ‘Quantum supremacy using a programmable superconducting processor’. In: *Nature* 574.7779 (2019), pp. 505–510.

[11] Han-Sen Zhong et al. ‘Quantum computational advantage using photons’. In: *Science* 370.6523 (2020), pp. 1460–1463.

[12] Abhinav Deshpande et al. ‘Quantum computational advantage via high-dimensional Gaussian boson sampling’. In: *Science advances* 8.1 (2022), eabi7894.

[13] Yulin Wu et al. ‘Strong quantum computational advantage using a superconducting quantum processor’. In: *Physical review letters* 127.18 (2021), p. 180501.

[14] Isaac L Chuang et al. ‘Quantum computers, factoring, and decoherence’. In: *Science* 270.5242 (1995), pp. 1633–1635. URL: https://www.science.org/doi/abs/10.1126/science.270.5242.1633.

[15] Michael A Nielsen and Isaac L Chuang. ‘Quantum computation and quantum information’. In: *Phys. Today* 54.2 (2001), p. 60. URL: https://www.cambridge.org/highereducation/books/quantum-computation-and-quantum-information/01E10196D0A682A6AEFFEA52D53BE9AE?utm_campaign=shareaholic&utm_medium=copy_link&utm_source=bookmark.

[16] Peter W. Shor. ‘Scheme for Reducing Decoherence in Quantum Computer Memory’. In: *Physical Review A* 52.4 (Oct. 1995), R2493–R2496. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.52.R2493. URL: https://link.aps.org/doi/10.1103/PhysRevA.52.R2493 (visited on 13/11/2021).

[17] Andrew Steane. ‘Multiple-particle interference and quantum error correction’. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 452.1954 (1996), pp. 2551–2577. DOI: 10.1098/rspa.1996.0136. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1996.0136.

[18] Peter W Shor. ‘Fault-tolerant quantum computation’. In: *Proceedings of 37th conference on foundations of computer science*. IEEE. 1996, pp. 56–65. DOI: https://doi.org/10.48550/arXiv.quant-ph/9605011.

[19] A Yu Kitaev. ‘Quantum computations: algorithms and error correction’. In: *Russian Mathematical Surveys* 52.6 (1997), p. 1191.

[20] Emanuel Knill, Raymond Laflamme and Wojciech H Zurek. 'Resilient quantum computation'. In: *Science* 279.5349 (1998), pp. 342–345.

[21] Panos Aliferis, Daniel Gottesman and John Preskill. 'Quantum accuracy threshold for concatenated distance-3 codes'. In: *arXiv preprint quant-ph/0504218* (2005).

[22] Dorit Aharonov and Michael Ben-Or. 'Fault-Tolerant Quantum Computation with Constant Error Rate'. In: *SIAM Journal on Computing* 38.4 (2008), pp. 1207–1282. DOI: 10.1137/S0097539799359385.

[23] Dorit Aharonov, Alexei Kitaev and John Preskill. 'Fault-Tolerant Quantum Computation with Long-Range Correlated Noise'. In: *Phys. Rev. Lett.* 96 (5 Feb. 2006), p. 050504. DOI: 10.1103/PhysRevLett.96.050504. URL: https://link.aps.org/doi/10.1103/PhysRevLett.96.050504.

[24] Richard W Hamming. 'Error detecting and error correcting codes'. In: *The Bell system technical journal* 29.2 (1950), pp. 147–160.

[25] Irving S Reed and Gustave Solomon. 'Polynomial codes over certain finite fields'. In: *Journal of the society for industrial and applied mathematics* 8.2 (1960), pp. 300–304.

[26] Frank Schiller and Tina Mattes. 'Analysis of CRC-polynomials for Safety-critical Communication by Deterministic and Stochastic Automata'. In: *Fault Detection, Supervision and Safety of Technical Processes 2006*. Elsevier, 2007, pp. 944–949.

[27] Peter Elias. 'Coding for noisy channels'. In: *IRE WESCON Convention Record, 1955*. Vol. 2. 1955, pp. 94–104.

[28] Claude Elwood Shannon. 'A mathematical theory of communication'. In: *The Bell system technical journal* 27.3 (1948), pp. 379–423.

[29] R Tanner. 'A recursive approach to low complexity codes'. In: *IEEE Transactions on information theory* 27.5 (1981), pp. 533–547.

[30] R. Gallager. 'Low-density parity-check codes'. In: *IRE Transactions on Information Theory* 8.1 (1962), pp. 21–28. DOI: 10.1109/TIT.1962.1057683.

[31] Judea Pearl. 'Reverend Bayes on inference engines: A distributed hierarchical approach'. In: *Probabilistic and Causal Inference: The Works of Judea Pearl*. 2022, pp. 129–138.

[32] D. Dieks. 'Communication by EPR devices'. In: *Physics Letters A* 92.6 (1982), pp. 271–272. ISSN: 0375-9601. DOI: https://doi.org/10.1016/0375-9601(82)90084-6. URL: https://www.sciencedirect.com/science/article/pii/0375960182900846.

[33] William K Wootters and Wojciech H Zurek. 'A single quantum cannot be cloned'. In: *Nature* 299.5886 (1982), pp. 802–803.

[34] Andrew M Steane. 'Error correcting codes in quantum theory'. In: *Physical Review Letters* 77.5 (1996), p. 793.

[35] Daniel Gottesman. 'Stabilizer Codes and Quantum Error Correction'. en. In: (May 1997). arXiv: quant-ph/9705052. URL: http://arxiv.org/abs/quant-ph/9705052 (visited on 17/04/2021).

[36] A. R. Calderbank and Peter W. Shor. 'Good quantum error-correcting codes exist'. In: *Phys. Rev. A* 54 (2 Aug. 1996), pp. 1098–1105. DOI: 10.1103/PhysRevA.54.1098. URL: https://link.aps.org/doi/10.1103/PhysRevA.54.1098.

[37] Zunaira Babar et al. 'Fifteen Years of Quantum LDPC Coding and Improved Decoding Strategies'. In: *IEEE Access* 3 (2015), pp. 2492–2519. DOI: 10.1109/ACCESS.2015.2503267.

[38] Nikolas P. Breuckmann and Jens Niklas Eberhardt. 'Quantum Low-Density Parity-Check Codes'. In: *PRX Quantum* 2.4 (Oct. 2021). ISSN: 2691-3399. DOI: 10.1103/prxquantum.2.040101. URL: http://dx.doi.org/10.1103/PRXQuantum.2.040101.

[39] Daniel Gottesman. 'Fault-tolerant quantum computation with constant overhead'. In: *Quantum Info. Comput.* 14.15–16 (Nov. 2014), pp. 1338–1372. ISSN: 1533-7146. DOI: 10.5555/2685179.2685184. URL: https://dl.acm.org/doi/10.5555/2685179.2685184.

[40] Jean-Pierre Tillich and Gilles Zemor. 'Quantum LDPC Codes With Positive Rate and Minimum Distance Proportional to the Square Root of the Blocklength'. In: *IEEE*

Transactions on Information Theory 60.2 (Feb. 2014), pp. 1193–1202. ISSN: 1557-9654. DOI: 10.1109/tit.2013.2292061. URL: http://dx.doi.org/10.1109/TIT.2013.2292061.

[41] Michael H Freedman, David A Meyer and Feng Luo. *Z2-systolic freedom and quantum codes*. 2002. URL: https://www.taylorfrancis.com/chapters/edit/10.1201/9781420035377-13/z2-systolic-freedom-quantum-codes-michael-freedman-david-meyer-feng-luo.

[42] Shai Evra, Tali Kaufman and Gilles Zémor. 'Decodable Quantum LDPC Codes beyond the $\sqrt{n}$ Distance Barrier Using High-Dimensional Expanders'. In: *SIAM Journal on Computing* 0.0 (0), FOCS20-276-FOCS20–316. DOI: 10.1137/20M1383689. URL: https://doi.org/10.1137/20M1383689.

[43] Tali Kaufman and Ran J. Tessler. 'New cosystolic expanders from tensors imply explicit Quantum LDPC codes with $\Omega(\sqrt{n}\log kn)$ distance'. In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 1317–1329. ISBN: 9781450380539. DOI: 10.1145/3406325.3451029. URL: https://doi.org/10.1145/3406325.3451029.

[44] Matthew B. Hastings, Jeongwan Haah and Ryan O'Donnell. 'Fiber bundle codes: breaking the $n^{1/2}$ polylog(n) barrier for Quantum LDPC codes'. In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2021. Virtual, Italy: Association for Computing Machinery, 2021, pp. 1276–1288. ISBN: 9781450380539. DOI: 10.1145/3406325.3451005. URL: https://doi.org/10.1145/3406325.3451005.

[45] Pavel Panteleev and Gleb Kalachev. 'Quantum LDPC Codes With Almost Linear Minimum Distance'. In: *IEEE Transactions on Information Theory* 68.1 (2022), pp. 213–229. DOI: 10.1109/TIT.2021.3119384.

[46] Nikolas P. Breuckmann and Jens N. Eberhardt. 'Balanced Product Quantum Codes'. In: *IEEE Transactions on Information Theory* 67.10 (Oct. 2021), pp. 6653–6674. ISSN: 1557-9654. DOI: 10.1109/tit.2021.3097347. URL: http://dx.doi.org/10.1109/TIT.2021.3097347.

[47] Pavel Panteleev and Gleb Kalachev. 'Asymptotically good Quantum and locally testable classical LDPC codes'. In: STOC 2022 (2022), pp. 375–388. DOI: 10.1145/3519935.3520017. URL: https://doi.org/10.1145/3519935.3520017.

[48] A. Leverrier and G. Zemor. 'Quantum Tanner codes'. In: *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (Nov. 2022), pp. 872–883. DOI: 10.1109/FOCS54457.2022.00117. URL: https://doi.ieeecomputersociety.org/10.1109/FOCS54457.2022.00117.

[49] Irit Dinur et al. 'Good Quantum LDPC Codes with Linear Time Decoders'. In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. STOC 2023 (2023), pp. 905–918. DOI: 10.1145/3564246.3585101. URL: https://doi.org/10.1145/3564246.3585101.

[50] Maxime A. Tremblay, Nicolas Delfosse and Michael E. Beverland. 'Constant-Overhead Quantum Error Correction with Thin Planar Connectivity'. In: *Phys. Rev. Lett.* 129 (5 July 2022), p. 050504. DOI: 10.1103/PhysRevLett.129.050504. URL: https://link.aps.org/doi/10.1103/PhysRevLett.129.050504.

[51] Qian Xu et al. 'Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays'. In: *arXiv* (2023). DOI: 10.48550/arXiv.2308.08648.

[52] Eric Dennis et al. 'Topological quantum memory'. In: *Journal of Mathematical Physics* 43.9 (Sept. 2002), pp. 4452–4505. ISSN: 1089-7658. DOI: 10.1063/1.1499754. URL: http://dx.doi.org/10.1063/1.1499754.

