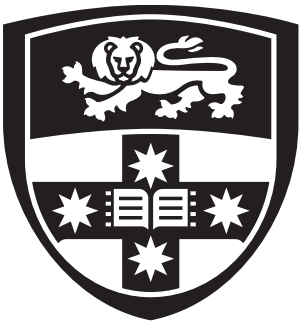
On the locality of quantum codes



Nouédyn Baspin

School of Physics
The University of Sydney

This thesis is submitted to the
Faculty of Science, School of Physics of The University of Sydney
in fulfillment of the requirements for the degree of
Doctor of Philosophy (Science)
July 2025

À maman et Zouzou, mon chez-moi loin de la maison

To Aimee and Niki; a four-dimensional girl living in my three dimensional space, and her cat

Acknowledgements

I cannot begin to imagine what my PhD experience would have been like without Anirudh Krishna. When I was stuck in Canada for the first few months due to covid he made sure I was never adrift, and kept feeding me his reflections and ideas. I have yet to meet someone as patient and with a comparable talent for explanations. We seem to share many obsessions and every conversation leads to a flow of questions I obsess over for a long time. In many ways he has been a step-advisor to me, and I hope this remains the case for many years.

I owe so much to my collaborators. With Harriet Apel, our year-long project was a scary, exciting, roller coaster. Countless times I would have lost faith and given up, if not for Harry's oasis of optimism and determination. Thank you for carrying this project with four arms when I was not pulling my weight.

When I first started working on the Layer codes and hit a wall, I asked around for help. I remember asking around at the IBM Summer School, pitching the construction, and asking for advice. I was about the give up when Dominic Williamson took me up on it, and I am forever thankful. During our collaboration I both marveled at, and was frightened by breadth of his knowledge. Starting a conversation with him often feels like chasing the Wonderland's rabbit through his tunnels, and he's always out of time.

Speaking of wizards, the dull days at the office were often enchanted by my partner in crime, Arkin Tikku. I could not count the times that my afternoon plans were derailed because he was talking about things that were too interesting. Concepts that are fuzzy in my mind often become clear through his words.

I cannot count my lucky stars enough for having Stephen Bartlett as my supervisor. Whether it be travels, conferences, accommodating my project ideas, it was always taken care of. I felt confident in my research because I knew I had his support.

On a personal note, I would like to thank my family's everlasting support, despite me never really explaining exactly what it is I do.

I often skipped days at the office, and I blame Aimee. The nest we have created is simply too amazing to ever leave. I used to be more than comfortable moving around,

now I get homesick the second I step on a plane. I admire and envy your effortless creativity, your inexplicable talent to simply will things into existence; you capture my attention with the fewest words. I must admit to having a secret collaborator, an uncredited peer, an unforgiving reviewer, the most handsome boy: Niki, our cat. He always knows how to press the keyboard to turn my latex file into incomprehensible gibberish; he becomes jealous and stomps when he feels like my laptop gets too much attention. Without his warm cuddles, I would have never survived the winter.

I would like to acknowledge the Sydney Quantum Academy for their financial support through the SQA PhD Scholarship.

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. This thesis contains fewer than 80,000 words, excluding the bibliography. No AI tools, generative, or otherwise, were used.

Nouédyn Baspin
July 2025

Statement of Student Contribution

This thesis consists of an introduction, 3 body chapters and a conclusion. The introduction and conclusion are in their entirety written by myself. The three chapters that comprise the body of this thesis have been adapted from multiple research articles, with additional edits. All articles have been put on arXiv. Chapter 2 was a plenary talk at QIP 2023. Chapter 3 is currently undergoing submission to a journal.

My contribution to each is described below.

Chapter 2: A lower bound on the overhead of quantum error correction in low dimensions

Published in: arXiv:2302.04317 [preprint] (2023)

Author list: Nouedyn Baspin, Omar Fawzi, Ala Shayeghi

Contributions: I am the lead author on this article. I wrote an initial draft of the manuscript which was then edited and finalized in collaboration with my co-authors.

Chapter 3 and Chapter 4: Combinatorial structure in linear codes

Published in: 2309.16411 [preprint] (2023)

Author list: Nouedyn Baspin

Contributions:

I am the sole author.

Chapter 4: Layer codes

Published in: Nat Commun 15, 9528 (2024)

Author list: Dominic Williamson, Nouedyn Baspin

Contributions: The research was conducted jointly with Dominic Williamson. The idea came from trying to generalise 'Combinatorial structure in linear codes' to quantum codes. I came up with the original idea and sketch of the construction.

Abstract

This thesis was written with a specific objective in mind. As a subfield of applied mathematics, a lot of theoretical quantum computing consists of writing theorems. A lot of us, then, are frantically looking for the proof that would enshrine the months of effort that come with researching an idea.

In this adventure, intuition has been a singular ally. Having an idea of what a proof *should* look like before writing it can save a lot of time bushwhacking through uncertain territories. Finding a proof sketch the size of a few lines was often a lot more time consuming than unraveling the mathematical flourish that a full-sized paper demands.

In many way, this distinction is akin to differentiating 'melody' and 'harmonics'. Though the latter can change the texture or appearance of a song, its heart will always lie within its melody. The silliness of this hazardous metaphor aside, this thesis attempts to present the 'melody' of the works it is based on. We keep the mathematical verbosity to a minimum, in the hope it leaves enough room to express the intuition that hides behind it.

Each of the results presented in the next chapters was obtained after spending considerable time studying existing results in the literature, trying to extract their 'melody', and then attempting to iterate on it.

———

This thesis concerns itself with the role locality plays in the performance of quantum codes.

We first address the difficulty of producing codestates from relatively good codes, specifically we investigate the depth of geometrically local encoding circuits. Previous results struggled to produce bounds allowing for general, non-local classical communications in the circuit. We show that, owed to the entanglement hidden in codestates, the depth of encoding circuits is lower bounded, even when allowing for boundless classical computation.

We then move on to address a converse question: given a particular code, what can we say about its geometrical structure? We show that implementing stabiliser codes requires satisfying some connectivity requirements that depend on k and d. We quantify these requirements through the notion of graph expansion.

Finally, in the last chapter, we ask the question of the achievability of these requirements. The celebrated BPT bound states upper bound on the parameters that can be achieved for codes local in three dimensions. We provide the first construction that matches this bound.

Contents

1 | Introduction

Colloquially, error correction aims to recover information that might have been corrupted by some unknown events. This definition, naturally, begs some questions. What type of information do we wish to preserve? What are these events that we hope to be protected from?

Often, these can be flippantly answered. Most of this thesis will be concerned with *qubits* and *Pauli errors*. Nevertheless, the questions above hold singular insights when it comes to navigating the many languages of error correction. Newcomers often struggle building an intuition for stabilizer codes – to say nothing of subsystem codes[1], or Floquet codes[2]. Arguably, this confusion is not born of an immanent complexity to these constructs; rather it is not always obvious what information these objects encode, nor how.

1.1 Representing physical systems

Our first challenge is to describe what 'information' looks like in a physical system. From the perspective of an experimentalist, a physical system is something we can perform measurements on. Visualise, say, an apparatus that allows for a single quantity to be measured. We write that observable, Z, as a linear operator:

$$Z : |v\rangle \to |v'\rangle$$

The vectors $|v\rangle, |v'\rangle$ are arbitrary and belong to a (yet unspecified) vector space representing the state of the system. We will assume that, from experimental data, Z has outcomes $\{+1, -1\}$. A measurement when repeated should give the same result, which gives us the following constraint on Z:

$$Z^2 = (\pm 1)^2 = 1$$

Further, it only makes sense to measure Z if it is possible that its value changes. Formally, this is represented by an operator X, called a 'gate' such that:

$$ZX = -XZ$$

The action of X is to 'flip' the value of Z. For example if $|u_{-1}\rangle$ obeys $Z|u_{-1}\rangle = -1|u_{-1}\rangle$, then $ZX|u_{-1}\rangle = -XZ|u_{-1}\rangle = +1X|u_{-1}\rangle$. The vectors $|u_{-1}\rangle$ and $X|u_{-1}\rangle$ yield opposite values when Z is measured. This relationship between these two operators can be neatly summarised by the group commutator $[Z,X] = ZXZ^{-1}X^{-1} = -1$ which[1] can also be expressed as $ZX = [Z,X]XZ$.

We now have enough elements to describe what vector space Z and X are acting on. Observe that the following set

$$\mathcal{R}_1 = \{X, Z, 1, -1\}$$

is closed under the action of the commutator. One should think of $\mathcal{R}_1$ as capturing all the 'data' of the apparatus: it contains the observable (Z), the way we can modify the value of the observable (X), and how these two interact (the commutators) [2]. We call such a set the *set of relations*.

A set closed under the action of the commutator corresponds to the basis of a Lie algebra in representation theory. Although the invocation of Lie theory here may seem digressive, that theory is exactly concerned with the relationship between $\mathcal{R}_1$ and the vector space $\mathcal{V}_1$ that it acts on. For example, a natural choice of representation in our case is:

$$\mathcal{V}_1 = \mathbb{R}[\mathbb{F}_2], X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, 1 = \mathbb{1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

Where $\mathbb{R}[\mathbb{F}_2]$ denotes the group algebra of $\mathbb{F}_2$ over $\mathbb{R}$. This – somewhat simplistic – representation gives our experimentalist a complete mathematical model of their apparatus. Indeed, we are merely describing a *classical bit*. We have

$$\mathcal{V} = \text{span}\{|1\rangle, |-1\rangle\}, \quad Z|-1\rangle = -1|-1\rangle, \quad X|-1\rangle = |1\rangle$$

[1] Since $Z^2 = 1$, we have $Z^{-1} = Z$. Similarly for X: as flipping the value of Z twice returns the original value, it should be noted that $X^2 = 1$.

[2] The set $\mathcal{R}$ should be strikingly similar to the Heisenberg algebra. The only difference here being the bracket used: the Heisenberg algebra is defined according to the ring commutator: $[Q,P] = QP - PQ$. The type of bracket used depends heavily on the physics of the system of interest, in our case the group commutator is simply more well fitted to expressing the relationship between X and Z.

Moved by curiosity, our experimentalist might decide to study multiple bits at a time – let us a fix a number n. How then should we update our model in order to describe these n bits simultaneously? Following the above method, we need to start by elucidating what the associated set of relations $\mathcal{R}_n$ corresponds to. As remarked upon earlier, one could decompose $\mathcal{R}_1$ as

$$\mathcal{R}_1 = \mathcal{O}_1 \cup \mathcal{U}_1 \cup \text{commutators}$$

Where $\mathcal{O}_1 = \{Z\}$ is the set of observables, and similarly $\mathcal{U}_1 = \{X\}$ is the set[3] of gates. When dealing with n bits, we can act on, and measure several bits independently. This suggests having access to sets $\{Z_i\}_{i=1}^{i=n}, \{X_j\}_{j=1}^{j=n}$ of operators obeying the following properties:

1. $[Z_i, X_j] = -1$ if $i = j$, and $[Z_i, X_j] = 1$ if $i \neq j$
2. $[Z_i, Z_{i'}] = 1$
3. $[X_j, X_{j'}] = 1$

The first property ensures that only X_i can change the value of Z_i – the two sets are dual under the group commutator, more concisely we have $[Z_i, X_j] = (-1)^{\delta_{i,j}}$. The second one expresses that measuring a bit does not affect the measurement of another one. The last one, similarly, ensures that acting on a bit does not induce non-local effects.

It is not unreasonable to also demand operations of the form $X_j X_{j'}$ to belong to $\mathcal{U}_n$: it reflects the ability of the experimentalist to be able to act on multiple bits simultaneously. Similarly, products of the form $Z_i Z_{i'}$ reflect the ability to measure the value of several bits at a time. Under these conditions, one can express $\mathcal{O}_n$, $\mathcal{U}_n$ and R_n as[4]

$$\mathcal{O}_n = \{\mathbb{1}, Z\}^{\otimes n}, \quad \mathcal{U}_n = \{\mathbb{1}, X\}^{\otimes n}, \quad \mathcal{R}_n = \mathcal{O}_n \cup \mathcal{U}_n \cup \{\mathbb{1}, -\mathbb{1}\}$$

As a notational convenience, we can also write $\mathcal{O}_n = \langle Z_1, \ldots, Z_n \rangle$ and $\mathcal{U}_n = \langle X_1, \ldots, X_n \rangle$, with $\langle \cdot \rangle$ the usual notation for a generating set. The operators in $\mathcal{R}_n$ act on $\mathcal{V}_n = \mathcal{V}_1^{\otimes n} = (\mathbb{R}[\mathbb{F}_2])^{\otimes n} = (\mathbb{R}^2)^{\otimes n}$ – which corresponds to a natural model for a system of n bits. The

[3] In fact, we could elect an alternative definition: $\mathcal{U}_1 = \{\mathbb{1}, X\}$ which would not change the resulting $\mathcal{R}$, and would have a coherent physical interpretation. In that case $\mathcal{U}$ is closed under multiplication, contains the identity, and each element is invertible, and becomes a group. To preserve duality, one could also update $\mathcal{O}$ to $\{\mathbb{1}, Z\}$.

[4] This way we also ensure that $\mathcal{O}_n$ and $\mathcal{U}_n$ are dual with respect to the commutator: $\mathcal{O}_n \cong \mathcal{U}_n \cong \mathbb{F}_2^n$.

space $\mathcal{V}_n$ can be taken as a model for n bits *because* it hosts a representation of the allowed observables and operations. The observables and operations dictate the actual degrees of freedom of the system, from which a mathematical representation in terms of vector spaces is derived. The core observations of this section can be summarised as follows:

The representation of the Lie algebra of $\mathcal{R}$ gives the experimentalist a formal way to model the physical system. The set $\mathcal{R}$ captures the observables, the operations, and the interaction between the two; i.e. the information stored in the system.

1.2 Linear codes

One day, while setting up an experiment, our experimentalist might unknowingly introduce a preparation error. Instead of producing the expected state $|v\rangle$, they might be operating on the state $X_i|v\rangle$, for some i. This type of error is undesirable since the outcome of subsequent measurements becomes unreliable. The way *codes* attempt to solve this issue is by identifying $\mathcal{R}' \subset \mathcal{R}_n$ such that $\mathcal{R}'$ is, in some sense, insensitive to noise. A *linear* code further entails that $\mathcal{R}'$ be isomorphic to $\mathcal{R}_k$ for some $k < n$. In a sense, we are attempting to find a way to hide k virtual bits, within n bits, in a way that isolates these k bits from errors. Such a subsystem can be written as – abstracting the ± 1 operators for convenience

$$\mathcal{R}' = \begin{matrix} \langle U_1 & U_2 & \dots & U_k \rangle \\ \langle O_1 & O_2 & \dots & O_k \rangle \end{matrix}$$

This notation should be read as $\mathcal{R}' = \mathcal{U}' \cup \mathcal{O}'$, with $\mathcal{U}' = \langle U_1, U_2, \dots, U_k \rangle$, and $\mathcal{O}' = \langle O_1, O_2, \dots, O_k \rangle$; such that $[U_i, O_j] = (-\mathbb{1})^{\delta_{i,j}}$. This last condition preserves the commutation relations expected of $\mathcal{R}_k$, and reflects the isomorphism $U_i \leftrightarrow X_i$, $O_j \leftrightarrow Z_j$.

As an example, a common assumption is that errors affecting many bits are much less common that errors affecting few bits. We can then pick an arbitrary cut-off number d, and choose $\mathcal{R}'$ such that $\mathcal{U}'$ contains no element $E \in \langle U_1, U_2, \dots, U_k \rangle$ with weight[5] less than d. Effectively, to act on these k virtual bits, one needs to act on at least d physical bits thus safeguarding them from the most likely errors.

The effective usage of such a scheme, so far, remains unclear. We have picked a virtual subsystem that is supposedly immune to small errors, but we are yet to describe how to correct, or even detect, them. Continuing our discussion requires

[5] An operator $E \in \mathcal{R}_n$ is said to have weight d if it is a permutation of some $E' \in \mathcal{R}_d \otimes \mathbb{1}_{n-d}$.

introducing some further information about linear codes. Whenever $\mathcal{R}' \cong \mathcal{R}_k$, then it is possible[6] to find $n-k$ pairs $(U_{k+1}, O_{k+1}), \ldots, (U_n, O_n) \subset \mathcal{U} \times \mathcal{O}$ such that

$$\mathcal{R}' \otimes \mathcal{R}'' = \begin{matrix} \langle U_1 & U_2 & \ldots & U_k \rangle \\ \langle O_1 & O_2 & \ldots & O_k \rangle \end{matrix} \otimes \begin{matrix} \langle U_{k+1} & U_{k+2} & \ldots & U_n \rangle \\ \langle O_{k+1} & O_{k+2} & \ldots & O_n \rangle \end{matrix} \cong \mathcal{R}_n$$

This additional structure will be instrumental in operating the code corresponding to $\mathcal{R}'$. Let us focus on an error $E \in \{\mathbb{1}, X\}^{\otimes n}$ acting on strictly less than d bits. As $\mathcal{R}_n \cong \mathcal{R}' \otimes \mathcal{R}''$, then the error can be decomposed into $E = E_k E_{n-k}$, where $E_k \in \mathcal{R}' \otimes \mathbb{1}$, and $E_{n-k} \in \mathbb{1} \otimes \mathcal{R}''$. Because E_k acts on more than d bits, the assumption that E acts on less than d bits guarantees that $E_{n-k} \neq \mathbb{1}$: E has to 'spill' out of $\mathcal{R}'$.