[53] Sergey Bravyi et al. *Subsystem surface codes with three-qubit check operators*. arXiv:1207.1443 [cond-mat, physics:quant-ph]. Dec. 2013. DOI: 10.48550/arXiv.1207.1443. URL: http://arxiv.org/abs/1207.1443 (visited on 16/02/2023).

[54] Nikolas P. Breuckmann and Barbara M. Terhal. 'Constructions and Noise Threshold of Hyperbolic Surface Codes'. In: *IEEE Trans. Inform. Theory* 62.6 (June 2016). arXiv:1506.04029 [quant-ph], pp. 3731–3744. ISSN: 0018-9448, 1557-9654. DOI: 10.1109/TIT.2016.2555700. URL: http://arxiv.org/abs/1506.04029 (visited on 16/02/2023).

[55] Oscar Higgott and Nikolas P. Breuckmann. 'Subsystem codes with high thresholds by gauge fixing and reduced qubit overhead'. In: *Phys. Rev. X* 11.3 (Aug. 2021). arXiv:2010.09626 [quant-ph], p. 031039. ISSN: 2160-3308. DOI: 10.1103/PhysRevX.11.031039. URL: http://arxiv.org/abs/2010.09626 (visited on 16/02/2023).

[56] J. Pablo Bonilla Ataides et al. 'The XZZX Surface Code'. In: *Nat Commun* 12.1 (Apr. 2021). arXiv:2009.07851 [quant-ph], p. 2172. ISSN: 2041-1723. DOI: 10.1038/s41467-021-22274-1. URL: http://arxiv.org/abs/2009.07851 (visited on 16/02/2023).

[57] Muyuan Li et al. '2-D Compass Codes'. In: *Phys. Rev. X* 9.2 (May 2019). arXiv:1809.01193 [quant-ph], p. 021041. ISSN: 2160-3308. DOI: 10.1103/PhysRevX.9.021041. URL: http://arxiv.org/abs/1809.01193 (visited on 16/02/2023).

[58] Christopher Chamberland et al. 'Topological and subsystem codes on low-degree graphs with flag qubits'. In: *Phys. Rev. X* 10.1 (Jan. 2020). arXiv:1907.09528 [cond-mat, physics:quant-ph], p. 011022. ISSN: 2160-3308. DOI: 10.1103/PhysRevX.10.011022. URL: http://arxiv.org/abs/1907.09528 (visited on 16/02/2023).

[59] Armanda O. Quintavalle et al. 'Single-Shot Error Correction of Three-Dimensional Homological Product Codes'. In: *PRX Quantum* 2.2 (June 2021). ISSN: 2691-3399. DOI: 10.1103/prxquantum.2.020340. URL: http://dx.doi.org/10.1103/PRXQuantum.2.020340.

[60] Benjamin J. Brown, Naomi H. Nickerson and Dan E. Browne. 'Fault-tolerant error correction with the gauge color code'. In: *Nat Commun* 7.1 (July 2016). arXiv:1503.08217 [quant-ph], p. 12302. ISSN: 2041-1723. DOI: 10.1038/ncomms12302. URL: http://arxiv.org/abs/1503.08217 (visited on 16/02/2023).

[61] Aleksander Kubica and Nicolas Delfosse. 'Efficient color code decoders in $d \geq 2$ dimensions from toric code decoders'. In: *Quantum* 7 (Feb. 2023), p. 929. ISSN: 2521-327X. DOI: 10.22331/q-2023-02-21-929. URL: http://dx.doi.org/10.22331/q-2023-02-21-929.

[62] Benjamin J. Brown and Dominic J. Williamson. 'Parallelized quantum error correction with fracton topological codes'. In: *Phys. Rev. Research* 2.1 (Mar. 2020). arXiv:1901.08061 [cond-mat, physics:quant-ph], p. 013303. ISSN: 2643-1564. DOI: 10.1103/PhysRevResearch.2.013303. URL: http://arxiv.org/abs/1901.08061 (visited on 16/02/2023).

[63] Georgia M. Nixon and Benjamin J. Brown. 'Correcting spanning errors with a fractal code'. In: *IEEE Trans. Inform. Theory* 67.7 (July 2021), pp. 4504–4516. ISSN: 0018-9448, 1557-9654. DOI: 10.1109/TIT.2021.3068359. URL: http://arxiv.org/abs/2002.11738 (visited on 16/02/2023).

[64] Zijun Chen et al. 'Exponential suppression of bit or phase flip errors with repetitive error correction'. In: *Nature* 595.7867 (July 2021). arXiv:2102.06132 [quant-ph], pp. 383–387. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-021-03588-y. URL: http://arxiv.org/abs/2102.06132 (visited on 16/02/2023).

[65] Nicolas Delfosse and Naomi H. Nickerson. 'Almost-linear time decoding algorithm for topological codes'. en. In: *Quantum* 5 (Dec. 2021). arXiv:1709.06218 [quant-ph], p. 595. ISSN: 2521-327X. DOI: 10.22331/q-2021-12-02-595. URL: http://arxiv.org/abs/1709.06218 (visited on 26/01/2023).

[66] Guillaume Duclos-Cianci and David Poulin. 'Fast Decoders for Topological Quantum Codes'. In: *Phys. Rev. Lett.* 104 (5 Feb. 2010), p. 050504. DOI: 10.1103/PhysRevLett.104.050504. URL: https://link.aps.org/doi/10.1103/PhysRevLett.104.050504.

[67] Adrian Hutter, James R. Wootton and Daniel Loss. 'Efficient Markov chain Monte Carlo algorithm for the surface code'. In: *Phys. Rev. A* 89 (2 Feb. 2014), p. 022326. DOI: 10.1103/PhysRevA.89.022326. URL: https://link.aps.org/doi/10.1103/PhysRevA.89.022326.

[68] Thomas M. Stace and Sean D. Barrett. 'Error correction and degeneracy in surface codes suffering loss'. In: *Phys. Rev. A* 81 (2 Feb. 2010), p. 022317. DOI: 10.1103/PhysRevA.81.022317. URL: https://link.aps.org/doi/10.1103/PhysRevA.81.022317.

[69] Patricio Fuentes et al. ‘Degeneracy and its impact on the decoding of sparse quantum codes’. In: *IEEE Access* 9 (2021), pp. 89093–89119.

[70] Nithin Raveendran and Bane Vasić. ‘Trapping sets of quantum LDPC codes’. In: *Quantum* 5 (2021), p. 562.

[71] Pavel Panteleev and Gleb Kalachev. ‘Degenerate Quantum LDPC Codes With Good Finite Length Performance’. In: *Quantum* 5 (Nov. 2021), p. 585. ISSN: 2521-327X. DOI: 10.22331/q-2021-11-22-585. URL: https://doi.org/10.22331/q-2021-11-22-585.

[72] Joschka Roffe et al. ‘Decoding across the quantum low-density parity-check code landscape’. In: *Phys. Rev. Res.* 2 (4 Dec. 2020), p. 043423. DOI: 10.1103/PhysRevResearch.2.043423. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.2.043423.

[73] Stasiu Wolanski and Ben Barber. ‘Ambiguity Clustering: an accurate and efficient decoder for qLDPC codes’. In: *arXiv preprint arXiv:2406.14527* (2024).

[74] T Hillmann et al. ‘Localized statistics decoding: A parallel decoding algorithm for quantum low-density parity-check codes (2024)’. In: *arXiv preprint arXiv:2406.18655* ().

[75] Antonio deMarti iOlius and Josu Etxezarreta Martinez. ‘The closed-branch decoder for quantum LDPC codes’. In: *arXiv e-prints* (2024), arXiv–2402.

[76] Antonio deMarti iOlius et al. ‘An almost-linear time decoding algorithm for quantum LDPC codes under circuit-level noise’. In: *arXiv e-prints* (2024), arXiv–2409.

[77] Barbara M. Terhal. ‘Quantum error correction for quantum memories’. In: *Reviews of Modern Physics* 87.2 (Apr. 2015), pp. 307–346. ISSN: 1539-0756. DOI: 10.1103/revmodphys.87.307. URL: http://dx.doi.org/10.1103/RevModPhys.87.307.

[78] Austin G. Fowler et al. ‘Surface codes: Towards practical large-scale quantum computation’. In: *Phys. Rev. A* 86 (3 Sept. 2012), p. 032324. DOI: 10.1103/PhysRevA.86.032324. URL: https://link.aps.org/doi/10.1103/PhysRevA.86.032324.

[79] James William Harrington. *Analysis of quantum error-correcting codes: symplectic lattice codes and toric codes*. California Institute of Technology, 2004.

[80] Michael Herold et al. 'Cellular-automaton decoders for topological quantum memories'. en. In: *npj Quantum Inf* 1.1 (Oct. 2015), p. 15010. ISSN: 2056-6387. DOI: 10.1038/npjqi.2015.10. URL: https://www.nature.com/articles/npjqi201510 (visited on 03/02/2023).

[81] Guillaume Dauphinais and David Poulin. 'Fault-Tolerant Quantum Error Correction for non-Abelian Anyons'. en. In: *Commun. Math. Phys.* 355.2 (Oct. 2017), pp. 519–560. ISSN: 0010-3616, 1432-0916. DOI: 10.1007/s00220-017-2923-9. URL: http://link.springer.com/10.1007/s00220-017-2923-9 (visited on 03/02/2023).

[82] Andrei L Toom. 'Stable and attractive trajectories in multicomponent systems'. In: *Multicomponent random systems* 6 (1980), pp. 549–575.

[83] Charles H. Bennett and G. Grinstein. 'Role of Irreversibility in Stabilizing Complex and Nonergodic Behavior in Locally Interacting Discrete Systems'. In: *Phys. Rev. Lett.* 55 (7 Aug. 1985), pp. 657–660. DOI: 10.1103/PhysRevLett.55.657. URL: https://link.aps.org/doi/10.1103/PhysRevLett.55.657.

[84] Fernando Pastawski, Lucas Clemente and Juan Ignacio Cirac. 'Quantum memories based on engineered dissipation'. In: *Phys. Rev. A* 83 (1 Jan. 2011), p. 012304. DOI: 10.1103/PhysRevA.83.012304. URL: https://link.aps.org/doi/10.1103/PhysRevA.83.012304.

[85] Nikolas P. Breuckmann et al. *Local Decoders for the 2D and 4D Toric Code*. 2016. arXiv: 1609.00510 [quant-ph]. URL: https://arxiv.org/abs/1609.00510.

[86] Nikolas P. Breuckmann and Xiaotong Ni. 'Scalable Neural Network Decoders for Higher Dimensional Quantum Codes'. In: *Quantum* 2 (May 2018), p. 68. ISSN: 2521-327X. DOI: 10.22331/q-2018-05-24-68. URL: http://dx.doi.org/10.22331/q-2018-05-24-68.

[87] Alexei Yu Kitaev, Alexander Shen and Mikhail N Vyalyi. *Classical and quantum computation*. 47. American Mathematical Soc., 2002.

[88] Andrew J. Landahl, Jonas T. Anderson and Patrick R. Rice. *Fault-tolerant quantum computing with color codes*. 2011. arXiv: 1108.5738 [quant-ph]. URL: https://arxiv.org/abs/1108.5738.

[89] Panos Aliferis, Daniel Gottesman and John Preskill. *Accuracy threshold for postselected quantum computation*. 2007. arXiv: quant-ph/0703264 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0703264.

[90] Panos Aliferis and Andrew W. Cross. 'Subsystem Fault Tolerance with the Bacon-Shor Code'. In: *Physical Review Letters* 98.22 (May 2007). ISSN: 1079-7114. DOI: 10.1103/physrevlett.98.220502. URL: http://dx.doi.org/10.1103/PhysRevLett.98.220502.

[91] Panos Aliferis and John Preskill. 'Fibonacci scheme for fault-tolerant quantum computation'. In: *Physical Review A* 79.1 (Jan. 2009). ISSN: 1094-1622. DOI: 10.1103/physreva.79.012332. URL: http://dx.doi.org/10.1103/PhysRevA.79.012332.

[92] Andrew W. Cross, David P. DiVincenzo and Barbara M. Terhal. *A comparative code study for quantum fault-tolerance*. 2009. arXiv: 0711.1556 [quant-ph]. URL: https://arxiv.org/abs/0711.1556.

[93] Panos Aliferis and John Preskill. 'Fault-tolerant quantum computation against biased noise'. In: *Physical Review A* 78.5 (Nov. 2008). ISSN: 1094-1622. DOI: 10.1103/physreva.78.052331. URL: http://dx.doi.org/10.1103/PhysRevA.78.052331.

[94] Masayuki Ohzeki. 'Locations of multicritical points for spin glasses on regular lattices'. In: *Phys. Rev. E* 79 (2 Feb. 2009), p. 021129. DOI: 10.1103/PhysRevE.79.021129. URL: https://link.aps.org/doi/10.1103/PhysRevE.79.021129.

[95] Takuya Ohno et al. 'Phase structure of the random-plaquette gauge model: accuracy threshold for a toric quantum memory'. In: *Nuclear Physics B* 697.3 (Oct. 2004), pp. 462–480. ISSN: 0550-3213. DOI: 10.1016/j.nuclphysb.2004.07.003. URL: http://dx.doi.org/10.1016/j.nuclphysb.2004.07.003.

[96] David S. Wang, Austin G. Fowler and Lloyd C. L. Hollenberg. 'Surface code quantum computing with error rates over 1%'. In: *Phys. Rev. A* 83 (2 Feb. 2011), p. 020302. DOI: 10.1103/PhysRevA.83.020302. URL: https://link.aps.org/doi/10.1103/PhysRevA.83.020302.

[97] Bryan Eastin and Emanuel Knill. 'Restrictions on Transversal Encoded Quantum Gate Sets'. In: *Physical Review Letters* 102.11 (Mar. 2009). ISSN: 1079-7114. DOI: 10.1103/physrevlett.102.110502. URL: http://dx.doi.org/10.1103/PhysRevLett.102.110502.

[98] Earl T Campbell, Barbara M Terhal and Christophe Vuillot. 'Roads towards fault-tolerant universal quantum computation'. In: *Nature* 549.7671 (2017), pp. 172–179.

[99] R Raussendorf, J Harrington and K Goyal. 'Topological fault-tolerance in cluster state quantum computation'. In: *New Journal of Physics* 9.6 (June 2007), pp. 199–199. ISSN: 1367-2630. DOI: 10.1088/1367-2630/9/6/199. URL: http://dx.doi.org/10.1088/1367-2630/9/6/199.

[100] Austin G. Fowler, Ashley M. Stephens and Peter Groszkowski. 'High-threshold universal quantum computation on the surface code'. In: *Physical Review A* 80.5 (Nov. 2009). ISSN: 1094-1622. DOI: 10.1103/physreva.80.052312. URL: http://dx.doi.org/10.1103/PhysRevA.80.052312.

[101] Emanuel Knill and Raymond Laflamme. *Concatenated Quantum Codes*. 1996. arXiv: quant-ph/9608012 [quant-ph].

[102] Anthony Leverrier, Jean-Pierre Tillich and Gilles Zemor. 'Quantum Expander Codes'. In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, Oct. 2015. DOI: 10.1109/focs.2015.55. URL: http://dx.doi.org/10.1109/FOCS.2015.55.

[103] Sergey Bravyi and Jeongwan Haah. 'Quantum Self-Correction in the 3D Cubic Code Model'. In: *Phys. Rev. Lett.* 111 (20 Nov. 2013), p. 200501. DOI: 10.1103/PhysRevLett.111.200501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.111.200501.

[104] Kamil P. Michnicki. '3D Topological Quantum Memory with a Power-Law Energy Barrier'. In: *Phys. Rev. Lett.* 113 (13 Sept. 2014), p. 130501. DOI: 10.1103/

PhysRevLett.113.130501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.113.130501.

[105] Karthik Siva and Beni Yoshida. 'Topological order and memory time in marginally-self-correcting quantum memory'. In: *Phys. Rev. A* 95 (3 Mar. 2017), p. 032324. DOI: 10.1103/PhysRevA.95.032324. URL: https://link.aps.org/doi/10.1103/PhysRevA.95.032324.

[106] Omar Fawzi, Antoine Grospellier and Anthony Leverrier. 'Constant overhead quantum fault tolerance with quantum expander codes'. In: *Commun. ACM* 64.1 (Dec. 2020), pp. 106–114. ISSN: 0001-0782. DOI: 10.1145/3434163. URL: https://doi.org/10.1145/3434163.

[107] Alexey A. Kovalev et al. 'Numerical and analytical bounds on threshold error rates for hypergraph-product codes'. In: *Phys. Rev. A* 97 (6 June 2018), p. 062320. DOI: 10.1103/PhysRevA.97.062320. URL: https://link.aps.org/doi/10.1103/PhysRevA.97.062320.

[108] Anirudh Krishna and David Poulin. 'Fault-Tolerant Gates on Hypergraph Product Codes'. In: *Phys. Rev. X* 11 (1 Feb. 2021), p. 011023. DOI: 10.1103/PhysRevX.11.011023. URL: https://link.aps.org/doi/10.1103/PhysRevX.11.011023.

[109] Guangqi Zhao, Andrew C. Doherty and Isaac H. Kim. 'Energy Barrier of Hypergraph Product Codes'. In: *Phys. Rev. Lett.* 134 (18 May 2025), p. 180601. DOI: 10.1103/PhysRevLett.134.180601. URL: https://link.aps.org/doi/10.1103/PhysRevLett.134.180601.

[110] Weilei Zeng and Leonid P. Pryadko. 'Higher-Dimensional Quantum Hypergraph-Product Codes with Finite Rates'. In: *Phys. Rev. Lett.* 122 (23 June 2019), p. 230501. DOI: 10.1103/PhysRevLett.122.230501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.122.230501.

[111] Shi Jie Samuel Tan and Lev Stambler. *Effective Distance of Higher Dimensional HGPs and Weight-Reduced Quantum LDPC Codes*. 2024. arXiv: 2409.02193 [quant-ph]. URL: https://arxiv.org/abs/2409.02193.

[112] Argyris Giannisis Manes and Jahan Claes. 'Distance-preserving stabilizer measurements in hypergraph product codes'. In: (2023). arXiv: 2308.15520 [quant-ph].

[113] Qian Xu et al. *Fast and Parallelizable Logical Computation with Homological Product Codes*. 2024. arXiv: 2407.18490 [quant-ph]. URL: https://arxiv.org/abs/2407.18490.

[114] Oscar Higgott and Nikolas P. Breuckmann. 'Improved Single-Shot Decoding of Higher-Dimensional Hypergraph-Product Codes'. In: *PRX Quantum* 4.2 (May 2023). ISSN: 2691-3399. DOI: 10.1103/prxquantum.4.020332. URL: http://dx.doi.org/10.1103/PRXQuantum.4.020332.

[115] Guangqi Zhao. *Improved energy barrier in higher-dimensional hypergraph product codes*. 2025. arXiv: 2506.19219 [quant-ph]. URL: https://arxiv.org/abs/2506.19219.

[116] Daniel Gottesman. *An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation*. 2009. arXiv: 0904.2557 [quant-ph]. URL: https://arxiv.org/abs/0904.2557.

[117] Penghua Chen, B. Yan and Shawn X. Cui. *Quantum circuits for toric code and X-cube fracton model*. 2022. DOI: 10.22331/q-2024-03-13-1276.

[118] Pengcheng Liao and David L. Feder. 'Quantum circuit for toric code state preparation via graph states'. In: 2021. URL: https://api.semanticscholar.org/CorpusID:232320735.

[119] K. Satzinger et al. *Realizing topologically ordered states on a quantum processor*. 2021. DOI: 10.1126/science.abi8378.

[120] Pengcheng Liao and David L. Feder. 'Graph-state representation of the toric code'. In: *Phys. Rev. A* 104 (1 July 2021), p. 012432. DOI: 10.1103/PhysRevA.104.012432. URL: https://link.aps.org/doi/10.1103/PhysRevA.104.012432.

[121] A.Yu. Kitaev. 'Fault-tolerant quantum computation by anyons'. In: *Annals of Physics* 303.1 (2003), pp. 2–30. ISSN: 0003-4916. DOI: https://doi.org/10.1016/S0003-4916(02)00018-0. URL: https://www.sciencedirect.com/science/article/pii/S0003491602000180.

[122] John Preskill. 'Reliable quantum computers'. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.1969 (Jan. 1998), pp. 385–410. ISSN: 1471-2946. DOI: 10.1098/rspa.1998.0167. URL: http://dx.doi.org/10.1098/rspa.1998.0167.

[123] Ruben Verresen, Mikhail D. Lukin and Ashvin Vishwanath. 'Prediction of Toric Code Topological Order from Rydberg Blockade'. en. In: *Phys. Rev. X* 11.3 (July 2021), p. 031005. ISSN: 2160-3308. DOI: 10.1103/PhysRevX.11.031005. URL: https://link.aps.org/doi/10.1103/PhysRevX.11.031005 (visited on 05/10/2022).

[124] Chenyang Wang, Jim Harrington and John Preskill. 'Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory'. In: *Annals of Physics* 303.1 (Jan. 2003), pp. 31–58. ISSN: 0003-4916. DOI: 10.1016/s0003-4916(02)00019-2. URL: http://dx.doi.org/10.1016/S0003-4916(02)00019-2.