We can use this to our advantage: the spill can be detected. Due to the duality between $\mathcal{O}_n$ and $\mathcal{U}_n$, there exists $S \in \mathcal{O}_{n-k}$ that can detect E_{n-k}, or equivalently, $[S, E_{n-k}] = -\mathbb{1}$. Further, by construction we have $[E_k, \mathcal{O}_{n-k}] = \mathbb{1}$, so $[E, S] = [E_{n-k}, S] = -\mathbb{1}$. One can then verify that if we were to measure the observables $\{O_{k+1}, \ldots, O_n\}$, at least one would see its value flipped by E. To detect errors, our experimentalist can then simply start in a state where $O_i = 1$, for all $i \in [n-k, \ldots, n]$, and keep track of their values across time. Whenever an error of weight less than d affects the system, the value of at least one O_i will flip to -1. The set of observables whose value is flipped by E is called the *syndrome* of E.

Once the spill is detected, then starts the process of correcting the error. To each error E of of sufficiently small weight, we want to associate a correction F depending only on the syndrome, and such that FE has no syndrome. The correction succeeds when $FE = \mathbb{1}$, and fails when $FE \in \mathcal{R}'$ – i.e. we end up acting on subsystem. We leverage the fact that F only needs to act to at most as many bits as E acts on: as long as E acts on at most $\lfloor \frac{d}{2} \rfloor$ bits, FE acts on less than d bits, and does not belong to the logicals. A code of distance d can correct any error of size less than $\lfloor \frac{d}{2} \rfloor$.

Just as the code $\mathcal{R}_k$ determines the observables $\{O_{k+1}, \ldots, O_n\}$, the converse is true. Often, to refer to a code, we will reference its stabilisers. Also called *stabilisers*, these operators make for a concise description of a code. We formalise this connection in Theorem 1.

[6] See Theorem 1.

Theorem 1. Let $\mathcal{R}' \subset \mathcal{R}_n$ such that

$$\mathcal{R}' = \begin{matrix} \langle U_1 & U_2 & \dots & U_k \rangle \\ \langle O_1 & O_2 & \dots & O_k \rangle \end{matrix}$$

then there exists a set of stabilisers $\{O_{k+1}, \dots, O_n\}$, and a set $\{U_{k+1}, \dots, U_n\}$ such that

$$\mathcal{R}_n = \begin{matrix} \langle U_1 & U_2 & \dots & U_k & U_{k+1} & U_{k+2} & \dots & U_n \rangle \\ \langle O_1 & O_2 & \dots & O_k & O_{k+1} & O_{k+2} & \dots & O_n \rangle \end{matrix}$$

Conversely, given a set of stabilisers $\{O_1, \dots, O_r\}$, and a dual set $\{U_1, \dots, U_r\}$, then there exists

$$\mathcal{R}' = \begin{matrix} \langle U_{r+1} & U_{r+2} & \dots & U_n \rangle \\ \langle O_{r+1} & O_{r+2} & \dots & O_n \rangle \end{matrix}$$

such that

$$\mathcal{R}_n = \begin{matrix} \langle U_1 & U_2 & \dots & U_r & U_{r+1} & U_{r+2} & \dots & U_n \rangle \\ \langle O_1 & O_2 & \dots & O_r & O_{r+1} & O_{r+2} & \dots & O_n \rangle \end{matrix}$$

Proof. Corollary of Lemma 1. □

1.3 Quantum codes

Quantum theory differs from classical theory by the utterly singular fact that actions become observables and vice versa. As a consequence, $X \in \mathcal{O}$, and $Z \in \mathcal{U}$. The operator XZ – or $iXZ \equiv Y$ to obtain real eigenvalues – becomes an observable, and an action. In a way, the transition from classical to quantum can be summarized by $\mathcal{O} = \mathcal{U}$. The set $\mathcal{R}$ is updated in the following fashion:

$$\mathcal{R} = \mathcal{O} = \mathcal{U} = \{\mathbb{1}, X, Y, Z\}^{\otimes n}$$

These operators (X, Y, and Z) correspond to the usual Pauli operators, and provide us with an opportunity to formally introduce the Pauli group.

Definition 1. The Pauli group on n qubits is a set of complex matrices acting on $(\mathbb{C}^2)^n$ defined as

$$\mathcal{P}_n = \{\mathbb{1}, X, Y, Z\}^{\otimes n}$$

The update of the relation set translates into a need for a new representation. We now have $\mathcal{V}_n = (\mathbb{C}^2)^{\otimes n}$ hosting the representation of these quantum operations. Despite the inclusion of a phase i in the definition of Y, the interactions between operators in

$\mathcal{R}$ are still phase-independent: for any $P, P' \in \mathcal{R}$, we have that $[P, P'] = [P, iP']$. For the sake of convenience we will thus often abstract phases away. This convention allows for the following compact representation:

$$\mathcal{R} = \left\langle \begin{matrix} X_1 & X_2 & \dots & X_n \\ Z_1 & Z_2 & \dots & Z_n \end{matrix} \right\rangle$$

The columns, once again, reflect the commutation relations of thee observable – $[X_i, Z_j] = (-\mathbb{1})^{\delta_{i,j}}$ – but the angle brackets encompasses both sets of X's and Z's to reflect the new structure of $\mathcal{R}$. Given familiarity with classical linear codes, the structure of quantum linear codes should note be surprising. We are indeed looking for sets $\mathcal{R}', \mathcal{R}'' \subset \mathcal{R}_n$, such that [7]

$$\mathcal{R}' = \left\langle \begin{matrix} P_1 & P_2 & \dots & P_k \\ Q_1 & Q_2 & \dots & Q_k \end{matrix} \right\rangle, \ \mathcal{R}'' = \left\langle \begin{matrix} P_{k+1} & P_{k+2} & \dots & P_n \\ Q_{k+1} & Q_{k+2} & \dots & Q_n \end{matrix} \right\rangle$$

$$\mathcal{R}' \otimes \mathcal{R}'' = \left\langle \begin{matrix} P_1 & P_2 & \dots & P_k \\ Q_1 & Q_2 & \dots & Q_k \end{matrix} \right\rangle \otimes \left\langle \begin{matrix} P_{k+1} & P_{k+2} & \dots & P_n \\ Q_{k+1} & Q_{k+2} & \dots & Q_n \end{matrix} \right\rangle = \mathcal{R}_n$$

There is nevertheless a stunning difference from the classical case. Out of the correspondence $\mathcal{O} = \mathcal{U}$ emerges a bespoke requirement: *all* of the elements of $\mathcal{R}'$ are required to act on many qubits, instead of merely those of $\langle P_1, P_2, \dots, P_k \rangle$. This characteristic of quantum codes will have significant ramifications in later chapters. Yet, as far their algebraic structure is concerned, little changes. The experimentalist will be able to identify and correct all errors acting on at most $\lfloor \frac{d}{2} \rfloor$ qubits, via measuring the operators $\{Q_{k+1}, \dots, Q_n\}$. In the context of quantum codes, these operators are the *stabilisers* of the code; and $\mathcal{Q} = \langle Q_{k+1}, \dots, Q_n \rangle$ is the *stabiliser group*.

To formalise the connection between the stabilisers and the code, we will exploit the following result by Yoshida and Chuang [3] – they refer to this result as "Framework I", see Section I-C "Canonical Representation".

Lemma 1 ([3])**.** Let $\mathcal{T} \subset \mathcal{R}_n$, and $a + b \leq n$ such that:

$$\mathcal{T} = \left\langle \begin{matrix} P_1 & \dots & P_a & & & \\ Q_1 & \dots & Q_a & Q_{a+1} & \dots & Q_{a+b} \end{matrix} \right\rangle \tag{1.1}$$

[7] We use Q/P in our notation to reflect the commutation relations without referencing an obsolete gate/observable distinction.

Then there exist $\{P_{a+1},\dots,P_n\} \subset \mathcal{R}_n$ and $\{Q_{a+b+1},\dots,Q_n\} \subset \mathcal{R}_n$ such that

$$\left\langle \begin{matrix} P_1 & \dots & P_a & P_{a+1} & \dots & P_{a+b} & P_{a+b+1} & \dots & P_n \\ Q_1 & \dots & Q_a & Q_{a+1} & \dots & Q_{a+b} & Q_{a+b+1} & \dots & Q_n \end{matrix} \right\rangle = \mathcal{R}_n \tag{1.2}$$

We now can state the duality between stabilisers and codespace.

Theorem 2. Let $\mathcal{R}' \subset \mathcal{R}_n$ such that

$$\mathcal{R}' = \left\langle \begin{matrix} P_1 & P_2 & \dots & P_k \\ Q_1 & Q_2 & \dots & Q_k \end{matrix} \right\rangle$$

then there exists a set of stabilisers $\{Q_{k+1},\dots,Q_n\}$, and a set $\{P_{k+1},\dots,P_n\}$ such that

$$\mathcal{R}_n = \left\langle \begin{matrix} P_1 & P_2 & \dots & P_k & P_{k+1} & P_{k+2} & \dots & P_n \\ Q_1 & Q_2 & \dots & Q_k & Q_{k+1} & Q_{k+2} & \dots & Q_n \end{matrix} \right\rangle$$

Conversely, given a set of stabilisers $\{Q_1,\dots,Q_r\}$, and a dual set $\{P_1,\dots,P_r\}$, then there exists

$$\mathcal{R}' = \left\langle \begin{matrix} P_{r+1} & P_{r+2} & \dots & P_n \\ Q_{r+1} & Q_{r+2} & \dots & Q_n \end{matrix} \right\rangle$$

such that

$$\mathcal{R}_n = \left\langle \begin{matrix} P_1 & P_2 & \dots & P_r & P_{r+1} & P_{r+2} & \dots & U_n \\ Q_1 & Q_2 & \dots & Q_r & Q_{r+1} & Q_{r+2} & \dots & O_n \end{matrix} \right\rangle$$

Proof. Corollary of Lemma 1. □

With this algebraic structure in place, we can now provide the definition of a stabiliser code.

Definition 2. Let $\mathcal{Q}$ be a subgroup of commuting Pauli operators. Then the associated code $\mathcal{C}$ is the subspace

$$\mathcal{C} = \{|\psi\rangle : Q|\psi\rangle = |\psi\rangle, \forall Q \in \mathcal{Q}\}$$

The code $\mathcal{C}$ is said to have k logicals if $\mathcal{Q}$ is generated by $n-k$ independent operators. The operators $P_1 \dots P_k$ given by Theorem 2 are often referred to as $\bar{X}_1,\dots,\bar{X}_k$ to emphasize the isomorphism with the Pauli operators. Similarly for $Q_1 \dots Q_k$ and $\bar{Z}_1,\dots,\bar{Z}_k$.

1.4 Correctability and Indistinguishability

The existence of a set $\mathcal{R}$ with large distance d results in a system whose information is delocalised, inaccessible locally. If the distance is large enough then acting on a small set of qubit cannot corrupt the encoded information.

Definition 3 (Error-free set)**.** For a code $\mathcal{C}$ on n qubits, a set $A \subset [n]$ is said to be error-free if, for any Pauli operator $P \in \mathcal{P}^A$ we have:

$$\Pi P \Pi = c_P \Pi$$

where Π is the projector onto the codespace, and c is a constant depending only on $\mathcal{C}$ and P. By definition any set of size strictly less than d is error-free.

This definition captures which operators meaningfully affect the codespace. For example, a stabiliser or a *detectable* error both satisfy $\Pi P \Pi = c\Pi$. Some might find this definition limiting nonetheless. Admittedly the information is not corrupted, but it is not obvious that it can be recovered. Once an error is detected it might not be trivial to work out the appropriate correction. For this reason we are motivated to introduce another definition; which encapsulate a different aspect of 'information delocalisation'.

Definition 4 (Correctable set)**.** For a code $\mathcal{C}$ on n qubits, a set $A \subset [n]$ is said to be error-free if there exists a CPTP channel $\mathcal{T}$ such that

$$\mathcal{T}(\mathrm{tr}_A(|\psi\rangle\langle\psi|)) = |\psi\rangle\langle\psi|, \forall\, |\psi\rangle \in \mathcal{C}$$

'Correctability' is a particularly strong guarantee: the set A is, in fact, completely superfluous to the recovery of the information in the system. Regardless of the error A has undergone, one can just as well discard it entirely and still recover the information the code holds. Interestingly, these two definitions turn out to be equivalent.

Lemma 2 ([4],[5] Corollary of Thm 10.1)**.** Let $\mathcal{C}$ be a quantum code, and $A \subset [n]$ a subset of qubits. Then we have

$$\Pi P^\dagger P' \Pi = c_{P,P'} \Pi$$

for every pair $P, P' \in \mathcal{P}_A$ if and only if there exists a channel $\mathcal{T}$ such that

$$\mathcal{T}(\mathrm{tr}_A(|\psi\rangle\langle\psi|)) = |\psi\rangle\langle\psi|, \forall\, |\psi\rangle \in \mathcal{C}$$

Given that $P^\dagger P' = P'' \in \mathcal{P}_A$, then Definition 3 clearly implies Definition 4. Conversely, if $\Pi P P' \Pi = c_{P,P'}\Pi$, then $\Pi P'' \Pi = c_{''}\Pi$; thus the two conditions are equivalent.

Theorem 3. Definition 4 and 3 are equivalent.

The equivalence of these two definitions is also not very hard to establish in the classical case when the general Paulis are replaced with X-type Paulis [8]. There is one distinctively quantum property however, called *indistinguishability*. To highlight its importance we start by definition the notion of a 'no-information' set.

Definition 5 (No-information set)**.** For a code $\mathcal{C}$ a subset $A \subset [n]$ is called a no-information set, if for every $|\psi_0\rangle, |\psi_1\rangle \in \mathcal{C}$, we have

$$\mathrm{tr}_{\bar{A}} |\psi_0\rangle\langle\psi_0| = \mathrm{tr}_{\bar{A}} |\psi_1\rangle\langle\psi_1|$$

In short, having access to only such a set, one can say nothing of the encoded logical information. Quantum information is singular in that a correctable set is necessarily a no-information set.

Lemma 3 (Indistinguishability)**.** A correctable set is a non-information set.

Proof. We will make use of the fact that any complex matrix M on n qubits can be expressed as

$$M = \sum_{P \in \mathcal{P}_n} P \,\mathrm{tr}(PM)$$

[8] By that we mean operators belonging to the set $\{\mathbb{1}, X\}^{\otimes n}$.

We thus write

$$\begin{aligned}
\mathrm{tr}_{\overline{A}}\left|\psi_0\right\rangle\!\left\langle\psi_0\right| - \mathrm{tr}_{\overline{A}}\left|\psi_1\right\rangle\!\left\langle\psi_1\right| &= \sum_{P\in\mathcal{P}_A} P\,\mathrm{tr}\Big(P(\mathrm{tr}_{\overline{A}}\left|\psi_0\right\rangle\!\left\langle\psi_0\right| - \mathrm{tr}_{\overline{A}}\left|\psi_1\right\rangle\!\left\langle\psi_1\right|)\Big) \\
&= \sum_{P\in\mathcal{P}_A} P\,\mathrm{tr}\Big(P(\left|\psi_0\right\rangle\!\left\langle\psi_0\right| - \left|\psi_1\right\rangle\!\left\langle\psi_1\right|)\Big) \\
&= \sum_{P\in\mathcal{P}_A} P\,\mathrm{tr}\Big(\Pi P\Pi(\left|\psi_0\right\rangle\!\left\langle\psi_0\right| - \left|\psi_1\right\rangle\!\left\langle\psi_1\right|)\Big) \\
&= \sum_{P\in\mathcal{P}_A} P c_P\,\mathrm{tr}\Big(\Pi(\left|\psi_0\right\rangle\!\left\langle\psi_0\right| - \left|\psi_1\right\rangle\!\left\langle\psi_1\right|)\Big) \\
&= \sum_{P\in\mathcal{P}_A} P c_P\,\mathrm{tr}\Big(\left|\psi_0\right\rangle\!\left\langle\psi_0\right| - \left|\psi_1\right\rangle\!\left\langle\psi_1\right|\Big) \\
&= \sum_{P\in\mathcal{P}_A} P c_P(1-1) \\
&= 0
\end{aligned}$$

Where, from the third to the fourth line we used the fact that a correctable set is an error-free set. □

This property of quantum codes is unique and deserves pause. By contrast, given a repetition code it suffices to read a single bit of the codeword to determine the information stored – while its distance is expansive. In hindsight, this behaviour is not surprising. Given that $\mathcal{O} = \mathcal{U}$, then distinguishability corresponds exactly to an operation, and thus, an error. This is exactly what we leveraged: $\Pi P\Pi \propto \Pi$, holds for both Pauli observables and operations. In the classical realm, the sum in the proof of Lemma 3 would involve operators for which $\Pi P\Pi \propto \Pi$ does not hold. Put more concisely, the indistinguishability property of quantum codes is a direct consequence of $\mathcal{O} = \mathcal{U}$.