[125] Michael P. Zaletel et al. 'Colloquium: Quantum and classical discrete time crystals'. In: *Rev. Mod. Phys.* 95 (3 July 2023), p. 031001. DOI: 10.1103/RevModPhys.95.031001. URL: https://link.aps.org/doi/10.1103/RevModPhys.95.031001.

[126] Raditya Weda Bomantara. 'Quantum repetition codes as building blocks of large period discrete time crystals'. en. In: *Phys. Rev. B* 104.18 (Nov. 2021). arXiv:2102.09113 [cond-mat, physics:quant-ph], p. L180304. ISSN: 2469-9950, 2469-9969. DOI: 10.1103/PhysRevB.104.L180304. URL: http://arxiv.org/abs/2102.09113 (visited on 05/10/2022).

[127] Raditya Weda Bomantara. 'Nonlocal discrete time crystals in periodically driven surface codes'. en. In: *Phys. Rev. B* 104.6 (Aug. 2021). arXiv:2106.01748 [quant-ph], p. 064302. ISSN: 2469-9950, 2469-9969. DOI: 10.1103/PhysRevB.104.064302. URL: http://arxiv.org/abs/2106.01748 (visited on 05/10/2022).

[128] M. B. Hastings. 'An Area Law for One Dimensional Quantum Systems'. en. In: *J. Stat. Mech.* 2007.08 (Aug. 2007). arXiv: 0705.2024, P08024–P08024. ISSN: 1742-5468.

DOI: 10.1088/1742-5468/2007/08/P08024. URL: http://arxiv.org/abs/0705.2024 (visited on 09/03/2022).

[129] Seth Lloyd. 'Universal quantum simulators'. In: *Science* 273.5278 (1996), pp. 1073–1078. DOI: 10.1126/science.273.5278.1073.

[130] Yifan Hong, Jinkang Guo and Andrew Lucas. *Quantum memory at nonzero temperature in a thermodynamically trivial system*. 2024. arXiv: 2403.10599 [quant-ph]. URL: https://arxiv.org/abs/2403.10599.

[131] Jeongwan Haah. 'Local stabilizer codes in three dimensions without string logical operators'. In: *Physical Review A* 83.4 (2011), p. 042330. ISSN: 1050-2947. DOI: 10.1103/physreva.83.042330. eprint: 1101.1962.

[132] R. Alicki et al. 'On Thermal Stability of Topological Qubit in Kitaev's 4D Model'. In: *Open Systems & Information Dynamics* 17.01 (2010), pp. 1–20. DOI: 10.1142/S1230161210000023. URL: https://doi.org/10.1142/S1230161210000023.

[133] Sergey Bravyi and Jeongwan Haah. 'Energy Landscape of 3D Spin Hamiltonians with Topological Order'. In: *Phys. Rev. Lett.* 107 (15 Oct. 2011), p. 150504. DOI: 10.1103/PhysRevLett.107.150504. URL: https://link.aps.org/doi/10.1103/PhysRevLett.107.150504.

[134] Dominic J. Williamson and Nouédyn Baspin. *Layer Codes*. 2023. arXiv: 2309.16503 [quant-ph].

[135] Ting-Chun Lin, Adam Wills and Min-Hsiu Hsieh. *Geometrically Local Quantum and Classical Codes from Subdivision*. 2023. arXiv: 2309.16104 [quant-ph].

[136] Antoine Grospellier et al. 'Combining hard and soft decoders for hypergraph product codes'. In: *Quantum* 5 (Apr. 2021), p. 432. ISSN: 2521-327X. DOI: 10.22331/q-2021-04-15-432. URL: https://doi.org/10.22331/q-2021-04-15-432.

[137] Lawrence Z. Cohen et al. 'Low-overhead fault-tolerant quantum computing using long-range connectivity'. In: *Science Advances* 8.20 (2022), eabn1717. DOI: 10.1126/sciadv.abn1717. URL: https://www.science.org/doi/abs/10.1126/sciadv.abn1717.

[138] Armanda O. Quintavalle, Paul Webster and Michael Vasmer. 'Partitioning qubits in hypergraph product codes to implement logical gates'. In: *Quantum* 7 (Oct. 2023), p. 1153. ISSN: 2521-327X. DOI: 10.22331/q-2023-10-24-1153. URL: https://doi.org/10.22331/q-2023-10-24-1153.

[139] Sergey Bravyi and Barbara Terhal. 'A no-go theorem for a two-dimensional self-correcting quantum memory based on stabilizer codes'. In: *New Journal of Physics* 11.4 (2009), p. 043029. DOI: 10.1088/1367-2630/11/4/043029. eprint: 0810.1983.

[140] Rudolf Peierls. 'On Ising's model of ferromagnetism'. In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol. 32. 3. Cambridge University Press. 1936, pp. 477–481.

[141] Michael Plischke and Birger Bergersen. *Equilibrium statistical physics*. World scientific, 1994. URL: https://www.worldscientific.com/worldscibooks/10.1142/5660#t=aboutBook.

[142] Armanda O. Quintavalle and Earl T. Campbell. 'ReShape: A Decoder for Hypergraph Product Codes'. In: *IEEE Transactions on Information Theory* 68.10 (2022), pp. 6569–6584. DOI: 10.1109/TIT.2022.3184108.

[143] M. Sipser and D.A. Spielman. 'Expander codes'. In: *IEEE Transactions on Information Theory* 42.6 (1996), pp. 1710–1722. DOI: 10.1109/18.556667.

[144] Tibor Rakovszky and Vedika Khemani. *The Physics of (good) LDPC Codes II. Product constructions*. 2024. arXiv: 2402.16831 [quant-ph]. URL: https://arxiv.org/abs/2402.16831.

[145] Alexey A. Kovalev and Leonid P. Pryadko. 'Quantum Kronecker sum-product low-density parity-check codes with finite rate'. In: *Phys. Rev. A* 88 (1 July 2013), p. 012311. DOI: 10.1103/PhysRevA.88.012311. URL: https://link.aps.org/doi/10.1103/PhysRevA.88.012311.

[146] Beni Yoshida. 'Feasibility of self-correcting quantum memory and thermal stability of topological order'. In: *Annals of Physics* 326.10 (2011), pp. 2566–2633. ISSN: 0003-4916. DOI: https://doi.org/10.1016/j.aop.2011.06.001.

URL: https://www.sciencedirect.com/science/article/pii/S0003491611001023.

[147] Koujin Takeda and Hidetoshi Nishimori. 'Self-dual random-plaquette gauge model and the quantum toric code'. In: *Nuclear Physics B* 686.3 (2004), pp. 377–396.

[148] Earl T Campbell. 'A theory of single-shot error correction for adversarial noise'. In: *Quantum Science and Technology* 4.2 (Feb. 2019), p. 025006. ISSN: 2058-9565. DOI: 10.1088/2058-9565/aafc8f. URL: http://dx.doi.org/10.1088/2058-9565/aafc8f.

[149] Héctor Bombín. 'Single-Shot Fault-Tolerant Quantum Error Correction'. In: *Phys. Rev. X* 5 (3 Sept. 2015), p. 031043. DOI: 10.1103/PhysRevX.5.031043. URL: https://link.aps.org/doi/10.1103/PhysRevX.5.031043.

[150] Dave Bacon et al. 'Sparse Quantum Codes From Quantum Circuits'. In: *IEEE Transactions on Information Theory* 63.4 (2017), pp. 2464–2479. DOI: 10.1109/TIT.2017.2663199.

[151] Renyu Wang, Hsiang-Ku Lin and Leonid P. Pryadko. *Abelian and non-abelian quantum two-block codes*. 2023. arXiv: 2305.06890 [quant-ph]. URL: https://arxiv.org/abs/2305.06890.

[152] Hsiang-Ku Lin and Leonid P. Pryadko. 'Quantum two-block group algebra codes'. In: *Phys. Rev. A* 109 (2 Feb. 2024), p. 022407. DOI: 10.1103/PhysRevA.109.022407. URL: https://link.aps.org/doi/10.1103/PhysRevA.109.022407.

[153] Sergey Bravyi et al. 'High-threshold and low-overhead fault-tolerant quantum memory'. In: *Nature* 627.8005 (2024), pp. 778–782. DOI: 10.1038/s41586-024-07107-7. URL: https://doi.org/10.1038/s41586-024-07107-7.

[154] Manuel Blum, Michael Luby and Ronitt Rubinfeld. 'Self-testing/correcting with applications to numerical problems'. In: *Proceedings of the twenty-second annual ACM symposium on Theory of computing*. 1990, pp. 73–83.

[155] Sanjeev Arora. *Probabilistic checking of proofs and hardness of approximation problems*. University of California, Berkeley, 1994.

[156] Ronitt Rubinfeld and Madhu Sudan. 'Robust characterizations of polynomials with applications to program testing'. In: *SIAM Journal on Computing* 25.2 (1996), pp. 252–271.

[157] Katalin Friedl and Madhu Sudan. 'Some improvements to total degree tests'. In: *Proceedings Third Israel Symposium on the Theory of Computing and Systems*. IEEE. 1995, pp. 190–198.

[158] Dorit Aharonov and Lior Eldar. 'Quantum locally testable codes'. In: *SIAM Journal on Computing* 44.5 (2015), pp. 1230–1262.

[159] Lior Eldar and Aram W Harrow. 'Local Hamiltonians whose ground states are hard to approximate'. In: *2017 IEEE 58th annual symposium on foundations of computer science (FOCS)*. IEEE. 2017, pp. 427–438.

[160] Michael H. Freedman and Matthew B. Hastings. 'Quantum systems on non-k-hyperfinite complexes: a generalization of classical statistical mechanics on expander graphs'. In: *Quantum Info. Comput.* 14.1–2 (Jan. 2014), pp. 144–180. ISSN: 1533-7146.

[161] Dorit Aharonov, Itai Arad and Thomas Vidick. 'Guest column: the quantum PCP conjecture'. In: *Acm sigact news* 44.2 (2013), pp. 47–79.

[162] Tibor Rakovszky and Vedika Khemani. *The Physics of (good) LDPC Codes I. Gauging and dualities*. 2023. arXiv: 2310.16032 [quant-ph]. URL: https://arxiv.org/abs/2310.16032.

[163] Eli Ben-Sasson et al. 'Locally testable codes require redundant testers'. In: *SIAM Journal on Computing* 39.7 (2010), pp. 3230–3247.

[164] Omar Fawzi, Antoine Grospellier and Anthony Leverrier. 'Efficient decoding of random errors for quantum expander codes'. In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. 2018, pp. 521–534.

[165] Benedikt Placke et al. *Topological Quantum Spin Glass Order and its realization in qLDPC codes*. 2024. arXiv: 2412.13248 [quant-ph]. URL: https://arxiv.org/abs/2412.13248.

[166] Eli Ben-Sasson, Prahladh Harsha and Sofya Raskhodnikova. 'Some 3CNF properties are hard to test'. In: *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*. STOC '03. San Diego, CA, USA: Association for Computing

Machinery, 2003, pp. 345–354. ISBN: 1581136749. DOI: 10.1145/780542.780594. URL: https://doi.org/10.1145/780542.780594.

[167] Shi Jie Samuel Tan and Lev Stambler. *Effective Distance of Higher Dimensional HGPs and Weight-Reduced Quantum LDPC Codes*. 2025. arXiv: 2409.02193 [quant-ph]. URL: https://arxiv.org/abs/2409.02193.

[168] 'Homological error correction: Classical and quantum codes'. In: *Journal of mathematical physics* 48.5 (2007).

[169] Yuichiro Fujiwara. 'Ability of stabilizer quantum error correction to protect itself from its own imperfection'. In: *Phys. Rev. A* 90 (6 Dec. 2014), p. 062304. DOI: 10.1103/PhysRevA.90.062304. URL: https://link.aps.org/doi/10.1103/PhysRevA.90.062304.

[170] Alexei Ashikhmin, Ching-Yi Lai and Todd A. Brun. 'Quantum Data-Syndrome Codes'. In: *IEEE Journal on Selected Areas in Communications* 38.3 (2020), pp. 449–462. DOI: 10.1109/JSAC.2020.2968997.

[171] Nicolas Delfosse, Ben W. Reichardt and Krysta M. Svore. 'Beyond Single-Shot Fault-Tolerant Quantum Error Correction'. In: *IEEE Transactions on Information Theory* 68.1 (2022), pp. 287–301. DOI: 10.1109/TIT.2021.3120685.

[172] Sam Roberts and Stephen D. Bartlett. 'Symmetry-Protected Self-Correcting Quantum Memories'. In: *Phys. Rev. X* 10 (3 Aug. 2020), p. 031041. DOI: 10.1103/PhysRevX.10.031041. URL: https://link.aps.org/doi/10.1103/PhysRevX.10.031041.

[173] Charles Stahl. 'Self-Correction from Higher-Form Symmetry Protection on a Boundary'. In: *PRX Quantum* 4 (3 Sept. 2023), p. 030341. DOI: 10.1103/PRXQuantum.4.030341. URL: https://link.aps.org/doi/10.1103/PRXQuantum.4.030341.

[174] Chao Yin and Andrew Lucas. *Low-density parity-check codes as stable phases of quantum matter*. 2024. arXiv: 2411.01002 [quant-ph]. URL: https://arxiv.org/abs/2411.01002.

[175] Wojciech De Roeck et al. *LDPC stabilizer codes as gapped quantum phases: stability under graph-local perturbations*. 2024. arXiv: 2411.02384 [quant-ph]. URL: https://arxiv.org/abs/2411.02384.

[176] Omar Fawzi, Antoine Grospellier and Anthony Leverrier. 'Constant Overhead Quantum Fault-Tolerance with Quantum Expander Codes'. In: *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, Oct. 2018, pp. 743–754. DOI: 10.1109/focs.2018.00076. URL: http://dx.doi.org/10.1109/FOCS.2018.00076.

[177] Aleksander Kubica and John Preskill. 'Cellular-Automaton Decoders with Provable Thresholds for Topological Codes'. In: *Phys. Rev. Lett.* 123 (2 July 2019), p. 020501. DOI: 10.1103/PhysRevLett.123.020501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.123.020501.

[178] Nikolas P. Breuckmann et al. 'Local decoders for the 2D and 4D toric code'. In: *Quantum Info. Comput.* 17.3–4 (Mar. 2017), pp. 181–208. ISSN: 1533-7146. URL: https://dl.acm.org/doi/abs/10.5555/3179532.3179533.

[179] Matthew B. Hastings. 'Decoding in hyperbolic spaces: quantum LDPC codes with linear rate and efficient error correction'. In: *Quantum Info. Comput.* 14.13–14 (Oct. 2014), pp. 1187–1202. ISSN: 1533-7146. URL: https://dl.acm.org/doi/abs/10.5555/2685164.2685173.

[180] Michael Herold et al. 'Cellular-automaton decoders for topological quantum memories'. In: *npj Quantum information* 1.1 (2015), pp. 1–8. URL: https://www.nature.com/articles/npjqi201510.

[181] Nikolas P. Breuckmann and Xiaotong Ni. 'Scalable Neural Network Decoders for Higher Dimensional Quantum Codes'. In: *Quantum* 2 (May 2018), p. 68. ISSN: 2521-327X. DOI: 10.22331/q-2018-05-24-68. URL: http://dx.doi.org/10.22331/q-2018-05-24-68.

[182] Guillaume Dauphinais and David Poulin. 'Fault-tolerant quantum error correction for non-abelian anyons'. In: *Communications in Mathematical Physics* 355 (2017), pp. 519–560. URL: https://link.springer.com/article/10.1007/s00220-017-2923-9.

[183] Fernando Pastawski, Lucas Clemente and Juan Ignacio Cirac. 'Quantum memories based on engineered dissipation'. In: *Phys. Rev. A* 83 (1 Jan. 2011), p. 012304. DOI: 10.1103/PhysRevA.83.012304. URL: https://link.aps.org/doi/10.1103/PhysRevA.83.012304.

[184] Michael Vasmer, Dan E. Browne and Aleksander Kubica. 'Cellular automaton decoders for topological quantum codes with noisy measurements and beyond'. In: *Scientific Reports* 11.1 (2021), p. 2027. DOI: 10.1038/s41598-021-81138-2. URL: https://doi.org/10.1038/s41598-021-81138-2.

[185] Aleksander Kubica and Michael Vasmer. 'Single-shot quantum error correction with the three-dimensional subsystem toric code'. In: *Nature Communications* 13.1 (Oct. 2022). ISSN: 2041-1723. DOI: 10.1038/s41467-022-33923-4. URL: http://dx.doi.org/10.1038/s41467-022-33923-4.

[186] Chao Yin and Andrew Lucas. *Low-density parity-check codes as stable phases of quantum matter*. 2024. arXiv: 2411.01002 [quant-ph]. URL: https://arxiv.org/abs/2411.01002.

[187] Jean-Pierre Tillich and Gilles Zemor. 'Quantum LDPC codes with positive rate and minimum distance proportional to $n^{1/2}$'. en. In: *2009 IEEE International Symposium on Information Theory*. Seoul, South Korea: IEEE, June 2009, pp. 799–803. ISBN: 978-1-4244-4312-3. DOI: 10.1109/ISIT.2009.5205648. URL: http://ieeexplore.ieee.org/document/5205648/ (visited on 24/10/2022).

[188] Anthony Leverrier, Simon Apers and Christophe Vuillot. 'Quantum XYZ Product Codes'. In: *Quantum* 6 (July 2022), p. 766. ISSN: 2521-327X. DOI: 10.22331/q-2022-07-14-766. URL: http://dx.doi.org/10.22331/q-2022-07-14-766.

APPENDIX A

Appendix

A1 Homology

A path on the surface $\mathcal{M}$ is defined as a map $\alpha : I \to \mathcal{M}$, where I is interval $[0, 1]$. While $\alpha(0) = \alpha(1)$, we call this path a loop.

Two paths α, β with common endpoints (means $\alpha(0) = \beta(0), \alpha(1) = \beta(1)$) are equivalent or homotopic when there exists a continuous function $H : I \times I \to \mathcal{M}$, such that

$$H(t, 0) = \alpha(t), H(t, 1) = \beta(t), t \in [0, 1] \tag{A.1}$$

and

$$H(0, s) = \alpha(0), H(1, s) = \alpha(1), s \in [0, 1] \tag{A.2}$$

which means that α can be continuous deformed to β, we write as $\alpha \sim \beta$. Essentially, H gives a family of paths that connect α and β.

A loop is defined as a path α with $\alpha(0) = \alpha(1)$, a special case is the constant loop: $\alpha(t) = \alpha(0)$ for all $t \in [0, 1]$. We can define a multiplication of paths as follows: If α, β are paths in $\mathcal{M}$, with property $\alpha(1) = \alpha(0)$, we can define $\alpha * \beta$ as

$$\alpha * \beta(t) = \begin{cases} \alpha(2t) & 0 \leq t \leq \frac{1}{2} \\ \beta(2t) & \frac{1}{2} \leq t \leq 1 \end{cases} \tag{A.3}$$

The equivalent class of α is denoted (all the paths that are homotopic to α) as $[\alpha]$. Thus, $[\alpha] = [\beta]$ if $\alpha \sim \beta$. Moreover, the multiplication is well-defined in these equivalent classes as

$$[\alpha] * [\beta] = [\alpha * \beta] \tag{A.4}$$

Then, the equivalence class, combined with the multiplication, gives us an algebra object $\pi(\mathcal{M})$, which is called the fundamental group of $\mathcal{M}$, and the equivalence class of constant loop is the identity of this group.

In mathematics, homology is a general way to abstract and generalize the boundary, which is associated with a sequence of algebraic objects. Here, we focus only on the homology in the group $\mathbb{Z}_2$. Consider a torus, cellulated with a square lattice, which is the same as a square lattice with periodic boundary conditions.

This lattice consists of three different objects: 0-dimensional vertices, 1-dimensional edges, and 2-dimensional plaquettes. We call an n-dimensional object an n-cell in homology language, which means the vertices are 0-cells, the edges are 1-cells, and the plaquettes are 2-cells.

The homology on the group $\mathbb{Z}_2$ means that for each n-cell, an element from $\mathbb{Z}_2$ (0 or 1) is associated with this n-cell. With such a setup, chains, which are the basic objects in homology, can be defined. n-chain is just one specific $\mathbb{Z}_2$ assignment of an n-cell. two n-chains can be added together via the bitwise addition operator of the group $\mathbb{Z}_2$ (which is addition modulo 2). Therefore, the n-chains will form a group. If the number of elements in n-cell is n, then the order of this group will be 2^n. Moreover, there are n independent generators for this Abelian group.

Now, consider the boundary map of this homology, which is associated with different n-chains. For example, considering a single plaquette on the square lattice (a 2-chain), its boundary contains four edges (a one-chain) enclosing it. In general, an n-boundary map ∂_n is a map for the set of n-chains c to the (n-1)-chains d

$$\partial_n(c) = d \tag{A.5}$$

and this map preserves the group structure.

Define an n-cycle as an n-chain c satisfying

$$\partial_n c = 0_{n-1}, \tag{A.6}$$

which means this n-chain has no boundary. Note that cycles form a group because

$$\partial_n(c+d) = \partial_n(c) + \partial_n(d) = 0_{n-1} + 0_{n-1} = 0_{n-1} \tag{A.7}$$

the group product between cycles will also be a cycle, so n-cycles is a subgroup of the n-chains.