1.5 Entanglement

A quantum state shared between systems A and B is said to be entangled when $\left|\psi\right\rangle_{AB} \neq \left|\psi_A\right\rangle \otimes \left|\psi_B\right\rangle$. The existence of such states is intrinsic to quantum theory, and the hallmark of non-classical physics. Here we wish to argue that entanglement is a

core property of quantum codes and their functioning, which derives directly from $\mathcal{O} = \mathcal{U}$ by way of the indistinguishability principle.

Despite how deceptively easy it is to define entanglement, quantifying it is an inextricable puzzle. Multiple, equally well-motivated measures frequently yield different quantities. Faced with this abundance of options, the best way through this haystack is often, simply, to pick the measure that is the most well-behaved in a given context. Under this prescription, we adopt the following definition.

Definition 6 (Qubits of entanglement)**.** We say that a pure state $\left|\psi\right\rangle_{AB}$ has m qubits of entanglement if there exists unitaries U_A, U_B such that

$$U_A \otimes U_B \left|\psi\right\rangle_{AB} = \left|\psi'_A\right\rangle \otimes |\Phi\rangle^{\otimes m} \otimes \left|\psi'_B\right\rangle$$

With $|\Phi\rangle$ the Bell state

$$|\Phi\rangle = \frac{|0\rangle_A |0\rangle_B + |1\rangle_A |1\rangle_B}{\sqrt{2}}$$

The Bell state is occasionally called an 'ebit', as in a bit of entanglement. The well known Bell teleportation protocol indeed asserts that a qubit can be teleported from A to B if they both share a Bell pair [6]. Due to the linear structure of stabilizer codes, it is remarkably easy to compute exactly how many Bell pairs they contain.

Lemma 4. Let $\left|\psi\right\rangle \in \mathcal{C}$, and $A \subset [1,\ldots,n]$ a correctable subset of qubits. Then $\left|\psi\right\rangle_{AB}$ has m qubits of entanglement where

$$m = |A| - |\mathcal{Q}_A|$$

With $|\mathcal{Q}_A|$ is the rank of the subgroup of the stabilizer group with support contained in A

Proof. We start by writing $\rho = \left|\psi\right\rangle\!\left\langle\psi\right|$. In [7] the authors compute the entanglement by analysing $\rho_A = \mathrm{tr}_B(\rho)$. In general, we do not expect ρ_A to have a lot of structure. For this reason, we invoke the Indistinguishability Lemma 3: $\rho_A = \sigma_A, \sigma = \frac{1}{2^k}\Pi$; where Π is the projector on $\mathcal{C}$. We remind the reader that this projector can be written as:

$$\Pi = \prod_{i=1}^{i=r} \frac{1}{2}(\mathbb{1} + Q_i)$$

Indeed, splitting the Hilbert space into $\mathcal{H} = \mathcal{C} \oplus \mathcal{C}^\perp$, one sees that $\Pi\mathcal{C}^\perp = \mathcal{C}^\perp\Pi = 0$, while it acts as the identity on $\mathcal{C}$. We chose σ to be a very specific state: the maximally

mixed state on $\mathcal{C}$, whose clear structure can be now leveraged. This allows us to obtain an explicit expression for σ_A:

$$\begin{aligned}\sigma_A &= \mathrm{tr}_B\left(\frac{1}{2^k}\prod_{i=1}^{i=r}\frac{1}{2}(\mathbb{1}+Q_i)\right)\\ &= \mathrm{tr}_B\left(\frac{1}{2^n}\prod_{i=1}^{i=r}(\mathbb{1}+Q_i)\right)\\ &= \frac{1}{2^{n-n_B}}\,\mathrm{tr}_B\left(\sum_{Q\in\mathcal{Q}} Q\right)\\ &= \frac{1}{2^{n_A}}\sum_{Q:\mathrm{supp}(Q)\subset A} Q\end{aligned}$$

As we have now elucidated the structure of σ_A, and thus ρ_A, the main result of [7] (Equation 5 and Equation 8) gives $m = |A| - |\mathcal{Q}_A|$. □

We would like to draw attention to how the proof of Lemma 4 unfolds. Our choice for the proxy state σ could come across as more idiosyncratic than it actually is. Note that σ is a uniform mixture of 2^k codestates, which characterises just how big our code is. This size happens to be captured by $k = n - |\mathcal{Q}| = n - r$, see Theorem 2. If one defines $H(A) = |A| - |\mathcal{Q}_A|$, we obtain a way to connect a global property ($H(AB) = k$) to a local one ($H(A) = m$).

Further, $H(A)$ is subadditive: since $\mathcal{Q}_A \cdot \mathcal{Q}_B \subset \mathcal{Q}_{AB}$, then $H(AB) \leq H(A) + H(B)$. Combining these elements, we see that $k = n - |\mathcal{Q}| = H(AB) \leq H(A) + H(B) = m + H(B)$. By picking the appropriate proxy state, we uncover a relationship between the amount of encoded information k, and the entanglement encoded in a subregion m. If we manage to control what $H(B)$ is, then there is an avenue towards a crisp expression of the k-m relation.

The next theorem is an attempt to provide such an expression. Summarily, it conveys the idea that if k is high, then the code contains at least one region where $H(A)$ is high; resulting in a least one highly entangled region.

Theorem 4. Let $\mathcal{C}$ be a stabiliser code with parameters $[[n,k,d]]$; and a partition $[n] = \bigsqcup_{i=1}^{i=l} \Lambda_i$ of the qubits, such that $|\Lambda_i| < d$. Then there exists i_* such that $|\psi\rangle_{\Lambda_{i_*}\overline{\Lambda_{i_*}}}$ has at least $\frac{k}{l}$ qubits of entanglement, for any $|\psi\rangle \in \mathcal{C}$.

Proof. We almost only leverage the subadditivity of H

$$\begin{aligned} k = H(\Lambda) \leq & \sum_i H(\Lambda_i) \\ \leq & l \max_i H(\Lambda_i) \\ \leq & l \cdot H(\Lambda_{i_*}) \end{aligned}$$

Finally, since Λ_{i_*} is correctable, then $H(\Lambda_{i_*}) = m$. □

Note that in general we can take $l \propto \frac{n}{d}$ as the number of partition. In which case, Theorem 4 guarantees $H(\Lambda_{i_*}) \gtrsim \frac{kd}{n}$

1.6 Same but different

In our previous section the function $H(A)$ played a crucial role in our analysis, and eventually allowed us to derive Theorem 4. While writing that section, we refrained from mentioning that $H(A)$, in fact, corresponds to the entropy for a stabilizer state (also see [7], Equation 6).

In this section we will reformulate that argument from a strictly and explicitly entropic viewpoint. We do not wish to merely rephrase the previous result, but instead to highlight that by using slightly different tools the same argument can deliver a more general result, and a 'cleaner' proof. By describing two different proofs of the same idea, we hope to provide a morsel of a Rosetta stone when it comes to the many languages of QEC [9].

For a state ρ the standard Von Neumann entropy is expressed as:

$$S(A)_\rho = \mathrm{tr}(\rho_A \log(\rho_A))$$

Admittedly, this expression is pretty opaque. What does it tell us about ρ_A? What does it mean? It is often said that the entropy characterises how "compressible" a state is. For example, there exists an isometry that maps the 3-qubit state $\rho_3 = \frac{1}{2}|000\rangle\langle 000| + \frac{1}{2}|111\rangle\langle 111|$ to the 1-qubit state $\rho_1 = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|$. Owed to the isometric nature of the transformation, it is reversible; and as it happens, both states have the same entropy. It might be tempting to then expect a statement of the following form to hold:

[9] The entropy $S(A)_\rho$ constitutes the main object of interest of Quantum Shannon Theory (QST), is very seldom encountered in QEC. The lack of collaboration between the two fields is rather striking. QST concerns itself precisely with information recovery and noisy processing.

For a state ρ on n qubits, and an integer $r < n$, there exists an isometry from $(\mathbb{C}^2)^n$ to $(\mathbb{C}^2)^r$ if and only if the entropy of ρ is less than r.

It is indeed *too* tempting[10]. Therein lies a lot of the oddity of the entropy: it only characterises the compressibility of ρ in the asymptotic limit. The correct statement reads instead [10]:

For a state ρ on n qubits, and an integer $r < n$, then in the limit $m \to \infty$, $\rho^{\otimes m}$ can be compressed into $r \cdot m$ qubits if and only if the entropy of ρ is less than r.

Why spend so much time splitting hair about what the entropy means? We want to highlight that the entropy is as mysterious as it appears. One can certainly intuit that the smaller the entropy, the more compressible a state is, and the less it contains "information"; however this rule of thumb only holds true to some extent. Why then use that definition? There are a plethora of other entropies, some with a much more direct operational meaning [11] which could be more appealing. Unfortunately, it seems that what we gain in granularity, we lose in structure: these alternative entropies are often less well-behaved and exhibit somewhat idiosyncratic behaviours. As a concrete example, the 0-th Rényi entropy relates directly to the rank of a state: $S_0(\rho) = \log \operatorname{rank} \rho$, a quantity that can be very easily understood. In spite of this natural definition, and contrary to the Von Neumann entropy, A_0 does not satisfy the so-called strong subadditivity property[12]: $S_0(\rho_A) + S_0(\rho_{ABC}) \not\leq S_0(\rho_{AB}) + S_0(\rho_{AC})$. This seemingly innocuous property has far ranging consequences and allows many, many computations in Shannon theory to be simplified. The point we are trying to drive at is that the Von Neumann entropy, notwithstanding its esoteric definition, is remarkably well behaved. Its meaning is unintuitive, but the way it transforms very much is. In essence, we use the Von Neumann entropy here because it makes calculations easy.

We now move on to recreate the main result from the previous section, while instead relying on the entropy. We start by showing that $S(A) = H(A)$ for the state $\sigma = \frac{1}{2^k}\Pi$. This will serve as motivation for the rest of this section: instead of relating entanglement to $|A| - |\mathcal{Q}_A|$, we can instead connect it to the entropy; and this connection is well motivated since both quantities are equal for the maximally mixed state.

[10] There exist entropies with an operational meaning even at finite scale, most notably the min-entropy and max-entropy [8]. They respectively capture the 'minimum' and the 'maximum' information that can be gained from performing a measurement on the state. By contrast, the usual entropy can be understood as the *average* of the information [9].

Remember that $\sigma_A = \frac{1}{2^{n_A}} \sum_{Q:\mathrm{supp}(Q)\subset A} Q$; this allows us to calculate

$$\log\left(\frac{1}{2^{n_A}} \sum_{Q:\mathrm{supp}(Q)\subset A} Q\right) = -n_A + \log\left(\sum_{Q:\mathrm{supp}(Q)\subset A} Q\right)$$

As $\mathrm{supp}(Q) \subset A$, then $\sigma_A Q = \sigma_A \cdot 1$. This gives

$$S(A)_\rho = -\mathrm{tr}\left(\sigma_A \log(\sigma_A)\right) = n_A - \log\left(\sum_{Q:\mathrm{supp}(Q)\subset A} 1\right) = n_A - |\mathcal{Q}_A| \tag{1.3}$$

Note how σ has as much entropy as possible for a state living supported on $\mathcal{C}$: k bits of entropy. The bigger k is, the more entropy σ has. This entropy has to reside *somewhere*, and from this indistinguishability principle, if a small enough subset of qubits has a lot of entropy, then it always has a lot of entropy regardless of the state. Much like $|A| - |\mathcal{Q}_A|$ could easily be related to the amount of qubits of entanglement hosted by A, we need a way to somehow relate $S(A)_\rho$ to the entanglement of the state ρ. We choose the coherent information.

Definition 7. For a state ρ defined on the space $\mathcal{H}_A \otimes \mathcal{H}_{\overline{A}}$, the coherent information $I(A\rangle\overline{A})$ is defined as

$$I(A\rangle\overline{A})_\rho = S(\overline{A})_\rho - S(A\overline{A})_\rho$$

Similarly to the entropy, the coherent information is fundamentally an asymptotic metric; it captures how many Bell pairs can be extracted from the state $\rho^{\otimes n}$, as $n \to \infty$ [13]. In that sense its immediate use it limited. Nevertheless, it inherits the good behaviour of the entropy; and it is straightforward to relate it with $S(A)_\rho$.

Lemma 5 (Entropic version of Lemma 4, [14])**.** Let $|\psi\rangle \in \mathcal{C}$, and $A \subset [1,\dots,n]$ a correctable subset of qubits. Then $I(A\rangle\overline{A})_{|\psi\rangle\langle\psi|} = S(A)_{|\psi\rangle\langle\psi|}$

With these tools in hand, we can finally prove an entropic version of Theorem 4.

Theorem 5. Let $\mathcal{C}$ be a code with parameters $[[n,k,d]]$; and a partition $[n] = \bigsqcup_{i=1}^{i=l} \Lambda_i$ of the qubits, such that $|\Lambda_i| < d$. Then there exists i_* such that $I(\Lambda_{i_*}\rangle\overline{\Lambda_{i_*}})_{|\psi\rangle\langle\psi|} \geq \frac{k}{l}$, for any $|\psi\rangle \in \mathcal{C}$.

Proof. The proof follows identically to that of Theorem 4, as the entropy also obeys subadditivity [15], i.e. $S(AB) \leq S(A) + S(B)$.

$$\begin{aligned} k &= S(\Lambda)_\sigma \\ &\leq \textstyle\sum_i S(\Lambda_i)_\sigma = \sum_i S(\Lambda_i)_{|\psi\rangle\langle\psi|} &&= \textstyle\sum_i I(\Lambda_i\rangle\overline{\Lambda_i})_{|\psi\rangle\langle\psi|} \\ &&&\leq l \cdot \max_i I(\Lambda_i\rangle\overline{\Lambda_i})_{|\psi\rangle\langle\psi|} \end{aligned}$$

The second equal sign on the second line uses Lemma 3. The last equal sign on that line uses Lemma 5. □

The reader might question the relevance of an entropic formulation. First, Theorem 4 relies on the assumption that $\mathcal{C}$ is a stabiliser code. This condition is non-trivial, as many codes are non-stabiliser in nature. Secondly, entropy provides an extensive and malleable framework that allows us to easily probe the entanglement of a code and its consequences. We illustrate this claim in the next chapter.

1.7 Open questions

The main purpose of this introductory chapter was to demonstrate some of the particularly quantum features of QEC. The correspondence $\mathcal{O} = \mathcal{U}$ leads to the indistinguishability principle, which itself leads to the presence of entanglement in quantum code. As this first part is coming to an end, we would like present a few open questions to the reader.

1. $\mathbb{Z}_2$ **vs** $\mathfrak{su}(2)$ **vs other algebras:** The difference between quantum and classical codes emerges from the distinct types of data they have to preserve. Classical codes preserve $(\mathbb{Z}_2)^{\otimes n}$ (the cyclic group) while quantum codes preserve $su(2)^{\otimes n}$. Further, entanglement seems to emerge from $\mathfrak{su}(2)$ being equal to its dual. To what extent could these observations be generalised to other groups?

2. **Lie Algebra Dual:** Is there always an easy way to find the dual of a Lie algebra under a certain Lie bracket? This would of course depend heavily on the structure of the measurements. For example, when a measurement has three outcomes, what does it mean to 'flip' the outcome?

3. **Protecting observables vs identifying errors:** It seems that the purpose of codes is specifically to give enough 'space' to errors so that they become identifiable and distinguishable from logicals. Meanwhile the observables themselves are not

protected, the value of a logical $\overline{L}$ can easily be flipped with a single-qubit error. This clashes somewhat with the notion of *fault-tolerance* whose main purpose is to obtain observables whose value can be reliably measured. Can we formalise fault-tolerance in terms of protected observables? Would that entail non-linear codes?

2 | Lower bound on the depth of encoding circuits

The observation that codes host high quantities of entanglement suggests itself as an obstacle to their use. Above certain temperatures entanglement tends to naturally break down in nature [16]; and systems that spontaneously exhibit $\frac{kd}{n}$ qubits of entanglement – for k, d very high – are presumed to be hard to come by. Thus the required entanglement has to be manufactured rather than harvested, and generating large amounts of it is a time consuming task. One then suspects that, given a code $\mathcal{C}$ with high k, d, preparing its codestates cannot be done quickly. Formally, this means that implementing a state-preparation channel Φ of the following form is a lengthy process:

$$|0\rangle\langle 0| \to \Phi(|0\rangle\langle 0|) \in \mathcal{D}(\mathcal{C})$$

Where $\mathcal{D}(\mathcal{C})$ denotes the density matrices supported on $\mathcal{C}$. We ought to expand on what a "quick" process might be. Quantum operations are often described as sequences of unitary operation:

$$\Phi = \mathcal{U}_T \circ \cdots \circ \mathcal{U}_1, \quad \mathcal{U}_t(\cdot) = U_t(\cdot)U_t^\dagger$$

The number T of these operations takes a natural interpretation as a measure of how lengthy the process is. In the case where the class of unitaries U_t is restricted then T becomes particularly significant: depending on what operations are (dis)allowed, T might be *required* to be large in order for $\mathcal{U}_T \circ \cdots \circ \mathcal{U}_1$ to yield Φ. In what follows, for clarity, we will write $U_\Phi : \Phi(\cdot) = U_\Phi(\cdot)U_\Phi^\dagger$.

Most computer architectures are not expected to implement *all* operations. A perennially encountered situation is the case where U_t is said to be 'local'. In our case, a circuit is said to be local when the qubits its act one are embedded in a 2D lattice; and $U_t = \bigotimes_\alpha U_{t,\alpha}$, where each $U_{t,\alpha}$ only acts on nearest neighbour qubits.

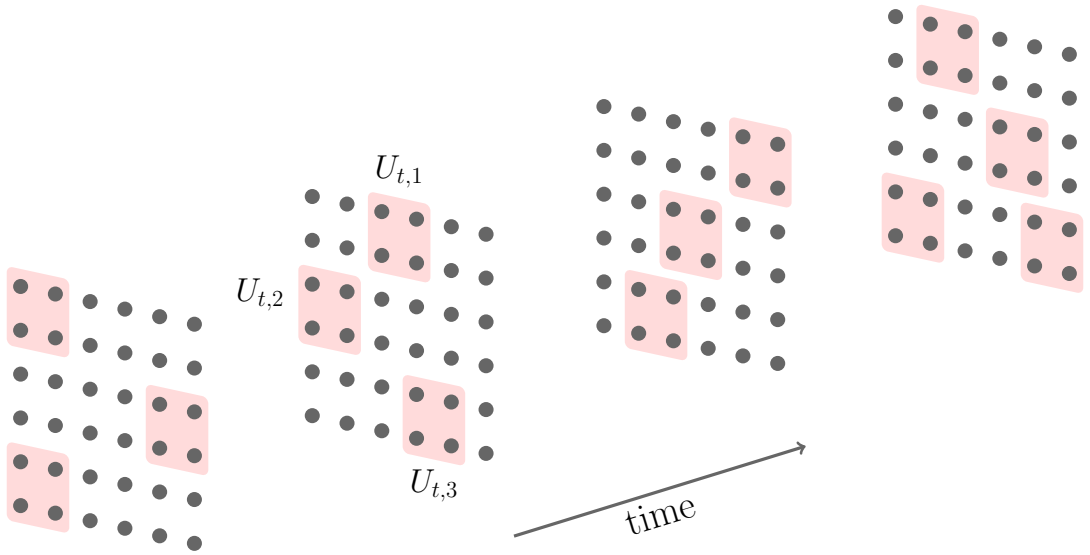


Fig. 2.1 Example of a local computation. Each grid represents the qubits at a given time step, and the red squares correspond to a local unitary operation.

2.1 Unitary circuit bounds: an inspiration

Attempts to show that quantum codes are hard to produce for local circuits are not particularly new. There is even a folklore argument[17] to prove a lower bound on T. A unitary encoding circuit U_Φ is generally taken to send states of the form $|i\rangle \otimes |0\rangle \in (\mathbb{C}^2)^k \otimes (\mathbb{C}^2)^{n-k}$ to the encoded state $|\bar{i}\rangle \in (\mathbb{C}^2)^n$. It follows that U must satisfy

$$U_\Phi Z_i U_\Phi^\dagger = \bar{Z}_i$$

Reading off this relation, the action of U_Φ on Z_i is to spread it across the system: $\bar{Z}_i$ is a much larger operator than Z_i whose support obeys $|\mathrm{supp}(\bar{Z}_i)| \geq d$. This is to be contrasted with the fact that when U_Φ is local then Z_i only propagates slowly. To illustrate this "slow propagation", consider an operator O_A, and a unitary $U_{ABC} = U_{AB} \otimes U_C$; then

$$U_{ABC}(O_A \otimes \mathbb{1}_{BC})U_{ABC}^\dagger = U_{AB}(O_A \otimes \mathbb{1}_B)U_{AB}^\dagger \otimes \mathbb{1}_C$$

Indeed, after acting with U_{ABC} the observable transforms as $O_A \to O'_{AB} = U_{AB}(O_A \otimes \mathbb{1}_B)U_{AB}^\dagger$. And while the support of O'_{AB} increases relative to O_A, it remains contained in the support of U_{AB}: the growth is limited by the structure of U_{ABC}. In the specific 2D context, if O_A is contained in a disc of radius r, then $U_{ABC}(O_A \otimes \mathbb{1}_{BC})U_{ABC}^\dagger$ is contained in a disc of radius $r+1$ – due to the locality of U_{ABC}. With such a growth rate, after

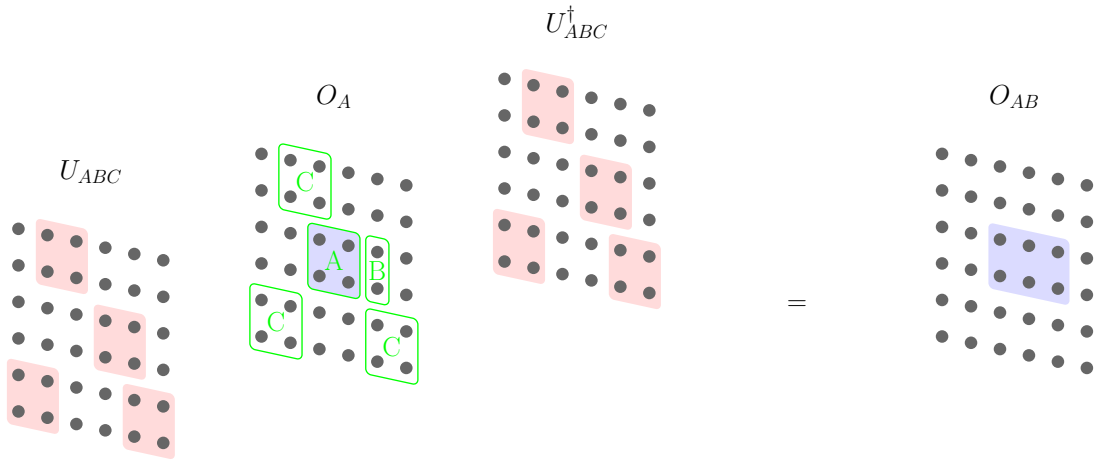


Fig. 2.2 Highlight of a local operation spreading an observable. The gates on the C regions cancel each other out, leading to $U_{ABC}(O_A \otimes \mathbb{1}_{BC})U_{ABC}^\dagger = U_{AB}(O_A \otimes \mathbb{1}_B)U_{AB}^\dagger \otimes \mathbb{1}_C$

T steps, the support of $U_\Phi Z_i U_\Phi^\dagger$ is contained in a ball of radius $T+1$. The volume of that ball gives us the desired upper bound: $|\mathrm{supp}(U_\Phi Z_i U_\Phi^\dagger)| \lesssim T^2$.

In summary, when applying U_Φ we can combine these two seemingly contradictory properties and obtain a lower bound on T [1].

1. $\mathrm{supp}(U_\Phi Z_i U_\Phi^\dagger)$ is big, i.e. $|\mathrm{supp}(UZ_iU)| \geq d$
2. $\mathrm{supp}(U_\Phi Z_i U_\Phi^\dagger)$ grows slowly, $|\mathrm{supp}(UZ_iU)| \lesssim T^2$
3. Consequently, we have $T \gtrsim \sqrt{d}$

This lower bound is striking as, in practice, quantum codes all have large distance, suggesting that initialising these codes is a non-negligible hassle. Yet we are far from the end of our story. We will broaden the set of allowed operations in a way that allows us to break the aforementioned bound, while still describing a realistic model for a quantum computer. The operations in question are the following:

1. Geometrically local unitary operations
2. Geometrically local measurements
3. Long-range classical computation[2]

[1] This argument can also be adapted to the classical setting [18]

[2] By 'long-range classical computation' we mean that we can send the outcome of measurements anywhere in the circuit instantaneously to a classical computer, and this computer can perform arbitrary computations in constant time. Although that model could seem laughably overpowered, it is meant to reflect how much faster classical computers are compared to their quantum counterparts. Further, despite the boundless computational power we are afforded, we show that it is still not enough to make codestate preparation trivial.

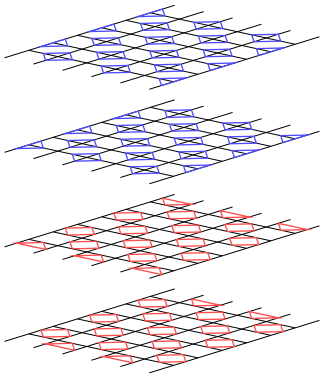


Fig. 2.3 Sketch of a depth four circuit measuring the syndrome of the surface code. On each lattice we highlight the stabilisers that are measured at the corresponding round. After four rounds all the stabilisers are measured.

Consider a 2D circuit measuring the syndrome of the surface code [19, 20]. Due to the locality of the surface code's stabilisers, such a circuit can be built with depth $T \leq O(1)$ from the newly introduced operations. From the syndrome data thus obtained, and the availability of instantaneous classical computation, one can in constant time determine a Pauli correction bringing back the state to the codespace. We end up building a surface code state in constant depth, while the previous lower bound would instead suggest $T \geq \Omega(\sqrt{d}) \gg O(1)$.

Nevertheless, the inclusion of measurements and classical computation does not *readily* make this task trivial for all codes. For example, our argument does not apply in the presence of long-range stabilisers. The lower bound remains broken still, and begs the question: given access to local measurements and non-local classical computation, can we still obtain a meaningful lower bound on the time required to prepare the state of a quantum code? What are the codes whose preparation are made trivial by this computational model, and those that are not?

2.2 The proof outline

In the previous section we attempted to sketch how unitary operations can have a starkly different behaviour from that of a larger, yet still realistic, computational model. This new model – which includes local operations and measurements, along with non-local classical computation – will be referred to as GLOCCs (an acronym of geometrically local operations & classical computation).

Beyond the question of practical applicability, the choice to focus on this set of operations is not innocent. A slightly different set of operations, known as LOCC, has

been intensely studied in Quantum Shannon theory, and it has been long established that they cannot create entanglement between space-separated systems[21]. For the reader unfamiliar with this definition, LOCC operations consist of two spatially separated systems, Alice and Bob, where each party can perform arbitrary (potentially non-geometrically local) unitary operations and measurements on their own system, *plus* they can communicate classical information to each other.

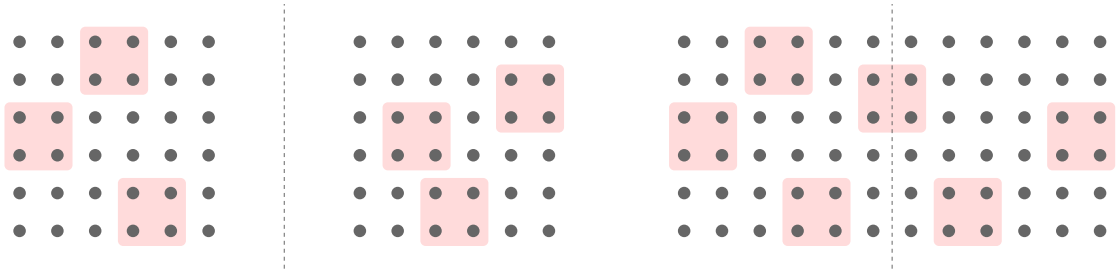


Fig. 2.4 Difference between LOCCs (left) and GLOCCs (right). On the left the system is composed of two completely spatially separated subsets, with the allowed operations acting separately on each; on the right these two subsets are joined and operations can straddle both systems.

The difference between GLOCCs and LOCCs is, primarily, the presence of boundary operations. For a given subset of qubits in a system acted on by a GLOCC, removing the boundary operations reduces to an LOCC. We mean here to appeal to elementary intuition: since LOCCs cannot create entanglement between two separated subsystems systems, then surely GLOCCs can only create entanglement through boundary operations? Should this not imply that GLOCCs are slow at creating entanglement?

Remember that we are trying to prove a bound using steps that mirror closely these of the unitary lower bound. Given the intuition we have sketched in the few paragraphs above, we expect the core of the proof to consist of roughly the following steps:

1. Given a code $\mathcal{C}$ and a codestate $|\psi\rangle \in \mathcal{C}$ on n qubits, there exists a region $A \subset [n]$ sharing a lot of (approximately $\frac{kd}{n}$) qubits of entanglement with the rest of the system, as guaranteed by Theorem 5

2. Creating that entanglement is slow. At each time step the amount of entanglement increases by at most $\partial A \sim \sqrt{d}$)

3. Combining these two elements, we should obtain $T \gtrsim \frac{\frac{kd}{n}}{\sqrt{d}} = \frac{k\sqrt{d}}{n}$.

Some remarks: From a philosophical perspective we could reflect on the following chicken-and-egg problem: Should we study GLOCCs because they accurately represent a relevant experimental context, or are we retrofitting the experimental context to LOCCs because they behave nicely with our ulterior object of interest (entanglement)? After spending a few years pondering this question, we observed that choosing the most convenient tool has often spared us months of tedious, fruitless efforts.

2.3 99 entanglement metrics

After outlining the steps our proof should follow, we are now left with the task of translating them into formal mathematical statements. Our primary worry should be to find an entanglement metric that fulfils both conditions – it is lower bounded by $\frac{kd}{n}$, and grows slowly under GLOCCs.

The first metric we should look at is naturally the coherent information, since Theorem 5 establishes the first condition already. Can we then establish its slow growth? Unfortunately, the coherent information cannot satisfy such a property. Observe that, for an arbitrary region $A \subset [n]$:

$$\rho_{\text{noisy}} = \mathbb{1}/2^{|A|} \otimes |0\rangle\langle 0|_B, \quad I(A\rangle B)_{\rho_{\text{noisy}}} = -|A|$$
$$\rho_{\text{clean}} = |0\rangle\langle 0|_A \otimes \text{tr}_A(\rho_{\text{noisy}}) = |0\rangle\langle 0|_A \otimes |0\rangle\langle 0|_B, \quad I(A\rangle B)_{\rho_{\text{clean}}} = 0$$

It is not hard to verify that $|0\rangle\langle 0|_A \otimes \text{tr}_A(\cdot)$ is a local operation acting strictly on A, and thus, in our model, can drastically increase the value of the coherent information by $\propto |A| \gg |\partial A|$ units in a single step. As a metric, the coherent information is very poorly adapted to the proof steps we are aiming to follow. One can even venture to argue that this incompatibility stems from $I(A\rangle B)$ being overly sensitive to local actions on the A subsystem. While a high coherent information does meaningfully lower bound the entanglement, the converse is not true. It is easy to artificially decrease it, and then just as easy to increase it back up.

Since we used ρ_{noisy} in our argument, we should expand on what entanglement looks like for *mixed states*. A state ρ is said to be entanglement between subsystems A

and B when it is not separable. A 'separable' state is one that has the following form

$$\rho_{AB} = \sum_i p_i \rho_{A,i} \otimes \rho_{B,i}$$

These states as non-entangled because they can be prepared through A and B exchanging only classical communications. Start with the unentangled state $|0\rangle\langle 0|_A \otimes (\sum_i p_i |i\rangle\langle i|)$. The subsystem B sends the value of i to A and prepare $\rho_{B,i}$. Similarly, once A receives i, it can prepare $\rho_{A,i}$ accordingly.

When it comes to replacing the coherent information with a different metric, we are confronted with a *vast* zoo of entanglement metrics ([22], Table 1). And searching for the 'right' one can be time consuming. Instead we suggest to follow our chicken-and-egg motto: the right metric is the one that is convenient to work with. Remember why the coherent information failed: it lacks robustness with respect to operations that do not affect entanglement. To find a better alternative we should then look for one that is strictly non-increasing under LOCCs [3].

However, we are yet to formally define LOCCs, and this for a good reason. In particular differentiating "classical computation" and the rest of quantum operations is less trivial than one might have thought[21]; for one, LOCCs are not closed under closure: an infinite sequence of LOCCs might converge towards a non-LOCC operation. This dispossesses us from some mathematical niceties, for instance we lack a closed form expression to use. Brandishing, once again, our chicken-and-egg motto, we escape this hitch by shifting our attention towards *separable* operations, a set that includes LOCCs, yet is much easier to describe.