One can also define n-boundary chains as the boundary of an (n+1)-chain. Simply, it is the n-boundary. Notice that every n-boundary is an n-cycle, but not every n-cycle is a boundary.

Every n-boundary is an n-cycle, which means applying a boundary map twice will always get a null chain

$$\partial_{n-1}\partial_n c = 0_{n-2} \tag{A.8}$$

This equation is called the fundamental lemma of homology.

There are three groups: the full group of n-chains C_n, which contains the subgroup of n-cycles Z_n, and the subgroup of n-boundaries B_n. Homology describes the relation between these groups to characterize the topology of the manifold.

There is an important equivalence relation, homological equivalence, between two n-chains. Two n-chains c_1 and c_2 are homologically equivalent if they are equal up to addition with an n-boundary b. Such two n-chains c_1 and c_2 have the same boundary. Therefore, what homological equivalence does for n-chains is partition the subgroup of n-chains with a given boundary into equivalence classes. This partitioning can be performed in cycles. In this way, the quotient group named as nth homology group H_n is defined as

$$H_n = \frac{Z_n}{B_n} \tag{A.9}$$

Finally, the nth Betti number β_n as the rank of the nth homology group is defined as

$$\beta_n = \operatorname{rank} H_n = \operatorname{rank} Z_n - \operatorname{rank} B_n \tag{A.10}$$

The Euler characteristic of the manifold and the Betti numbers have a relation

$$\chi = \beta_2 - \beta_1 + \beta_0. \tag{A.11}$$

For the torus, the Betti numbers are $\beta_0 = 1, \beta_1 = 2, \beta_2 = 1$. The number of encoded qubits on a surface is equal to the first Betti number β_1, which is the number of independent generators of the homological group H_1. Moreover, we can define cohomology in the dual lattice, and we also have co-chains, co-boundaries, and co-cycles, which hold the same structure as their counterpart in homology. Then, we have the cohomology group

$$H^n = \frac{Z^n}{B^n} \tag{A.12}$$

A2 CSS code and its homological description

Let C_1 and C_2 be two classical codes with relation $C_2 \subseteq C_1^{\perp}$. Then, one can construct a quantum error correcting code (CSS code) with H_1, H_2, where H_i is the parity check matrix of C_i, because

$$C_2 \subseteq C_1^{\perp} \Longleftrightarrow H_2 H_1^T = 0 \tag{A.13}$$

Within the description of the stabilizer code, the rows of H_2 give all X (or Z) type stabilizers. In contrast, the rows of H_1 contain all Z-type stabilizers (or X-type stabilizers), and X-type stabilizers commute with Z-type stabilizers because $H_2 H_1^T = 0$.

One can construct a CSS code with the property of homology. given a chain complex

$$C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \tag{A.14}$$

and its co-chain

$$\tilde{C}_0 \xrightarrow{\tilde{\partial}_0} \tilde{C}_1 \xrightarrow{\tilde{\partial}_1} \tilde{C}_0 \tag{A.15}$$

then we can define the parity check spaces $H_z, H_x \subseteq C_1$, where C_1 is the qubit space,

$$H_z = \text{im}\partial_2, \qquad H_x = \text{im}\tilde{\partial}_0 \tag{A.16}$$

and one can check the orthogonality by choosing any vector $z = \partial_2 f \in H_z$, $x = \tilde{\partial}_0 g \in H_x$, then

$$\begin{aligned} \langle x, z \rangle &= \langle \tilde{\partial}_0 g, \partial_2 f \rangle & \text{(A.17)} \\ &= \langle g, \partial_1 \partial_2 f \rangle & \text{(A.18)} \\ &= 0 & \text{(A.19)} \end{aligned}$$

as $\partial_1 \partial_2 = 0$. In this CSS code, the Z-type stabilizers belong to $im\partial_2$, and the X-type stabilizers belong to $im\tilde{\partial}_0$. Logical Z operators are in $\ker \partial_1 \setminus \text{im}\partial_2$, while logical X operators are in $\ker \tilde{\partial}_1 \setminus \text{im}\tilde{\partial}_0$, the number of qubits is given by the Betti number, i.e., $\ker \partial_1 / \text{im}\partial_2$.

The homologous interpretation of the CSS code is extremely useful because it connects the classical and quantum codes. Thus, it provides a way to construct quantum codes with classical codes. To some extent, the quantum code flourishes as we can use the numerous good classical codes that have been developed for many years.

Quantum low-density parity-check (LDPC) codes, designed for encoding many logical quantum bits within the same block, provide excellent performance for large-scale fault-tolerant quantum computing. Quantum LDPC codes have two attractive features. (i) They are defined as stabilizers that involve only a small number of quantum bits, making quantum LDPC codes easier to implement than general codes. (ii) some can encode k logical quantum bits into n physical quantum bits, $k = \Omega(n)$, with a large code distance d. With two classical LDPC codes, we can construct a quantum LDPC code using their homological property.

There are many quantum codes constructed in this way, such as the hypergraph product code (tensor product code)[187, 40], the fiber bundle code (twisted product code)[44], and the balanced product code[46].

A3 Hypergraph product code from the homological construction

Given two classical codes, represented by two length-1 chain complexes, one can construct a quantum code by taking the tensor product of these two classical codes

$$(C_1 \xrightarrow{\partial_C} C_0) \otimes (D_1 \xrightarrow{\partial_D} D_0) \tag{A.20}$$

which gives a new length-2 chain complex

$$(C_1 \otimes D_1) \xrightarrow{\partial_2} (C_0 \otimes D_1 + C_1 \otimes D_0) \xrightarrow{\partial_1} (C_0 \otimes D_0) \tag{A.21}$$

with

$$\partial_1 = \partial_2 = (\partial_c \otimes \mathrm{id}_D + \mathrm{id}_C \otimes \partial_D) \tag{A.22}$$

one can check that

$$\begin{aligned} \partial_1\partial_2 &= (\partial_C \otimes \mathrm{id}_D + \mathrm{id}_C \otimes \partial_D)^2 && \text{(A.23)} \\ &= \partial_C \otimes \partial_D + \partial_C \otimes \partial_D && \text{(A.24)} \\ &= 0 && \text{(A.25)} \end{aligned}$$

since we work over $\mathbb{Z}_2$. If we have two classical codes with parameters $[n_1, k_1, d_1]$, and $[n_2, k_2, d_2]$. Let $d_{min} = \min(d_1, d_2)$ denote the minimum distance of the two codes. The hypergraph product code is a $[n_1^2 + n_2^2, k_1^2 + k_2^2, d_{min}]$ quantum code[108].

It is worth noting that any CSS quantum code corresponds to a hypergraph product, while a general hypergraph product code is not necessarily a CSS code. The hypergraph product can be used to construct a non-CSS code through the XYZ product [188].

A4 Vector reshaping

Consider a basis $\mathcal{B}$ of the vector space $\mathbb{F}_2^{n_1} \otimes \mathbb{F}_2^{n_2}$:

$$\mathcal{B} = \{a_i \otimes b_j \mid i = 1, \ldots, n_1 \text{ and } j = 1, \ldots, n_2\}. \tag{A.26}$$

Then any vector $v \in \mathbb{F}_2^{n_1} \otimes \mathbb{F}_2^{n_2}$ can be written as

$$v = \sum_{a_i \otimes b_j \in \mathcal{B}} v_{ij} \, (a_i \otimes b_j) \tag{A.27}$$

for some $v_{ij} \in \mathbb{F}_2$. We call the $n_1 \times n_2$ matrix V with entries v_{ij} the *reshaping* of the vector v. By this definition, if A, B are respectively $m_1 \times n_1$ and $m_2 \times n_2$ matrices, then

$$(A \otimes B)v \Rightarrow AVB^T. \tag{A.28}$$

Define $\mathrm{wt}(M)$ as the number of ones in the vector (or matrix) M. We have $\mathrm{wt}((A \otimes B)v) = \mathrm{wt}(AVB^T)$.

Now, considering a vector v lies in the tensor product space $\mathbb{F}_2^{n_1} \otimes \mathbb{F}_2^{n_2} \otimes \mathbb{F}_2^{n_3}$, we need to extend the reshaping and transformation concepts to accommodate this higher-order tensor structure.

The basis for the tensor product space $\mathbb{F}_2^{n_1} \otimes \mathbb{F}_2^{n_2} \otimes \mathbb{F}_2^{n_3}$ is:

$$\mathcal{B} = \{a_i \otimes b_j \otimes c_k \mid i = 1, \ldots, n_1; j = 1, \ldots, n_2; k = 1, \ldots, n_3\}. \tag{A.29}$$

Thus, any vector v in this space can be written as

$$v = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} v_{ijk} \, (a_i \otimes b_j \otimes c_k), \tag{A.30}$$

where $v_{ijk} \in \mathbb{F}_2$. Here, the coefficients v_{ijk} form a 3-dimensional tensor (or array) V with shape $n_1 \times n_2 \times n_3$.

To analyze the action of a Kronecker product transformation, say $(A \otimes B \otimes C)$, we extend the idea of reshaping v: The vector v can be interpreted as a 3D tensor V with entries v_{ijk}.

Let A, B, and C be matrices of dimensions $m_1 \times n_1, m_2 \times n_2$, and $m_3 \times n_3$, respectively. The action of $A \otimes B \otimes C$ on v can be expressed as

$$(A \otimes B \otimes C)v \equiv AVB^T C^T. \tag{A.31}$$

Here, V is treated as a tensor, and the transformation applies A, B, and C along the first, second, and third modes of V, respectively. The resulting transformed tensor $AVB^T C^T$ is a new tensor of shape $m_1 \times m_2 \times m_3$.

The weight of the tensor V, denoted wt (V), counts the total number of 1 s in V. The Kronecker product transformation preserves the weight

$$\text{wt}((A \otimes B \otimes C)v) = \text{wt}\left(AVB^T C^T\right). \tag{A.32}$$

A4.1 Proof of Lemma 4.4.1

Recall that a Z-type Pauli error of the 3D hypergraph product code can be expressed as a bit-string $z = \left(z^{(1)}, z^{(2)}, z^{(3)}\right)^T$. The corresponding energy $\epsilon(z) = \text{wt}(H_X z)$ is

$$\begin{aligned} \epsilon(z) \;=\; & \text{wt}\left((I_{r_a} \otimes \delta_b \otimes I_{n_c})z^{(1)} + (\delta_a \otimes I_{r_b} \otimes I_{n_c})z^{(2)}\right) \\ & + \text{wt}\left((I_{r_a} \otimes I_{n_b} \otimes \delta_c)z^{(1)} + (\delta_a \otimes I_{n_b} \otimes I_{r_c})z^{(3)}\right) \\ & + \text{wt}\left((I_{n_a} \otimes I_{r_b} \otimes \delta_c)z^{(2)} + (I_{n_a} \otimes \delta_b \otimes I_{r_c})z^{(3)}\right). \end{aligned} \tag{A.33}$$

By applying vector reshaping, the energy can be written as

$$\begin{aligned} \epsilon(z) \;=\; & \text{wt}\left(Z^{(1)}\delta_b^T + \delta_a Z^{(2)}\right) + \text{wt}\left(Z^{(1)}\delta_c^T + \delta_a Z^{(3)}\right) \\ & + \text{wt}\left(Z^{(2)}\delta_c^T + Z^{(3)}\delta_b^T\right), \end{aligned} \tag{A.34}$$

where $Z^{(1)}$, $Z^{(2)}$ and $Z^{(3)}$ are the 3D matrices reshaped from $z^{(1)}$, $z^{(2)}$, and $z^{(3)}$, respectively. The jth a-slice of $Z^{(1)}$ is supported on qubit subset $r_1^j \otimes V_2 \otimes V_3$.