Definition 8. A CPTP map $\mathcal{T} : \mathcal{D}(\mathcal{H}_A) \otimes \mathcal{D}(\mathcal{H}_B) \to \mathcal{D}(\mathcal{H}_A) \otimes \mathcal{D}(\mathcal{H}_B)$ is said to belong to $SEP(A:B)$ if

$$\mathcal{T}(\cdot) = \sum_i K_i^A \otimes K_i^B (\cdot) (K_i^A)^\dagger \otimes (K_i^B)^\dagger$$

We move on now to describing the GLOCC analogue of separable operations, GSEP. We will need to highlight the difference between the classical computer subsystem – denoted X – and the rest of the system, labelled Λ.

Definition 9. A CPTP map $\mathcal{T} : \mathcal{D}(\mathcal{H}_\Lambda) \otimes \mathcal{D}(\mathcal{H}_X) \to \mathcal{D}(\mathcal{H}_\Lambda) \otimes \mathcal{D}(\mathcal{H}_X)$ is said to be GSEP (geometrically separable) if

$$\mathcal{T}(\cdot) = \sum_i \left(\bigotimes_j K_{i,j}^X \right) \otimes K_i^\Lambda (\cdot) \left(\bigotimes_j K_{i,j}^{X\dagger} \right) \otimes K_i^{\Lambda\dagger}$$

[3] I.e., we want a metric that increases as $|\partial A|$ for GLOCCs, but not at all under LOCCs.

where each $K^X_{i,j}$ acts on qubits that are a distance at most 1 apart.

One might wonder how expanding the set of allowed operations could be a good idea. Do we not run the risk of including longe-range, entanglement generating operations? As it turns out, separable operations are peculiar in that they are strictly larger than LOCCs, have a nice closed form, but do *not* allow for the creation of entanglement[4]. Separable operations map separable states to separable states:

$$\sum_i K^A_i \otimes K^B_i \left(\sum_j p_j \rho_{A,j} \otimes \rho_{B,j} \right) (K^A_i)^\dagger \otimes (K^B_i)^\dagger = \sum_{i,j} p_j K^A_i \rho_{A,j} (K^A_i)^\dagger \otimes K^B_i \rho_{B,j} (K^B_i)^\dagger$$

Note that $K^A_i \rho_{A,j} (K^A_i)^\dagger \otimes K^B_i \rho_{B,j} (K^B_i)^\dagger$ is positive semi-definite. When can then write

$$p_j K^A_i \rho_{A,j} (K^A_i)^\dagger \otimes K^B_i \rho_{B,j} (K^B_i)^\dagger = p'_{i,j} \rho'_{A,i,j} \otimes \rho'_{B,i,j}$$

where $\rho'_{A,i,j}, \rho'_{B,i,j}$ are bona fide quantum states. In summary, the resulting state is, indeed, a separable state:

$$\sum_i K^A_i \otimes K^B_i \left(\sum_j p_j \rho_{A,j} \otimes \rho_{B,j} \right) (K^A_i)^\dagger \otimes (K^B_i)^\dagger = \sum_{i,j} p'_{i,j} \rho'_{A,i,j} \otimes \rho'_{B,i,j}$$

2.4 A convenient metric: the relative entropy of entanglement

Hopefully by now we have convinced our reader that a metric that is convenient to our purposes should be non-increasing under separable operations. After sifting through many options, we settle on the Relative Entropy of Entanglement (REE).

Definition 10. Let $\rho \in \mathcal{D}(\mathcal{H}_A \otimes \mathcal{H}_B)$, then the relative entropy of entanglement of ρ between A and B is

$$E(A:B)_\rho = \min_{\sigma = \sum_i p_i \sigma_{A,i} \otimes \sigma_{B,i}} D(\rho \| \sigma)$$

With $D(\rho\|\sigma)$ the quantum relative entropy

$$D(\rho\|\sigma) = \operatorname{tr} \rho (\log \rho - \log \sigma)$$

[4] It seems though that they allow for the boosting of entanglement [23].

The REE always satisfies $0 \leq E(A:B)_\rho \leq 2 \cdot \min|A|,|B|$.

If one thinks of $D(\rho\|\sigma)$ as a 'distance' between ρ and σ – a fragile analogy – then $E(A:B)_\rho$ can naturally be interpreted as the distance from ρ to the closest separable, non-entangled state. The larger $E(A:B)_\rho$, the more ρ is *distinguishable*[5] from all separable states, the more it is entangled. Much can be said about the quantum relative entropy, a dizzying amount in fact. In our context there is no quick way to gain intuition about how this quantity behaves either; to prevent leading our reader astray, we will give as few unnecessary details as possible.

As previously hinted at, the REE makes for a very convenient metric with a lot of structure. We will illustrate this via a remarkably simple proof that the REE is non-increasing under SEP operations. At the root of this demonstration is the so-called "data processing inequality":

$$\text{Let } \mathcal{T} \text{ be a quantum channel, } D(\mathcal{T}(\rho)\|\mathcal{T}(\sigma)) \leq D(\rho\|\sigma)$$

This inequality has a natural interpretation: whatever operation $\mathcal{T}$ is applied to the system, it cannot make ρ and σ more distinguishable. The more we manipulate the system, the less easy it is to tell which state we started with. Let $\mathcal{T} \in SEP$, and two states ρ, σ with σ a separable state such that

$$E(A:B)_\rho = D(\rho\|\sigma)$$

Since $\mathcal{T}(\sigma)$ is a separable state too, we then have

$$\begin{aligned} E(A:B)_{\mathcal{T}(\rho)} &= \min_{\sigma=\sum_i p_i \sigma_{A,i}\otimes\sigma_{B,i}} D(\mathcal{T}(\rho)\|\sigma) \\ &\leq D(\mathcal{T}(\rho)\|\mathcal{T}(\sigma)) \\ &\leq D(\rho\|\sigma) \text{ by the Data Processing Inequality} \\ &= E(A:B)_\rho \end{aligned}$$

As desired, for any separable operation $\mathcal{T}$, and any state ρ, we have $E(A:B)_{\mathcal{T}(\rho)} \leq E(A:B)_\rho$

[5] Intuitively, the QRE capture how likely one is to mistake σ for ρ. Obfuscating a fair amount of details: let any M be a projector such that $\mathrm{tr}(M\rho) = 1$, i.e. M reliable 'detects' ρ, then $\mathrm{tr}(M\sigma) \geq 2^{-D(\rho\|\sigma)}$: when trying to identify ρ, there is an irreducible probability $\gtrsim 2^{-D(\rho\|\sigma)}$ that, when presented with σ, we mistakenly identify it as ρ. Formally, this is known as Stein's lemma [24].

2.5 Tying everything together: the proof

With the REE revealing itself to be a 'convenient' metric for our purposes, we can now turn to elaborating the proof of the lower bound. As previously promised, we hope to be able to following these steps to reach the proof.

1. Given a code $\mathcal{C}$ and a codestate $|\psi\rangle \in \mathcal{C}$ on n qubits, there exists a region $A \subset [n]$ sharing a lot of (approximately $\frac{kd}{n}$) qubits of entanglement with the rest of the system, as guaranteed by Theorems 4,5

2. Creating that entanglement is slow. At each time step the amount of entanglement increases by at most $\partial A \sim \sqrt{d}$)

3. Combining these two elements, we should obtain $T \gtrsim \frac{\frac{kd}{n}}{\sqrt{d}} = \frac{k\sqrt{d}}{n}$.

The first condition can be verified without much effort.

Lemma 6 ([25] Lemma 31)**.**

$$E(A:B)_\rho \geq I(A\rangle B)_\rho$$

This establishes that we do not have to worry to much about the first condition. In a sense the REE is a less restrictive quantification of entanglement: if a system holds coherent information, it also holds REE. The second condition however requires a bit more work; we are yet to formally define ∂A.

Definition 11. Let $\mathcal{T}$ be a quantum channel. Given $A \subset [1,\dots,n]$, we say that a subset $S \subset [1,\dots,n]$ contains the boundary ∂A, if there are no Kraus operators in $\mathcal{T}$ with support on both A and $\overline{A} \setminus S$.

With this definition we now address the slow growth of the REE. As previously detailed, to compute $E(A:B)_{\mathcal{T}(\rho)}$ entails the hard task of optimising over separable states σ, until we find the one minimising the quantum relative entropy. Since our primary interest is to upper bound $E(A:B)_{\mathcal{T}(\rho)}$ rather than computing it exactly, we will sidestep this tricky optimisation by instead picking a *proxy* separable state σ_*. The dual purpose of that proxy state will be to, simultaneously, be 'easy' to describe, and be close enough to $\mathcal{T}(\rho)$ so that $D(\mathcal{T}(\rho)\|\sigma_*)$ does not blow up. We choose

$$\sigma_* = \mathcal{T}(\mathrm{tr}_{\partial A}(\sigma) \otimes \mathbb{1}/2^{|\partial A|})$$

where σ is the separable state that minimises $D(\rho\|\sigma)$.

Now that the foundations have been set, the crux of the proof relies on the observation that the quantum relative entropy is "subsystem-continuous", $D(\rho\|\tau)$ and $D(\rho\|\operatorname{tr}_{\partial A}\tau\otimes\mathbb{1}/2^{|\partial A|})$ differ by at most $O(|\partial A|)$.

$$\begin{aligned}
E(A:B)_{\mathcal{T}(\rho)} &\leq D(\mathcal{T}(\rho)\|\sigma_*) \text{ because } \sigma_* \text{ is separable}\\
&= D(\mathcal{T}(\rho)\|\mathcal{T}(\operatorname{tr}_{\partial A}(\sigma)\otimes\mathbb{1}/2^{|\partial A|}))\\
&\leq D(\rho\|\operatorname{tr}_{\partial A}(\sigma)\otimes\mathbb{1}/2^{|\partial A|}) \text{ by the Data Processing Inequality}\\
&= D(\operatorname{tr}_{\partial A}\rho\|\operatorname{tr}_{\partial A}\sigma)+|\partial A|+I(\partial A\rangle AB)_\rho \text{ Proposition 2 of [26]}\\
&\leq D(\operatorname{tr}_{\partial A}\rho\|\operatorname{tr}_{\partial A}\sigma)+2|\partial A| \text{ Theorem 11.5.1 of [15]}\\
&\leq D(\rho\|\sigma)+2|\partial A| \text{ by the Data Processing Inequality}\\
&= E(A:B)_\rho+2|\partial A|
\end{aligned}$$

Note that using a proxy state of the form $\mathcal{T}(\sigma_{\text{some state}})$ is crucial to allow us to 'peel off' $\mathcal{T}$ via the DPI. Once that step is done, it becomes a matter of relating $D(\,\cdot\,\|\operatorname{tr}_{\partial A}(\sigma)\otimes\mathbb{1}/2^{|\partial A|})$ and $D(\,\cdot\,\|\sigma)$. Having followed the previous steps, we formally state the upper bound on the growth of the REE.

Lemma 7. Let $\mathcal{T}\in GSEP$, then $E(A:B)_{\mathcal{T}(\rho)}\leq E(A:B)_\rho+2|\partial A|$

We have thus proved that for every step the REE increases by at most $O(|\partial A|)$ units. To produce $\frac{kd}{n}$ units of entanglement, as requires by Theorem 5, this requires $\Omega(\frac{kd}{n|\partial A|})$ time steps.

Theorem 6. Consider $\mathcal{C}$ a code with parameters $[[n,k,d]]$, and some $|\psi\rangle\in\mathcal{C}$. Let $\mathcal{U}\in CPTP$ such that

$$\mathcal{U}=\mathcal{U}_T\circ\cdots\circ\mathcal{U}_1,\quad \mathcal{U}_t\in GSEP,\quad \text{and}\quad \mathcal{U}(|0\rangle\langle 0|_\Lambda\otimes|0\rangle\langle 0|_X)=|\psi\rangle\langle\psi|$$

Then, T obeys

$$T\gtrsim\frac{k\sqrt{d}}{n}$$

Proof. From Theorem 5, and Lemma 6, there exists $A\subset[n]$ of size $A\propto d$, such that $E(A:\overline{A})_{|\psi\rangle\langle\psi|}\gtrsim\frac{kd}{n}$. Meanwhile, due to the structure of GSEP operations, we have $|\partial A|\lesssim\sqrt{d}$. Lemma 7 then guarantees that $E(A:\overline{A})_{|\psi\rangle\langle\psi|}\lesssim 2T\sqrt{d}$. The theorem then follows. □

2.6 This result in context

Circuit bounds on codes have a long history. Most of the early results were concerned with local unitary gates [27] and showed that the circuit depth had to be extensive. On the opposite end, it was shown that if the locality requirement were dropped, one could access much more efficient circuits [28]. These examples highlight how assumptions can be quite salient in QEC: different platforms offer varying frameworks, which can lead to a landscape of theoretical results. A decade after these result, the community seems to take interest in an intermediary model: local quantum gates but long-range classical communications, which, we assume, is motivated by the existing state of quantum computing engineering. Following this interest, Delfosse et al [29] showed that, for a specific class of circuits incorporating some type of long-range communications, one could obtain non-trivial bounds. That result was striking and suggested more could be said about intermediary locality. It also had some limitations; it made strong assumptions on the type of communication that was allowed, and on the type of code it was concerned with. Their proof was both ingenious and hard to generalize. Some time after that result, we presented the argument presented in this chapter [25]; which improves on both the simplicity of the proof, and its generality.

3 | Combinatorial structures in codes

In previous chapters we have extensively discussed the role entanglement plays in manipulating codes. Better codes host larger amount of entanglement, and creating this entanglement takes time in low dimensions. It is tempting to extent that intuition further: systems encountered in nature frequently contain limited entanglement, does that limit their ability to host good codes?

We ought to sharpen our description of "a physical system hosting a code". In quantum computing a system is usually described by a finite Hamiltonian $H = \sum_i e_i \left|\psi_i\right\rangle\!\left\langle\psi_i\right|$. The subspace spanned by the states of lowest energy, with respect to H, is referred to as the *groundspace* of H. This subspace is of great physical interest, given nature's propensity to stay in the groundspace. Naturally, whenever a groundspace corresponds to a code, the Hamiltonian can be said to *host* that code.

This code/groundspace correspondence is frequently referred to in the context of stabiliser codes. From a set $\mathcal{Q} = \{Q_i\}_i$ of stabilisers describing a code $\mathcal{C}$, we can define a Hamiltonian:

$$H = -\sum Q_i$$

such that its groundspace is precisely $\mathcal{C}$. Such a Hamiltonian evidently inherits the locality of the stabilisers. After discussing these details we can ask our question more precisely:

Given a code $\mathcal{C}$ hosted by a 2D Hamiltonian, are there bounds on the achievable k, d?

3.1 Area law, question mark?

At first, the question that interests us seems very approachable, as many 2D Hamiltonians obey an 'area law'. This terms commonly describe a property of the groundspace where the entanglement of a subset of qubits is upper bounded by the size of its boundary. In the context of the present discussion we introduce the following definition of 'boundary'

Definition 12. Let $H = \sum_i g_i$ be a 2D Hamiltonian on n qubits, and a subset $A \subset [n]$. We write ∂A the set of qubits in $\overline{A}$ connected to A through a stabiliser:

$$\partial A = \{q : q \in \overline{A}, \exists g_i, \text{supp}(g_i) \cap A \neq \emptyset, q \in \text{supp}()g_i\}$$

Then H is said to exhibit an *area law* if:

$$I(A\rangle\overline{A})_{|\psi\rangle} \lesssim |\partial A| \lesssim \sqrt{|A|}$$

for any state $\left|\psi\right\rangle$ in the groundspace of H.

This observation can be combined with Lemma 5 which stipulates a *lower bound* on the entanglement that A holds. If A is taken to have size $\propto d$, then one might expect to obtain a bound as follows:

$$\frac{kd}{n} \lesssim I(A\rangle\overline{A}) \lesssim \sqrt{d} \implies k\sqrt{d} \leq n$$

To be viable, this approach needs resolving several concerns:

1. Proving an area law is extremely hard. The question of 3D (and above) frustration-free Hamiltonians in particular is wide open.

2. Area laws are specifically formulated for Hamiltonians with unique ground states, while code Hamiltonians are designed to have high degeneracy.

Fortunately, *stabiliser* Hamiltonians are much easier to analyse and they will consist our primary object of study. Moreover, although this might come as a surprise, it is not hard to prove an area law *for correctable areas* for these Hamiltonians [1], even when they have high degeneracy. This property can be proved through multiple avenues, the most obvious one being the Disentangling Lemma[14] – although for the sake of investigating a less obvious connection, we will employ a more linear algebraic method.

Equation 1.3 gives us that for a given groundstate $\left|\psi\right\rangle$ of a stabilizer Hamiltonian, and a correctable region M, then we have

$$S(M)_{|\psi\rangle} = |M| - |\mathcal{Q}_M|$$

[1] This might not be as surprising as it sounds. Due to the indistinguishability of quantum code, the reduced density matrix on a correctable area is very much unique, for a given code: we do recover some type of 'uniqueness'.

We will also make use of the projection of $\mathcal{Q}$ onto M: let $Q_M \otimes Q_{\overline{M}} \in \mathcal{Q}$, then we define $\mathrm{proj}_M : \mathcal{Q} \to \mathcal{P}_M$ in the following way:

$$\mathrm{proj}_M(Q_M \otimes Q_{\overline{M}}) = Q_M$$

By extension, the group $\mathrm{proj}_M(\mathcal{Q})$ is generated by $\mathrm{proj}_M(Q)$ for every $Q \in \mathcal{Q}$. With this new group, we can give a group theoretic characterisation of correctable regions. Specifically, M is correctable if and only if [30]

$$|\mathcal{P}_M| - |\mathrm{proj}_M(\mathcal{Q})| = |\mathcal{Q}_M|$$

This will allow us to rewrite the entropy of the region M into a more suitable form. Note that $|\mathcal{P}_M| = 2|M|$, then we obtain:

$$S(M)_{|\psi\rangle} = |M| - 2|M| + |\mathrm{proj}_M(\mathcal{Q})| = |\mathrm{proj}_M(\mathcal{Q})| - |M|$$

The group $\mathrm{proj}_M(\mathcal{Q})$ can be decomposed into two parts. On one hand, $\mathrm{proj}_M(\mathcal{Q}) \cap \mathcal{Q}$ are the elements that are also stabilisers, which can be seen to be equal to $\mathcal{Q}_M$. The remaining elements, $\mathrm{proj}_M(\mathcal{Q}) \setminus \mathcal{Q}$, were obtained from stabilisers straddling both M and $\overline{M}$. If $P_M \in \mathrm{proj}_M(\mathcal{Q}) \setminus \mathcal{Q}$, then there exists $P_M \otimes P_{\overline{M}} \in \mathcal{Q}$ where $P_{\overline{M}} \neq \mathbb{1}$. This allows us to conclude that $\mathrm{proj}_M(\mathcal{Q}) \setminus \mathcal{Q} \subset \mathcal{P}_{\partial M}$. Putting things together, we get:

$$\begin{aligned} I(M\rangle\overline{M})_{|\psi\rangle} &= S(M)_{|\psi\rangle} \\ &= |\mathrm{proj}_M(\mathcal{Q})| - |M| \\ &\leq |\mathcal{Q}_M| + |\mathcal{P}_{\partial M}| - |M| \\ &\leq |M| + |\mathcal{P}_{\partial M}| - |M| \\ &= 2|\partial M| \end{aligned}$$

With this final piece in hand, we obtain the desired bound on the code's parameters.

Theorem 7. Let $\mathcal{C}$ hosted by a 2D stabiliser Hamiltonian, then

$$k\sqrt{d} \lesssim n$$

Proof. Due to the 2D nature of the Hamiltonian, its qubits can be divided into correctable sets $\{A_i\}_i$ of size $\propto d$, and with boundary $|\partial A_i| \lesssim \sqrt{|A_i|} \propto \sqrt{d}$. From the area law, we have $I(A_i\rangle\overline{A_i})_{|\psi\rangle} \lesssim \sqrt{d}$, for any groundstate $|\psi\rangle$.

On the other hand, Lemma 5 guarantees the existence of at least one A_{i_*} such that $I(A_{i_*}\rangle\overline{A_{i_*}})_{|\psi\rangle} \gtrsim \frac{kd}{n}$, for any groundstate $|\psi\rangle$. Combining both bounds on the coherent information gives the expected result. □

3.2 Beyond area laws

The bound of Theorem 7 should feel satisfying. It matches the scaling of Theorem 6, and this is no surprise. Given a local stabiliser Hamiltonian, one can measure its stabilisers with a local circuit of constant depth, then a classical computer can compute a Pauli operator that will correct the syndrome. Overall, this operation can be achieved in constant time given access to GSEP operations. One might then expect a symmetry between GSEP circuits and local Hamiltonians. In this section we show that this is far from true, in fact local Hamiltonians are much worse at reliably storing information – which was originally proved by Bravyi, Poulin, and Terhal (BPT)[14] in the form of a bound on k, d stricter than that of Theorem 6 for T a constant.

Let us first sketch why there exists a discrepancy between the circuit bound, and the Hamiltonian bound. The circuit bound relies on a specific structure when it comes to k and the entanglement:

$$k \leq \sum_i S(A_i) = \sum_i I(A_i\rangle\overline{A_i}) \lesssim \sum_i |\partial A_i| \lesssim \frac{n}{d}\sqrt{d} \tag{3.1}$$

The BPT bound, on the other hand, relies on a different structure. Let $A, B \subset [n]$ be disjoint correctable subsets, and write $C = \overline{AB}$, then

$$k \leq S(C) \tag{3.2}$$

At first sight these two relations seem unrelated, and the second one actually seems much weaker – would not C be huge most of the time? In fact, a significant part of the tour de force that is the BPT result is to provide a method of building large correctable areas, in a way that $|AB| \gg |C|$. The first leg of their result stands on the *Union Lemma* – which states that two correctable regions sufficiently far apart are *jointly* correctable.

Lemma 8 (Union Lemma, [14, 31])**.** Let $\mathcal{C}$ be a stabiliser code on n qubits. For two correctable subsets $U_1, U_2 \subset [n]$, such that $\partial U_1 \cap U_2 = U_1 \cap \partial U_2 = \varnothing$, then $U_1 \cup U_2$ is correctable.

Given this lemma, we can recover the first relation while keeping $B = \varnothing$. Start with a partition $\Lambda = \bigsqcup_i A_i$ into correctable regions with $A_i \propto d$, where $\Lambda = [n]$ denotes the

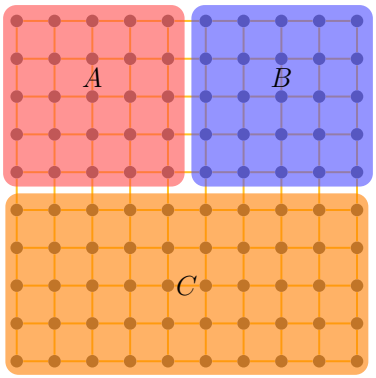


Fig. 3.1 The partitioning lemma BPT [14] uses to upper bound k. Here A and B are two distinct correctable regions, and $k \leq |C|$.

set of qubits. We will write $C = \bigcup_i \partial A_i$. Note that $A \equiv \Lambda \setminus C$ is correctable because of the Union Lemma. Then, we get

$$S(C) \leq \sum_i S(\partial A_i) \lesssim \sum_i |\partial A_i| \lesssim \frac{n}{d}\sqrt{d} \tag{3.3}$$

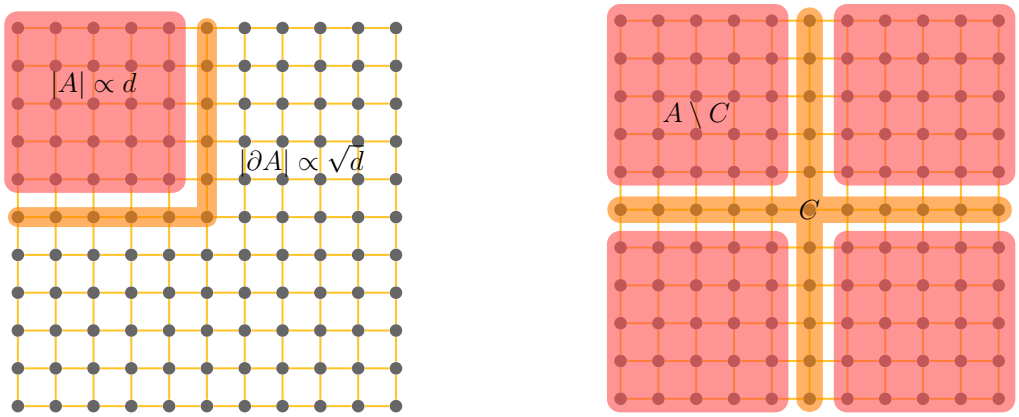


Fig. 3.2 Comparing the partition derived from the Union Lemma (right), see Equation 3.3, and the one from the area law (left), see Equation 3.1. Both end up achieving a similar scaling in 2D.

Which is quantitatively the same as Equation 3.1 *in our context* [2]. Having recovered the first relation from the second, one naturally wonders if we can go beyond it? What happens when $B \neq \varnothing$? The partition can be further augmented, as depicted in Figure 3.3.

[2] For example the Union Lemma is inapplicable in the context of GSEP circuits where the non-overlap condition cannot be guaranteed. Classical communications allows precisely for creating long range entanglement in short time.

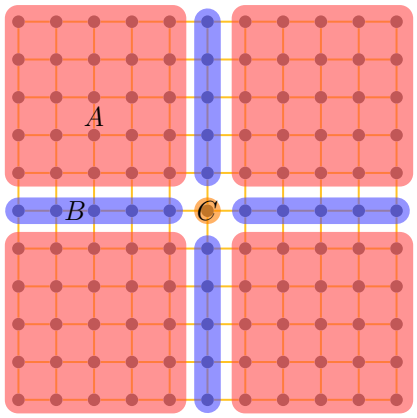


Fig. 3.3 The updated partition. In this case, each of the red squares has size $\propto d$, and each of the blue lines have size $\propto \sqrt{d}$. Applying the Union Lemma one sees that both A and B are correctable, and therefore $k \leq |C|$ a constant.

With this new partition, one can calculate that $k \leq |C| \lesssim \frac{n}{d}$, a significant improvement over Equation 3.1, and consequent the resulting bound is stricter than that of the circuit bound. Local circuits appear 'stronger' than local Hamiltonians. We should note that the proper BPT bound is yet stricter and reads $kd^2 \lesssim n$. It is obtained through the use of the Expansion Lemma, which we elected not to discuss for the sake of conciseness. We summarize the different bounds in Figure 3.4.

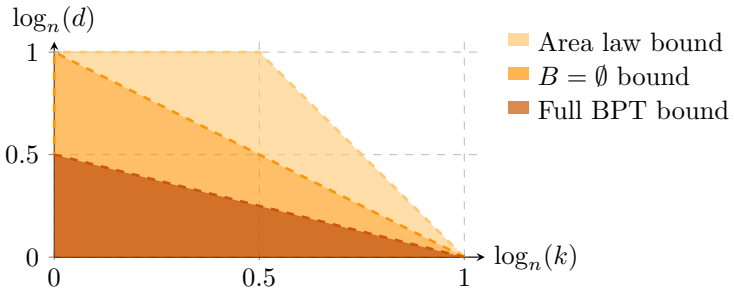


Fig. 3.4 Summary of the different bounds in this section. The 'Area law bound' corresponds to Equation 3.1, the '$B = \varnothing$ bound' corresponds to Equation 3.3; and the 'Full BPT bound' refers to the bound as originally stated in [14].

3.3 Expansion

The BPT argument can be generalised to Hamiltonians local in D Euclidean dimension, with a resulting bound of $kd^{2/(D-1)} \lesssim n$. In 2023 it was shown that the BPT bound can be saturated , and this discovery confirms how D is a good quantifier of the locality of

a code. The lower D is, the less efficient the codes are; conversely, the larger D is, the better are the codes we can build.

There is nevertheless a significant drawback to the dimension. Not every architecture can be neatly modelled as a D dimensional lattice. Long range interactions abound in optical systems, and superconducting platforms can even afford some amount of non-locality. In short, the BPT bound falls short in situations of intermediate connectivity: what can we say about models that are neither exactly 3D nor exactly 2D? From a theoretical perspective, it is also worth asking how non-Euclidean spaces compare: maybe a computer living in hyperbolic space would be a lot better at correcting errors? The general issue we want to answer is as follows.

Given an architecture, can we precisely capture how well this architecture can correct errors based on its connectivity alone?

At first sight, the task is overwhelming. "Connectivity" is a vague, context dependent notion that is hard to reduce to a number. How do we formalise "connectivity" [32]? Drawing inspiration from graph theory, our approach will be three fold:

1. To every stabiliser code we assign a connectivity structure, a *connectivity graph*
2. The extent to which a connectivity graph is well connected will be quantified by *expansion*
3. We will show that lower expansion implies worse codes

For physicists, graph theory is often uncharted territory. Fortunately graph theory is also very amenable to figures, which we will rely on to build intuition. Let us illustrate this by defining the *connectivity graph* of a code.

Definition 13 (Connectivity graph)**.** Let $\mathcal{C}$ be a code on n qubits with stabilisers $\mathcal{S} = \{S_i\}_i$; then the connectivity graph $G = (V, E)$ associated with these stabilisers is defined as:

1. $V = [n]$, i.e. each vertex is associated with a qubit, and
2. $(u, v) \in E$ if and only if there exists a generator $S \in \mathcal{S}$ such that $u, v \in \mathrm{supp}(S)$.

Remarks: We would like to note that the connectivity graph is a function of $\mathcal{S}$, and not of $\mathcal{C}$; different sets of generators might give the same code but different connectivity graphs. Similarly, different codes might yield the same connectivity graph, if the right

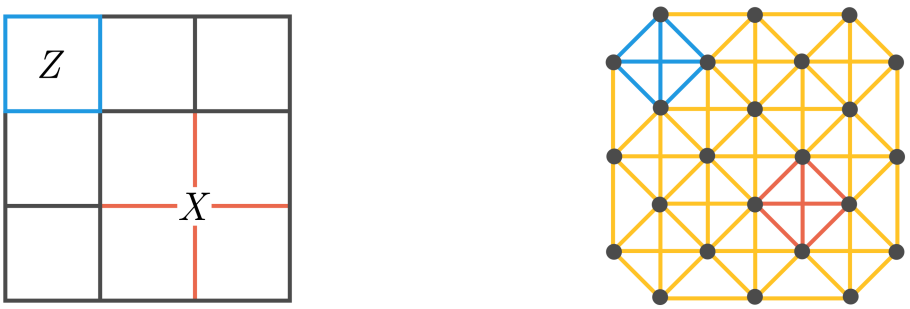


Fig. 3.5 Comparison between a patch of surface code (left), with the resulting connectivity graph (right). The Z and X stabilisers highlighted on the left result in the edges highlighted in in the right picture. From [31].

generating sets are chosen. This representation 'forgets' some information about the code to focus on its connectivity.

We move on now to *expansion*. In the previous sections, the reader might have noticed how much of an emphasis we put on *boundaries*. From the Lemma 7, to area laws, to our sketch of the BPT bound, the relationship between $|A|$ and $|\partial A|$ is omnipresent, and almost overwhelming [3]. The expansion of a graph is merely a way to quantify this relationship for *any* graph, and not solely D-dimensional lattices [4].

Definition 14. Let $G = (V, E)$ a finite graph. For any $U \subset V$, we define

$$\phi_G(U) = \frac{|\partial U|}{|U|}$$

where $\partial U \subset E$ is the set of edges with exactly one endpoint in U. The small scale expansion constant $h_\alpha(G)$ is defined as

$$\min_{U \subset V, |U| \leq \alpha |V|} \phi_G(U)$$

And we say that G is ϵ-expander if $h_{1/2}(G) \geq \epsilon$.

As an example a 2D lattice on n vertices obeys $h_{1/2}(G) \propto \frac{1}{\sqrt{n}}$. In general, a D-dimensional lattice follows $h_{1/2}(G) \propto n^{(D-1)/D}/n = 1/n^{1/D}$: the larger the dimension, the larger the boundary/volume ratio gets, and the larger the expansion is. This trend

[3] The key link between these topics is the necessity to partition a graph into regions of small boundary.

[4] Expansion is only one of many graph theoretical quantification of 'boundary'. One can also point to separators[31, 32, 33], and other metrics could be amenable to bounds yet not studied, like graph growth. Interestingly, different metrics tend to give inequivalent bounds, with no metric to 'rule them all'.

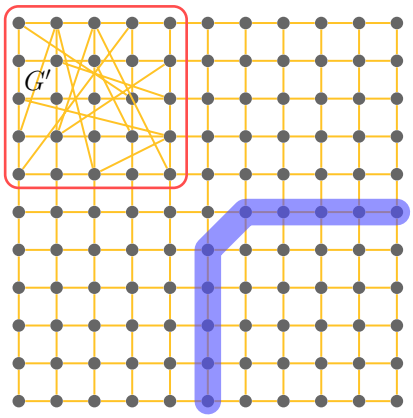


Fig. 3.6 Example of a graph G that is not very expanding due to the cut in blue, yet still contains a very dense subgraph.

should guide our intuition; we expect that graphs with small expansion correspond with poor parameters.

Some remarks on 'expansion' vs 'dimension': It is always possible to embed low Euclidean D structures into higher D structure. However it is generally not possible to do the reverse. In this sense the dimension is a consistent quantifier of locality. Similarly, it is always possible to embed low expansion graph into high expansion graphs at low cost, and not possible to do the reverse [34].

3.4 Expansion concentration

We have hinted several times at our present goal: we wish to show that a graph that does not contain expansion can be easily partitioned into subgraphs with small boundaries. Such a decomposition would allow us to transpose the BPT argument, and obtain bounds on codes from expansion.