Using Eq. (A.34), we aim to prove a lower bound on the energy $\epsilon(z)$ [Lemma A4.1]. To that end, we shall use the following convention. Let L_a be a nontrivial codeword of δ_a^T.

The set of a-slices of $Z^{(1)}$ associated with the nonzero entries of L_a will play an important role. We define the index set of such columns as $\mathcal{C}(L_a)$:

$$\mathcal{C}(L_a) := \{j : (L_a)_j = 1\}. \tag{A.35}$$

Given a codeword L_a of δ_a^T, one can construct a matrix $Z^{1,s}$ from the 3D matrix $Z^{(1)}$ by summing all the a-slices in the set $\mathcal{C}(L_a)$. More formally,

$$Z^{1,s}_{jk} = \sum_{i:i\in\mathcal{C}(L_a)} Z^{(1)}_{ijk}, \tag{A.36}$$

where the addition is modulo 2. For example, if $L_a = 110100$, we would sum a-slices $1, 2$, and 4. Using Eq. (A.36), we can deform an arbitrary path to a path consisting of Paulis only supported on subset $r_1^i \otimes V_2 \otimes V_3$ for some i. [1] In particular, we can prove an inequality between the energy of the original Pauli and the deformed Pauli, proved in Lemma A4.1.

LEMMA A4.1.

$$\mathrm{wt}(Z^{1,s}\delta_b^T) \leqslant \mathrm{wt}\left(Z^{(1)}\delta_b^T + \delta_a Z^{(2)}\right). \tag{A.37}$$

PROOF. We prove this by contradiction. Consider the elements of $Z^{1,s}\delta_b^T$ and $Z^{(1)}\delta_b^T + \delta_a Z^{(2)}$. If $\mathrm{wt}(Z^{1,s}\delta_b^T) > \mathrm{wt}\left(Z^{(1)}\delta_b^T + \delta_a Z^{(2)}\right)$, then there exists j, k such that

$$\mathrm{wt}((Z^{1,s}\delta_b^T)_{j,k}) > \mathrm{wt}\left((Z^{(1)}\delta_b^T + \delta_a Z^{(2)})_{j,k}\right). \tag{A.38}$$

We will prove that Eq. (A.38) cannot be satisfied, thereby proving the claim.

Without loss of generality, consider the j, k elements. Notice that $(Z^{1,s}\delta_b^T)_{jk}$ is a bit, thus the weight of it is either 0 or 1. If $\mathrm{wt}((Z^{1,s}\delta_b^T)_{jk}) = 0$, Eq. (A.38) cannot be satisfied. Therefore, we consider the $\mathrm{wt}((Z^{1,s}\delta_b^T)_{jk}) = 1$ case.

If $\mathrm{wt}((Z^{1,s}\delta_b^T)_{jk}) = 1$, $(Z^{(1)}\delta_b^T)_{jk}$ must contain an odd number of ones on the a-slice in the set $\mathcal{C}(L_a)$. Otherwise, we would have had $\mathrm{wt}((Z^{1,s}\delta_b^T)_{jk}) = 0$, which is a contradiction. On

[1] The precise choice of i does not matter; any $i \in \mathcal{C}(L_a)$ would suffice.

the other hand, we remark that $(\delta_a Z^{(2)})_{jk}$ consists of an even number of ones on the a-slice set $\mathcal{C}(L_a)$. To see why, note that each jk-line of $\delta_a Z^{(2)}$ is a linear combination of the checks in δ_a^T. Because L_a is a codeword of δ_a^T, $(\delta_a Z^{(2)})_{jk} L_a = 0$. Therefore, in the $\mathrm{wt}((Z^{1,s}\delta_b^T)_{jk}) = 1$ case, the number of ones in $Z^{(1)}\delta_b^T + \delta_a Z^{(2)}$ on the jk-line and the a-slice in $\mathcal{C}(L_a)$ is odd.

Thus we conclude $\mathrm{wt}\left((Z^{(1)}\delta_b^T + \delta_a Z^{(2)})_{jk}\right) \geq 1$. As such, Eq. (A.38) cannot be satisfied. This completes the proof. □

With the same logic, one can prove that

LEMMA A4.2.

$$\mathrm{wt}(Z^{1,s}\delta_c^T) \leqslant \mathrm{wt}\left(Z^{(1)}\delta_c^T + \delta_a Z^{(3)}\right). \tag{A.39}$$

Now we are in a position to prove Lemma 4.4.1. We do so by identifying a path $r' = \{P_0', P_1', \cdots, P_F'\}$ that is only supported on $r_1^\alpha \otimes V_2 \otimes V_3$ while ensuring that $\epsilon_{\max}(r') \leqslant \epsilon_{\max}(r)$, for any path r supported on the entire qubit set. Lemma A4.1 and Lemma A4.2 suggest a way to deform the path r to the one supported on $r_1^\alpha \otimes V_2 \otimes V_3$.

Without loss of generality, let $r = \{P_0, P_1, \cdots, P_F\}$ be a path that $\Delta(L) = \epsilon_{\max}(r)$, with $P_0 = I$ and $P_F = L$. We consider a Pauli P_i in the path r. It will be convenient to work in its binary representation, written as $(p^{(1)}, p^{(2)}, p^{(3)})^T$, where $p^{(1)}$,$p^{(2)}$ and $p^{(3)}$ represent the Paulis supported on $C_1 \otimes V_2 \otimes V_3$, $V_1 \otimes C_2 \otimes V_3$ and $V_1 \times V_2 \otimes C_3$, respectively. First, we remove all the Paulis supported on $V_1 \otimes C_2 \otimes V_3$ and $V_1 \times V_2 \otimes C_3$ by setting $p^{(2)}$ and $p^{(3)}$ as the zero vector. Next, we apply the following transformations to $p^{(1)}$. We reshape $p^{(1)}$ and refer to the reshaped 3D matrix as $P^{(1)}$. Let L_a be a codeword of δ_a^T such that $(L_a)_\alpha = 1$. Consider a set of a-slices in $P^{(1)}$ corresponding to the index set $\mathcal{C}(L_a)$. Denoting each a-slice as r_1^k, where $k \in \mathcal{C}(L_a)$, we update the a-slice r_1^α in the following way:

$$r_1^\alpha \to r_1^\alpha + \sum_{k \in \mathcal{C}(L_a)\setminus\{\alpha\}} r_1^k. \tag{A.40}$$

Afterward, the other a-slices of $P^{(1)}$ are set to the zero matrices. This yields the deformed Pauli operator P_i'.

By construction, the resulting P_i' is supported on $r_1^\alpha \otimes V_2 \otimes V_3$. Note that r' is a valid path because $\mathrm{wt}(P_i' P_{i+1}') \leq 1$ for every i. Also, because $P_F' = P_F = L$, r' is still a path for L. Moreover, because $\epsilon\left(P_i'\right) \leqslant \epsilon\left(P_i\right)$ for all i [Lemma A4.1 and A4.2] as

$$\begin{aligned} \mathrm{wt}(Z^{1,s}\delta_c^T) + \mathrm{wt}(Z^{1,s}\delta_b^T) \leqslant \epsilon(z) = \mathrm{wt}\left(Z^{(1)}\delta_b^T + \delta_a Z^{(2)}\right) \\ + \mathrm{wt}\left(Z^{(1)}\delta_c^T + \delta_a Z^{(3)}\right) + \mathrm{wt}\left(Z^{(2)}\delta_c^T + Z^{(3)}\delta_b^T\right), \end{aligned} \tag{A.41}$$

we have $\epsilon_{\max}\left(r'\right) \leqslant \epsilon_{\max}(r)$. Both r' and r are paths for L, by definition $\epsilon_{\max}\left(r'\right) \geqslant \epsilon_{\max}(r)$, we conclude $\epsilon_{\max}\left(r'\right) = \epsilon_{\max}(r)$. Thus, by deforming r, we obtained a new path r' supported on $r_1^\alpha \otimes V_2 \otimes V_3$ that yields the energy barrier $\Delta(L)$.

This argument can be applied to prove similar lower bounds for logical operators on $V_1 \otimes r_2^\beta \otimes V_3$ and $V_1 \otimes V_2 \otimes r_3^\gamma$. To conclude, for any elementary canonical logical operator L supported on $r_1^\alpha \otimes V_2 \otimes V_3$ (resp. $V_1 \otimes r_2^\beta \otimes V_3$ and $V_1 \otimes V_2 \otimes r_3^\gamma$), their energy barrier can be given by a path supported on $r_1^\alpha \otimes V_2 \otimes V_3$ (resp. $V_1 \otimes r_2^\beta \otimes V_3$ and $V_1 \otimes V_2 \otimes r_3^\gamma$).

A4.2 Proof of Lemma 4.4.2

Any nontrivial canonical logical-Z operator L belongs to one of the following categories:

- Case 1: L is supported solely on the qubit subset $C_1 \otimes V_2 \otimes V_3$.
- Case 2: L is supported solely on the qubit subset $V_1 \otimes C_2 \otimes V_3$.
- Case 3: L is supported solely on the qubit subset $V_1 \otimes V_2 \otimes C_3$.
- Case 4: L is supported on multiple subsets.

We will focus solely on Case 1. Case 2 and Case 3 can be analyzed similarly by considering subsets $V_1 \otimes C_2 \otimes V_3$ and $V_1 \otimes V_2 \otimes C_3$, while case 4 can be treated as Case 1, 2 or 3.

Without loss of generality, let the energy barrier of L be attained by a path $r = \{P_0, P_1, \cdots, P_F\}$, with $P_0 = I$ and $P_F = L$. Similar to the approach taken in Lemma 4.4.1, we aim to deform the path r to the one supported on $r_1^k \otimes V_2 \otimes V_3$ for some k, such that the energy barrier of the deformed path lower bounds that of the r.