Although this might make intuitive sense, it is not readily clear how this should be formally translated into mathematical terms. The very first obstacle we have to face is that even a graph with $h_\alpha(G)$ very small might still contain a subgraph $G' \subset G$ that very densely connected, and thus hard to partition.

How do we verify whether h_α is *uniformly* small, even for subgraphs of G? How do we capture the small scale behaviour of G? Intuitively, if we remove these smaller, densely connected subgraphs, then G should be 'uniformly' low-expanding at all

scales. Extending our reasoning further, if G does not have *any* such small[5] expanding subgraphs, then it is uniformly low-expanding.

Therefore in this section we introduce new tools and results that will allow us to prove just that. In fact, we will ascertain the contrapositive: if $h_\alpha(G)$ is large, then there must exist a very expanding subgraph G'. For concreteness, we choose this to mean that $h_{1/2}(G')$ is large.

Definition 15. Let $G = (V, E)$ be a graph. A separator is a subset $S \subset E$ of edges such that its removal leaves two disconnected subgraphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ such that $|V_1|, |V_2| \leq \frac{2}{3}n$. We can now define $\mathsf{sep}(G)$:

$$\mathsf{sep}(G) = \min_{S \text{ is a separator of } G'} |S|$$

At times it will be useful to consider a subset of ∂U. For some $W \subset V$, we write $\partial^W U$ the set of edges that have one endpoint in U, and the other in W. Finally $G[U] = (U, E_U)$ is the subgraph induced by U where $E_U \subset E$ denotes the edges in E that have both endpoints in U.

Lemma 9 (Inspired by [35], Lemma 12)**.** Let $G = (V, E)$ be a graph on n vertices. If there is no $U \subset V$ with $|U| \geq 2n/3$ such that $G[U]$ is ϵ-expander then $\mathsf{sep}(G) \leq \frac{2}{3}\epsilon n$.

Proof. We recursively define three sequences $(A_i)_i, (S_i)_i, (B_i)_i$. Let $A_0 = S_0 = \emptyset$, $B_0 = V$. As long as $|B_i| \geq \frac{2}{3}n$ then $G[B_i]$ is not an ϵ-expander, and we can find a set $A_{i+1} \subset B_i, |A_{i+1}| \leq |B_i|/2$ such that $\phi_{G[B_i]}(A_{i+1}) \leq \epsilon$. We let $S_{i+1} \equiv \partial^{B_i} A_{i+1}$, and $B_{i+1} = B_i \setminus A_i$. Write l the first index for which $|B_l| \leq \frac{2}{3}n$. We then have, by definition $|B_l| = |B_{l-1}| - |A_l| \geq |B_{l-1}| - \frac{1}{2}|B_{l-1}| = \frac{1}{2}|B_{l-1}| \geq \frac{1}{3}n$.

We obtain $A' \equiv \bigcup_{i=1}^{i=l} A_i$, $B' \equiv B_l$, $S' \equiv \bigcup_{i=1}^{i=l} S_i$, where $A' \sqcup B'$ is a partition of V, and removing the edges S' disconnects the vertices of A' from the vertices of B'. Since $\frac{1}{3}n \leq B' \leq \frac{2}{3}n$, then $\frac{1}{3}n \leq A' \leq \frac{2}{3}n$, we conclude that S' is a separator for G. Further we can verify that

$$|S'| = \sum_i |\partial^{B_i} A_{i+1}| = \sum \phi_{G[B_i]}(A_{i+1})|A_{i+1}| \leq \epsilon \sum_i |A_{i+1}| = \epsilon |A'| \leq \frac{2}{3}\epsilon n \,.$$

This gives $\mathsf{sep}(G) \leq \frac{2}{3}\epsilon n$. □

Corollary 1. Let $G = (V, E)$ be a graph on n vertices. If $\mathsf{sep}(G) \geq \epsilon n$, then there exists $U \subset V$ with $|U| \geq 2n/3$ such that $G[U]$ is $\frac{3}{2}\epsilon$-expander.

[5] We hope these subgraphs are not too small either, as to avoid having 'zoom in' the graph too much. We hope that an intermediary scale is able to tell us enough about the graph at all scales.

Proof. Contrapositive of the previous lemma. □

With these tools in hand we can obtain our first proposition.

Proposition 1 (Expansion concentration)**.** Let $G = (V, E)$ be a graph on n vertices satisfying

$$h_\alpha(G) \geq \epsilon$$

then there exists $U \subset V, |U| \geq \frac{2}{3}\alpha|V|$, such that

$$h_{1/2}(G[U]) \geq \frac{\epsilon}{4\log_{3/2}(1/\alpha)} \in \Omega\left(\frac{\epsilon}{\log(n)}\right)$$

Proof. If $h_{1/2}(G) \geq \epsilon/2$ the lemma follows, so we will assume $h_{1/2}(G) \leq \epsilon/2$. This assumption guarantees the existence of $U \subset V, \alpha|V| \leq |U| \leq |V|/2$ such that $\phi_G(U) \leq \epsilon/2$. We will recursively define a sequence $(Y_i)_i, Y_i \subset V$, and we take $Y_1 \equiv U$.

By definition, for any partition $Y_i = A \sqcup B$, we have

$$\phi_G(Y_i) = \frac{|\partial Y_i|}{|Y_i|} = \frac{|\partial^{\overline{Y_i}} A| + |\partial^{\overline{Y_i}} B|}{|A| + |B|}$$

where $\overline{Y_i} \equiv V \setminus Y_i$.

WLOG, we take $\frac{|\partial^{\overline{Y_i}} A|}{|A|} \leq \frac{|\partial^{\overline{Y_i}} B|}{|B|}$, this gives:

$$\frac{|\partial^{\overline{Y_i}} A|}{|A|} \leq \phi_G(Y_i) \leq \frac{|\partial^{\overline{Y_i}} B|}{|B|}$$

Using the fact that $\partial A = \partial^{\overline{Y_i}} A \cup \partial^B A$, we can obtain:

$$\frac{|\partial A| - |\partial^B A|}{|A|} = \frac{|\partial^{\overline{Y_i}} A|}{|A|} \leq \phi_G(Y_i)$$

It remains to specify our choice of A, B. Let $G[Y_i]$ have an edge separator S_i, then $Y_i = A \sqcup B$ with $\frac{1}{3}|Y_i| \leq |A| \leq \frac{2}{3}|Y_i|$. We choose $Y_{i+1} = A$, applying the previous inequality, this yields

$$\phi_G(A) \leq \phi_G(Y_i) + \frac{|S_i|}{|Y_{i+1}|}\,.$$

We write $\delta_i \equiv \frac{|S_i|}{|Y_{i+1}|}$. Substituting Y_{i+1} for Y_i, we repeat the argument recursively until the first l such that $|Y_{l+1}| \leq \alpha|V|$. At that point, we have

$$\epsilon \leq \phi_G(Y_{l+1}) \leq \phi_G(Y_l) + \delta_l = \phi_G(Y_1) + \sum_{i=1}^{i=l} \delta_i$$

or

$$\sum_i \delta_i \geq \epsilon/2$$

Due to the upper bound on the size of $|Y_{i+1}| \leq \frac{2}{3}|Y_i|$, l is at most $\lceil \log_{3/2}(n/\alpha n) \rceil$. Write $i' = \arg\max_i \delta_i$, then

$$\lceil \log_{3/2}(\alpha^{-1}) \rceil \delta_{i'} \geq \epsilon/2$$

or

$$\delta_{i'} \geq \frac{\epsilon}{2\lceil \log_{3/2}(\alpha^{-1}) \rceil}$$

By the definition of a separator, we know that $|Y_{i'+1}| \geq \frac{1}{3}|Y_{i'}|$, which yields

$$\frac{|S_{i'}|}{\frac{1}{3}|Y_{i'}|} \geq \frac{|S_{i'}|}{|Y_{i'+1}|} = \delta_{i'} \geq \frac{\epsilon}{2\lceil \log_{3/2}(\alpha^{-1}) \rceil}$$

or

$$|S_{i'}| \geq \frac{\epsilon}{6\lceil \log_{3/2}(\alpha^{-1}) \rceil} |Y_{i'}|$$

By applying Corollary 1, we can guarantee that there exists $U \subset Y_{i'}$ such that $|U| \geq \frac{2}{3}|Y_{i'}| \geq \frac{2}{3}\alpha|V|$, and $G[U]$ is $\frac{\epsilon}{4\lceil \log_{3/2}(\alpha^{-1}) \rceil}$-expander. □

3.5 Partitioning non-expanding graphs at low cost

We have shown that graphs without no 'dense' subgraphs of size $\alpha|V|$ necessarily have a small $h_\alpha(G)$. In this section, we will use this stepping stone to show our main theoretical result: a graph with no "dense" subgraphs can be partitioned at low cost. For the sake of convenience, we will use a slightly different definition for the small scale expansion constant.

Definition 16. Let $G = (V, E)$ be a finite graph. The small scale expansion constant $\hat{h}_m(G)$ is defined as

$$\hat{h}_m(G) = \min_{U \subset V, |U| \leq m} \phi_G(U)$$

Proposition 1 can then be rephrased using the modified definition.

Proposition 2 (Expansion concentration)**.** Let $G = (V, E)$ be a graph on n vertices satisfying

$$\hat{h}_m(G) \geq \epsilon$$

then there exists $U \subset V, |U| \geq \frac{2}{3}m$, and a universal constant c such that $G[U]$ is $\frac{\epsilon}{c \cdot \log(n)}$-expander.

As a result, we obtain the following corollary – a simple converse of Proposition 2.

Corollary 2. Let $G = (V, E)$ be a graph on n vertices for which there is no $U \subset V$, $|U| \geq m$ such that $G[U]$ is ϵ-expander, then

$$\forall G' \subset G,\ \hat{h}_{\frac{3}{2}m}(G') \leq c \cdot \epsilon \cdot \log(n)$$

Lemma 10. Let $G = (V, E)$ such that any $G' \subset G$ with $|G'| \geq m$ satisfies $\hat{h}_m(G') \leq \epsilon$. Then there exists a partition of V into regions $\{A'_j\}_{j=1}^{j=t}$, with $|A'_j| < 2m$, $t \leq n/m + 1$, and at most ϵn edges from E have endpoints in different A'_j's.

Proof. We recursively define three sequences $(A_i)_i, (S_i)_i, (B_i)_i$. Let $A_0 = S_0 = \emptyset$, $B_0 = V$. As long as $|B_i| \geq m$ then $\hat{h}_m(G[B_i]) \leq \epsilon$, and we can find a set $A_{i+1} \subset B_i, |A_{i+1}| \leq m$ such that $\phi_{G[B_i]}(A_{i+1}) \leq \epsilon$. We let $S_{i+1} \equiv \partial^{B_i} A_{i+1}$. Write l the first index for which $|B_l| \leq m$, and we pick $A_{l+1} \equiv B_l$.

Now note that $V = \bigsqcup_{i=1}^{i=l+1} A_i$, and $|A_i| \leq m$. We can reorganize those A_i's to form bigger (but no too big) sets A'_j. Indeed define

$$A'_1 = \bigcup_{i=1}^{i=l'_1} A_i \ ,$$

where l'_1 is the smallest integer such that

$$|\bigcup_{i=1}^{i=l'_1+1} A_i| \geq 2m \ ,$$

then one can verify that $m \leq A'_1 < 2m$. We can repeat this process a number of times t until we exhaust all A_i's. We thus obtain $\{A'_j\}_{j=1}^{j=t}$, where $V = \bigsqcup_{j=1}^{j=t} A'_j$. For $1 \leq j \leq t-1$

we have $m \leq |A'_j| < 2m$, and $|A'_t| \leq m$: we cannot lower bound the size of the last set. We now obtain an upper bound on t by noting that

$$n \geq (t-1) \cdot \min_{j:1\leq j\leq t-1} |A'_j| \geq (t-1) \cdot m \,,$$

this gives

$$t \leq \frac{n}{m} + 1$$

By definition all the edges between A'_{j_1}, A'_{j_2} are contained within $S' = \bigcup_{i=1}^{l} S_i$. We can bound its size

$$|S'| = \sum_{i=0}^{i=l-1} |\partial^{B_i} A_{i+1}| = \sum_{i=0}^{i=l-1} \phi_{G[B_i]}(A_{i+1}) |A_{i+1}| \leq \epsilon \sum_{i=0}^{i=l-1} |A_{i+1}| = \epsilon n \,.$$

□

Combining Corollary 2 and Lemma 10, we obtain a general partitioning theorem:

Theorem 8. Let $G = (V, E)$ be a graph on n vertices for which there is no $U \subset V$, $|U| \geq m$ such that $G[U]$ is ϵ-expander, then there exists disjoint subsets $\{W_i\}_i$ such that

1. $|W_i| < 3m$
2. Writing $W = \bigcup_i W_i$, we have $|V \setminus W| \leq 2c \cdot \log(n) \cdot \epsilon n$

Proof. Applying Corollary 2 to Proposition 10, we obtain a partition of V:

1. $V = \bigcup_i A_i$
2. $|\bigcup_i \partial A_i| \leq c \cdot \log(n) \cdot \epsilon n$

Let S be the vertices corresponding to the endpoints of the edges in $\bigcup_i \partial A_i$; then $W_i \equiv A_i \setminus S$ gives the desired subsets. Note that $|S| = |V \setminus W| \leq 2c \cdot \log(n) \cdot \epsilon n$. □

This partitioning theorem can be applied recursively to give us a corollary that will be more readily applicable to quantum codes.

Corollary 3. Let $G = (V, E)$ be a graph on n vertices for which there is no $U \subset V$, $|U| \geq m$ such that $G[U]$ is ϵ-expander, then there exists subsets $\{W_i\}_i$ and $\{W'_j\}_j$ such that

1. The sets $\{W_i\}_i$ are mutually disjoint

2. The sets $\{W'_j\}_j$ are mutually disjoint
3. $|W_i|, |W'_j| < 3m$
4. Writing $W = \bigcup_i W_i$, $W' = \bigcup_j W'_j$, we have $|V \setminus W \setminus W'| \leq (2c \cdot \log(n) \cdot \epsilon)^2 n$

Proof. We apply Theorem 8, and get the sets $\{W_i\}_i$. We re-apply Theorem 8 to $V \setminus W$ which gives us $\{W'_j\}_j$. □

3.6 Expansion-based bounds on codes

Having finally obtained a theoretical guarantee on which graphs can be easily partitioned, we can derive the main result of this chapter. Equation 3.2, used with Corollary 3 gives the following result.

Theorem 9. Let $G = (V, E)$ be a connectivity graph for a code $\mathcal{C}$ on n vertices for which there is no $U \subset V$, $|U| \geq m$ such that $G[U]$ is ϵ-expander, then if $\mathcal{C}$ is a quantum code, it obeys either

1. $d < 3m$, or
2. $k \leq (2c \cdot \log(n) \cdot \epsilon)^2 n$

Proof. We assume $d > 3m$. Corollary 3 gives us two correctable sets of qubits, W, W', and $|V \setminus W \setminus W'| \leq (2c \cdot \log(n) \cdot \epsilon)^2 n$. Invoking the BPT lemma (Equation 3.2) guarantees that W and W' are correctable, and yields the desired result.

□

Theorem 9 can be understood as putting a constraint over the architecture implementing a code $\mathcal{C}$. The better $\mathcal{C}$ is, the larger and denser the resulting expander graph is. In fact, this result can be amplified to show that one needs *many* subgraphs with expansion.

Corollary 4. If $\mathcal{C}$ is a quantum $[[n, k, d]]$ code then, there exists $\{K_i\}_i, K_i \subset G$, with G a connectivity graph for $\mathcal{C}$, such that $\sum_i |K_i| \geq \frac{k}{2}$, $|K_i| \geq \frac{d}{3}$, and each K_i is ϵ-expander, for $\epsilon \gtrsim \sqrt{\frac{k}{n \log(n)^2}}$.

Proof. Let $\{K_i\}_i$ be a (potentially empty) set of distinct subgraphs such that $G[V \setminus \{K_i\}_i]$ does not contain any ϵ-expander subgraph of size at least m. We write $\sum_i |K_i| = M$, and WLOG, all K_i's can be assumed to satisfy:

1. $|K_i| \geq m$
2. K_i is ϵ-expander

Then, by definition, we can apply Theorem 9 to $G[V \setminus \{K_i\}_i]$, to get:

1. $d < 3m$, or
2. $k \leq |V \setminus W \setminus W'| \leq M + (2c \cdot \log(n) \cdot \epsilon)^2 n$

Setting $M \leq \frac{1}{2}k, m = d/3$ we obtain the desired result by contradiction.

□

Optimality of our results. While the surface code perfectly saturates the BPT bound, it does not saturate the bound of Corollary 4. Plugging in $k = 1, d \propto \sqrt{n}$ we are guaranteed the existence of a subgraph K of size $d/3 \propto \sqrt{n}$, and of expansion $\propto 1/\sqrt{n}\log\left(n^2\right)$. Meanwhile the surface code lattice has size n, and expansion $1/\sqrt{n}$. This highlights that our bound is very much not tight for 2D graphs; it seems that what we gain in generality, we lose in tightness.