The deformation works in the same way as in the proof of Lemma 4.4.1. We describe this procedure again for the readers' convenience. Let L_a be a nontrivial codeword of δ_a^T and $\mathcal{C}(L_a)$ be its corresponding column index set. We consider a binary representation of a Pauli P_i, written as $(p^{(1)}, p^{(2)}, p^{(3)})^T$. As in the proof of Lemma 4.4.1, we remove the Paulis supported on $V_1 \otimes C_2 \otimes V_3$ and $V_1 \otimes V_2 \otimes C_3$ by setting $p^{(2)}, p^{(3)}$ as the zero vector. Next, reshape $p^{(1)}$ into a 3D matrix $P^{(1)}$ and update its columns in the following way. Choose $\alpha \in \mathcal{C}(L_a)$. This column is updated as Eq. (A.40). The other columns of $P^{(1)}$ are converted to zero vectors.

Thanks to Lemma A4.1 and A4.2, we obtain a new path $r' = \{P'_0, P'_1, \cdots, P'_F\}$ supported on $r_1^\alpha \otimes V_2 \otimes V_3$ with the property $\epsilon_{\max}(r') \leqslant \epsilon_{\max}(r)$. Note that P'_F, in the binary representation, is of the form $r_1^\alpha \otimes \bar{y} \otimes \bar{z}$, where $\bar{y}, \bar{z}$ are codewords of δ_b and δ_c respectively. Therefore, P'_F is either a nontrivial elementary canonical logical operator or the identity. In the latter case, P'_F is the trivial codeword (zero vector) in the binary representation. Henceforth, we denote this as $L' = P'_F$.

If L' is nontrivial, we can use the relation between the energy barriers of L and L':

$$\Delta(L) = \epsilon_{\max}(r) \geqslant \epsilon_{\max}(r') \geqslant \Delta(L'). \tag{A.42}$$

Because L' is an elementary logical operator, $\Delta(L)$ is greater or equal to the minimum energy barrier of elementary canonical logical operators. Thus, if L' is nontrivial, the proof follows immediately.

If L' is an identity, the above argument does not work. Fortunately, it turns out that for any L', one can choose L_a (the codeword of δ_a^T used in the current proof) such that L' is not an identity.

Without loss of generality, consider a canonical logical-Z operator L, expressed as

$$L = \sum_{i,j,k} \alpha_{ijk}(x_i \otimes \bar{y}_j \otimes \bar{z}_k, 0, 0)^T, \tag{A.43}$$

where (i) $\delta_b \bar{y}_j = \delta_c \bar{z}_k = 0$ and (ii) $x_i \notin \mathrm{Im}\,(\delta_a)$ are unit vectors. If a given path ends with L, its deformation (using Eq. (A.40)) yields the following operator L':

$$L' = \sum_{j,k} c_{jk} (x_\alpha \otimes \bar{y}_j \otimes \bar{z}_k, 0, 0)^T, \tag{A.44}$$

where $\alpha \in \mathcal{C}(L_a)$ and c_{jk} is defined as

$$c_{jk} := \sum_{i \in \mathcal{C}(L_a)} \alpha_{ikj}. \tag{A.45}$$

Note that L' is trivial if and only if $c_{jk} = 0$ for all j, k. Therefore, we aim to prove that there exists a choice of L_a such that $c_{jk} = 1$ for at least one choice of j, k.

Let us prove the contrapositive. Suppose $c_{jk} = 0$ for all k, for any choice of L_a. Consider the following vector:

$$u_{jk} := \sum_i \alpha_{ijk} x_i. \tag{A.46}$$

Note that $c_{jk} = u_{jk}^T L_a$. By our assumption, $c_{jk} = 0$ for any choice of L_a, so the inner product of u_{jk} with any codeword of δ_a^T must be zero. On the other hand, u_{jk}, if it is nonzero, must lie outside of $\mathrm{Im}(\delta_a)$ by the definition of the x_i. Thus, u_{jk} is not an element of the row space of δ_a^T. However, this is a contradiction for the following reasons. For a linear code, let H and G be the parity check matrix and the generator matrix. Then $v^T G = 0$ if and only if v is a vector in the row space of H. In our setup, if $c_{jk} = 0$ for any L_a, then $u_{jk}^T G = 0$. This implies that u_{jk} must be in the row space of δ_a^T, which contradicts the fact that it lies outside of $\mathrm{Im}(\delta_a)$. To conclude, there must be at least one choice of k, j such that $c_{jk} = 1$. Thus, there is always a choice of L_a such that L' is not an identity, thereby proving the claim.

Case 2 and Case 3 can be analyzed similarly to Case 1 by considering subset $V_1 \otimes C_2 \otimes V_3$ and $V_1 \otimes V_2 \otimes C_3$. For Case 4, one can treat it just as Case 1, Case 2, or Case 3. For example, when treating it as Case 1, the logical operator L has a nontrivial part in the subset $C_1 \otimes V_2 \otimes V_3$. One can prove there exists a codeword L_a of δ_a^T, such that after the deformation, the resulting L' is a nontrivial elementary canonical logical operator.

A4.3 Discussion on Conjecture 4.3.1

Let L_a and L_b denote the logical operators of δ_a and δ_b, with distances d_a, d_b and energy barriers E_a, E_b, respectively. Our goal is to determine the energy barrier of the operator $L_c = L_a \otimes L_b$, which serves as a logical operator for δ_c.

To compute the energy barrier of L_c, we consider a path $r = \{P_0, P_1, \cdots, P_F\}$ with $P_0 = I$ and $P_F = L_c$. Then

$$\Delta(L_c) = \min\{\epsilon_{\max}(r) : r \in w(0, L_c)\}. \tag{A.47}$$

Note that r is a path that walks on the bit of code δ_c. Given a $P_\ell \in r$, the energy (the number of violated checks) of P_ℓ is

$$\begin{aligned} \epsilon(P_\ell) &= \text{wt}(\delta_c P_\ell) \\ &= \text{wt}\left(\begin{pmatrix} \delta_a \otimes I_{n_b} \\ I_{n_a} \otimes \delta_b \end{pmatrix} P_\ell\right) \\ &= \text{wt}((\delta_a \otimes I_{n_b})P_\ell) + \text{wt}((I_{n_a} \otimes \delta_b)P_\ell). \end{aligned} \tag{A.48}$$

To proceed, one can express the identity matrices I_{n_a} and I_{n_b} as

$$I_{n_a} = \sum_{j=1}^{n_a} w_j w_j^T, \quad I_{n_b} = \sum_{k=1}^{n_b} u_k u_k^T. \tag{A.49}$$

Eq. (A.48) can then be further processed as

$$\begin{aligned}
\epsilon(P_\ell) &= \mathrm{wt}\left(\delta_a \otimes I_{n_b} P_\ell\right) + \mathrm{wt}\left(I_{n_a} \otimes \delta_b P_\ell\right) \\
&= \mathrm{wt}\left(\delta_a \otimes \sum_{k=1}^{n_b} u_k u_k^T P_\ell\right) + \mathrm{wt}\left(\sum_{j=1}^{n_a} w_j w_j^T \otimes \delta_b P_\ell\right) \\
&= \mathrm{wt}\left(\sum_{k=1}^{n_a} (\delta_a \otimes u_k)(I_{n_a} \otimes u_k^T) P_\ell\right) \\
&\quad + \mathrm{wt}\left(\sum_{j=1}^{n_a} (w_j \otimes \delta_b)(w_j^T \otimes I_{n_b}) P_\ell\right) \\
&= \mathrm{wt}\left(\sum_{k=1}^{n_b} (\delta_a \otimes u_k) P_\ell^k\right) + \mathrm{wt}\left(\sum_{j=1}^{n_a} (w_j \otimes \delta_b) P_\ell^j\right) \\
&= \sum_{k=1}^{n_b} \mathrm{wt}\left(\delta_a P_\ell^k\right) + \sum_{j=1}^{n_a} \mathrm{wt}\left(\delta_b P_\ell^j\right),
\end{aligned} \tag{A.50}$$

where

$$P_\ell^k = (I_{n_a} \otimes u_k^T) P_\ell, \quad P_\ell^j = (w_j^T \otimes I_{n_b}) P_\ell. \tag{A.51}$$

To calculate the energy barrier of L_c, we minimize $\epsilon_{\max}(P_\ell)$ over all possible r. Here, $\epsilon(P_\ell)$ is the sum of two components: $\sum_{k=1}^{n_b} \mathrm{wt}\left(\delta_a P_\ell^k\right)$ and $\sum_{j=1}^{n_a} \mathrm{wt}\left(\delta_b P_\ell^j\right)$. Although these two components depend on each other, we first consider minimizing them independently.

We can first minimize $\sum_{k=1}^{n_b} \mathrm{wt}\left(\delta_a P_\ell^k\right)$, followed by minimizing $\sum_{j=1}^{n_a} \mathrm{wt}\left(\delta_b P_\ell^j\right)$. To minimize $\sum_{k=1}^{n_b} \mathrm{wt}\left(\delta_a P_\ell^k\right)$, we apply L_a along the path r_a, which corresponds to the energy barrier of L_a for each k sequentially. The maximum energy contribution for this part is E_a. Subsequently, we minimize $\sum_{j=1}^{n_a} \mathrm{wt}\left(\delta_b P_\ell^j\right)$ by setting the order of k according to the path r_b that gives the energy barrier of L_b. The highest energy contribution for this part is $d_a E_b$. In particular, when the second part reaches $d_a E_b$, the first part has an energy of 0. Therefore, this path results in an energy barrier of $\epsilon_{\max}(r) = d_a E_b$. Alternatively, we can reverse the process by first minimizing $\sum_{j=1}^{n_a} \mathrm{wt}\left(\delta_b P_\ell^j\right)$ and then $\sum_{k=1}^{n_b} \mathrm{wt}\left(\delta_a P_\ell^k\right)$. Following a similar analysis, this approach leads to an energy barrier of $\epsilon_{\max}(r') = d_b E_a$.

To complete the argument, one needs to prove that there is no path with an energy barrier lower than $\min[d_a E_b, d_b E_a]$. We developed a program to verify the conjecture [2]. This program generates random codes with their logical operators, computes their energy barriers and distances, constructs the tensor product code, and calculates its energy barrier.

Computing the energy barrier of a general code is hard. Our implementation uses a best-first search algorithm that may not always identify the absolute minimum. In our tests across many random codes, most results support the conjecture. In the rare cases where the energy barrier of the tensor product code differs from our product-based estimate, the discrepancy is minor and can be attributed to computational limitations.

While our numerical simulations strongly support the conjecture, one needs to develop alternative combinatorial methods for a rigorous proof. We believe that solving this problem will not only strengthen our current findings but also have broader applications for tensor product codes, particularly given their close relationship to locally testable codes.

[2]Github repository: https://github.com/Guangqi-Phys/Energy-Barrier-of-tensor-product-codes.git