3.7 Epilogue

What are these results relevant to? Constructing expanding graphs *explicitly* is a rather tricky task theoretically, and even more so experimentally. Since expander graphs are the most 'general' type of graphs[6], the connections one has to implement are fairly complex, and often inaccessible to specific platforms. These results reveal that, in the quest toward efficient quantum computers, the connectivity of a platform has a significant impact on its eventual performance. From an experimental perspective, there exist algorithms to efficiently estimate the expansion profile of a graph [36]; which could allow one to grasp the error correcting capabilities of their platform.

It is notable that our bounds from Corollary 4 do not exactly recover the BPT bounds when specialised to 2D codes. Although our result is more general, it is also less expressive in that specific case – which seems to be a common issue with graph theoretical bounds[31, 32, 33]. As we close this chapter, we reflect on some potential explanations.

1. **Expansion Lemma:** The BPT bound relies on the use of the so-called expansion lemma, which allows for obtaining correctable regions A_i that are much larger

[6] General graphs are 'easy' to embed into expanders, while the reverse is false. [34]

than d. The use of this lemma requires control over the size of the boundary of A_i, which our partitioning method does not allow for. If a version of Theorem 8 holds with the guarantee that $|\partial A_i| \leq \epsilon m$, there is a good chance that Theorem 9 can be improved upon through the Expansion Lemma – in a way similar to [33].

2. **Swiss cheese:** The partitioning used in Theorem 9 proceeds in two steps. We first carve out A, then B. This two-time procedure leads to some loss of information: after carving out A, the remaining graph is left with big holes – much akin to a piece of Swiss cheese – and a presumably weakened connectivity. Our proof does not take this into account and assumes that G' is just as well connected as G – which is fanciful. 'Experimentally', one can verify that a 2D lattice is much less expanding after going through the first step, but can we quantify that in the general setting? As a concrete instance, consider Figure 3.3; the graph induced by removing the A region is clearly much less connected than the original graph – it even appears quasi-1D.

3. **Non-Euclidean spaces:** Although we would be surprised if our bounds cannot be improved upon, it is possible that some spaces exist with an expansion similar to that of Euclidean space while hosting better codes; hence not 'allowing' the bound to be tightened too much. In particular, isotropic spaces make for an appealing object of study due to their uniform structure, and their simple classification. For example, it is known that the hyperbolic space is less good at hosting codes than the Euclidean space [31] for a fixed dimension. Could it be that the Euclidean space is the 'best' of the isotropic spaces at hosting codes?

4. **Other combinatorial metrics.** What other combinatorial metrics can be used to obtain bounds on codes? Is there a general way to saturate them? Some partial results have addressed the case of expansion for classical and subsystem codes [37, 38]. Is there one metric objectively better than the others?

4 | Layer codes

We have talked at lengths about bounds on codes; yet it remains unclear whether there exist codes saturating these bounds, and what these codes might look like. Conversations surrounding these bounds can feel lacking in substance: what do they really characterise if they are not tight?

"Extremal QEC", the study of quantum codes given certain constraints, has found little success; despite years of research very few results were known prior to 2023. Among these rare successes are the Haah code [39] – which demonstrated non-string operators – and the welded code [40] – which demonstrated a power law energy barrier. Meanwhile creating codes is comparatively easier in a non-local setting. Tools like high-dimensional expanders have led to a flurry of quantum LDPC codes, some with optimal parameters $[[n, \Theta(n), \Theta(n)]]$ [41, 42]; while comparable methods do not seem to be readily available for low dimensional spaces. Finding classical codes saturating the classical BPT bound, $k\sqrt{d} \lesssim n$, has turned out equally laboursome. Despite the sustained interest in the community for such codes, only one partial result is known and at the cost of great effort [43]. In this chapter we will show how to saturate the 2D classical BPT bound, and the 3D quantum BPT bound. Our solution can come across as simple, and maybe even disappointingly so, as it does not involve very sophisticated mathematical tools. We elect to view it as a reminder that choosing the right perspective when tackling a problem is often half of the solution. When it comes to the problem that interests us, as bottom-up approaches have remained fruitless, we were left to consider more naive approaches. Could we create non-local codes, and then find a way to embed them into lower dimensions?

The obvious concern with this method is that, in the process of embedding these codes, we are very likely to come across stabilisers spanning long distances. Ideally, we wish to be able to replace these long range interactions with short range ones. This process, however, should decrease neither the distance, nor the number of logical qubits.

Fig. 4.1 A well connected graph (on the left) tends to induce very long edges when embedded in 2D [32].

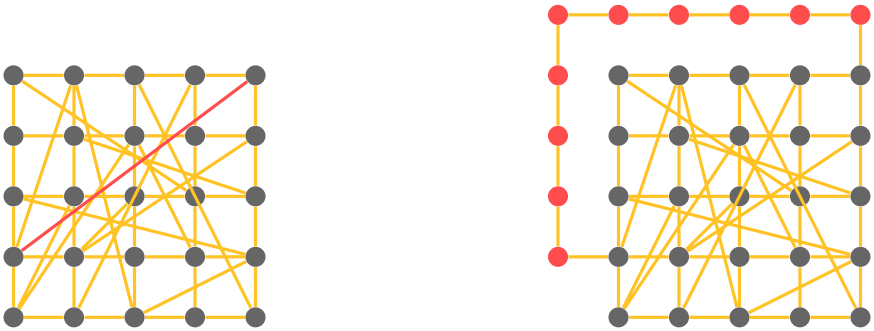


Fig. 4.2 We hope to be able to replace a long edge (highlighted on the left), with a set of ancillary qubits and some local stabilisers (on the right).

'Simulating' some physical interaction with different, simpler interactions is a topic that has accumulated a significant interest in the past 20 years – mostly in complexity theoretic subfields, and with no overlap with QEC. In this chapter we will use a definition that differs significantly from that commonly encountered in the literature. During the simulation process we are primarily interested in decreasing neither the distance, nor the number of logical qubits of the resulting code. A simple way to ensure both conditions hold is to ensure that the new logicals merely consist of the former logicals, 'augmented' with some operator on the ancillary subsystem:

Definition 17. Let $\mathcal{C} \subset \mathcal{H}$ be a stabiliser code with logicals $\{L_i\}_{i=1}^{i=l}$, then $\mathcal{C}' \subset \mathcal{H} \otimes \mathcal{H}_{\text{ancilla}}$ is an acceptable simulation of $\mathcal{C}$ if its logicals can be expressed as $\{L_i \otimes L_i^{\text{ancilla}}\}_{i=1}^{i=l}$, for some L_i^{ancilla}'s.

Related works Our work came a little bit after an approximate result by Portnoy [44]. His construction relies on a two step process: mapping an LDPC code to a 4D manifold, then leveraging the Gromov-Guth embedding to obtain a local code for *any* target dimension. While both of these steps were probabilistic in [44], the work of [45, 46] described an explicit code-to-manifold mapping. According to [46], an

undisclosed result would also allow this line of construction to exactly saturate the BPT bound. Interestingly, all of these results, as do ours, rely on embedding high dimensional objects.

We would also like to highlight [47], which achieves the remarkable feat of expressing the layer codes construction in the more general language of algebraic topology/homological algebra.

4.1 Simulating classical codes

Before tackling the delicate case of simulating quantum codes, we find it insightful to begin with a discussion of classical codes. Historically, finding optimal local classical codes has been no easier a task, with the first constructions being publicised in the same year of 2023. Both problems, classical and quantum, being solved simultaneously points to them presenting similar challenges. We hope that introducing the classical case is conducive to building intuition that will be helpful for the next section. We will frequently use a graphical language to represent codes and their stabilisers, see Figure 4.3. A blue square represents a gauge check, and orange circles are qubits; the solid lines correspond to a Z operators, while the dashed lines correspond to X operators. Correspondingly, two parallel solid and dashed lines correspond to a Y operator.

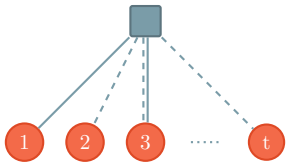


Fig. 4.3 In this example, the check represented is $Z_1X_2Y_3\ldots X_t$.

We begin our study by considering a ZZ check between two distant bits. Although we only depict the two bits, the assumption of the distance being significant is important to the relevance of the argument. We will argue that this ZZ check can be simulated by the code in Figure 4.4.

Our suggestion is, admittedly, ex-nihilo [1]. Although it appears disarmingly simple, it remains non-trivial to construct it from first principles. Anecdotally, we came up with this observation after staring at [48] for too long.

[1] Ironically, the resulting code is very commonly encountered in introductory courses on error correction, under the name 'repetition code'. However, to the best of our knowledge, the connection with simulation is new.

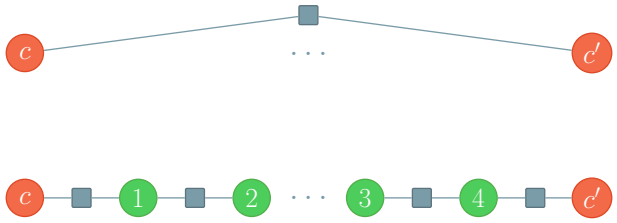


Fig. 4.4 The ZZ stabiliser depicted above can be simulated with the code illustrated below.

Nevertheless, we can still demonstrate *why* this simulation works. A Z_1Z_2 check commutes with the operator $Q_1 \otimes Q_2$ if and only if $[Q_1 \otimes Q_2, Z_1 \otimes Z_1] = [Q_1, Z_1] \otimes [Q_2, Z_1] = \mathbb{1}$, or equivalently, $[Q_1, Z_1] = [Q_2, Z_2]$ [2]. Since we are considering classical codes, the set of errors is restricted to $Q_1, Q_2 \in \{\mathbb{1}, X\}$; this ensures that $Z_1 \otimes Z_2$ commutes with $Q_1 \otimes Q_2$ if and only if $Q_1 = Q_2$. In effect, when all checks are satisfied, we are effectively copying the value of a bit across space, all the way to its neighbour (see Figure 4.5); if $[Q_i \otimes Q_{i+1}, Z_i \otimes Z_{i+1}] = [Q_{i+1} \otimes Q_{i+2}, Z_{i+1} \otimes Z_{i+2}] = \mathbb{1}$ then $[Q_i \otimes Q_{i+2}, Z_i \otimes Z_{i+2}] = \mathbb{1}$. We can conclude that the checks on the ancillary subsystem commute only if $Q_c \otimes Q_{c'}$ commutes with $Z_c \otimes Z_{c'}$.

Fig. 4.5 If all the checks commute, then the value of the ancillary bit has to coincide with c. In this sense, the repetition code allows for the value c to 'travel' through space.

Similarly, it is not hard to see that once Q_c and $Q_{c'}$ are fixed such that $Q_c \otimes Q_{c'}$ commutes with $Z_c \otimes Z_{c'}$, there is a unique L_i^{ancilla} such that $Q_c \otimes Q_{c'} \otimes L_i^{\text{ancilla}}$ commutes with all the checks on the ancillary subsystem. Combining these observations we arrive to the desired result:

Proposition 3. Let $\mathcal{C} \subset \mathcal{H}$ be a stabiliser code with logicals $\{L_i\}_{i=1}^{i=l}$. Simulating an ZZ arbitrary check of $\mathcal{C}$ with the gadget of Figure 4.4 yields $\mathcal{C}' \subset \mathcal{H} \otimes \mathcal{H}_{\text{ancilla}}$; such that its logicals can be expressed as $\{L_i \otimes L_i^{\text{ancilla}}\}_{i=1}^{i=l}$.

We would only be fully satisfied if this approach allowed us to simulate arbitrary ω-body checks – which will demonstrate shortly. In fact, any ω-body check can be simulated by using checks of weight at most 3, see Figure 4.6.

The explanation of why this works is very similar to the ZZ check case. We will focus on an arbitrary logical $L = Q_1 \otimes \cdots \otimes Q_\omega \otimes Q_{a_1} \otimes \cdots \otimes Q_{a_{\omega-1}}$ supported on

[2] Although classically we have $[Q_1, Z] = [Q_2, Z] \implies Q_1 = Q_2$, in the quantum real the only thing we propagate is the anti-commutation value.

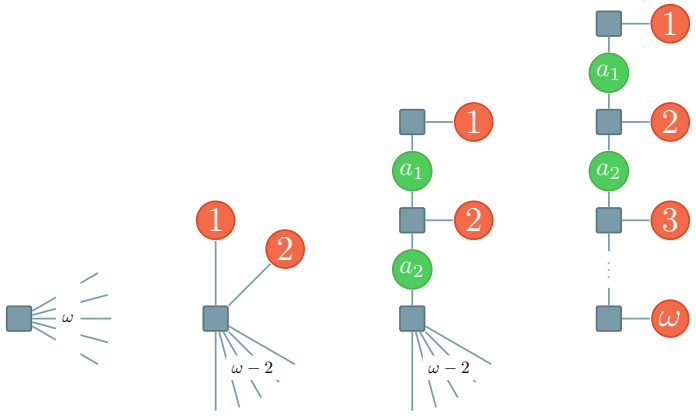


Fig. 4.6 For an arbitrary weight ω, we can recursively obtain a gadget simulating a ω-body check.

$\mathcal{H} \otimes \mathcal{H}_{\text{ancilla}}$. First note that, $Q_{a_1} \otimes Q_1$ commutes with $Z_{a_1} \otimes Z_1$ if and only if $Q_{a_1} = Q_1$; as previously argued for the ZZ check. Secondly, we have that $Z_{a_1} \otimes Z_1 \otimes Z_{a_2}$ and $Q_{a_1} \otimes Q_2 \otimes Q_{a_2}$ commutes if and only if $[Z_{a_2}, Q_{a_2}] = [Z_{a_1}, Q_{a_1}] \cdot [Z_2, Q_2] = [Z_1, Q_1] \cdot [Z_2, Q_2]$. This argument can be repeated until we obtain $[Z_{a_{\omega-1}}, Q_{a_{\omega-1}}] = [Z_1, Q_1] \cdot [Z_2, Q_2] \ldots [Z_{\omega-1}, Q_{\omega-1}]$. Finally, the last check obeys:

$$\begin{aligned}
[Z_{a_{\omega-1}} \otimes Z_\omega, Q_{a_{\omega-1}} \otimes Q_\omega] &= [Z_{a_{\omega-1}}, Q_{a_{\omega-1}}] \otimes [Z_\omega, Q_\omega] \\
&= ([Z_1, Q_1] \cdot [Z_2, Q_2] \ldots [Z_{\omega-1}, Q_{\omega-1}]) \otimes [Z_\omega, Q_\omega] \\
&= [Z_1 \otimes \cdots \otimes Z_\omega, Q_1 \otimes \cdots \otimes Q_\omega]
\end{aligned}$$

We conclude that the new checks all commute only if $Z_1 \otimes \cdots \otimes Z_\omega$ commutes with $Q_1 \otimes \cdots \otimes Q_\omega$. Conversely, for any operator $Q_1 \otimes \cdots \otimes Q_\omega$ commuting with the check $Z_1 \otimes \cdots \otimes Z_\omega$, there exists an operator $Q_{a_1} \otimes \cdots \otimes Q_{a_{\omega-1}}$ such that $Q_1 \otimes \cdots \otimes Q_\omega \otimes Q_{a_1} \otimes \cdots \otimes Q_{a_{\omega-1}}$ commutes with all the new checks. This allows us to formally establish the ω-body equivalent of Proposition 3:

Proposition 4. Let $\mathcal{C} \subset \mathcal{H}$ be a stabiliser code with logicals $\{L_i\}_{i=1}^{i=l}$. Simulating an $Z^{\otimes\omega}$ arbitrary check of $\mathcal{C}$ with the gadget of Figure 4.6 yields $\mathcal{C}' \subset \mathcal{H} \otimes \mathcal{H}_{\text{ancilla}}$; such that its logicals can be expressed as $\{L_i \otimes L_i^{\text{ancilla}}\}_{i=1}^{i=l}$.

4.1.1 Embedding codes

The last section hopefully makes for a satisfactory description of how one can trade long-range connections for the overhead of copying values across space. One can easily imagine how this can be useful in practice: a single puzzlesome interaction can

be replaced by a simpler, local, apparatus. It is tempting though to push beyond a mere punctual application, and to investigate the outer limits of our new tool. Can we embed a code constituting of *mostly* long range interactions into a 2D platform?

The answer is a resounding yes. In what follows we will describe how to map an arbitrary classical LDPC code $\mathcal{C}$ to a 2D local code $\mathcal{C}'$. This mapping will obey Definition 17, and as such we expect the parameters $[n,k,d]$ of $\mathcal{C}$ to become $[n',k,d']$ in $\mathcal{C}'$ [3].

Algorithm 1

Require: A classical code $\mathcal{C}$ with stabilisers $\mathcal{S} = \{S_i\}_{i=1}^{i=m}$, and parameters $[n,k,d]$.
Ensure: A classical code $\mathcal{C}'$ local in 2D, and with parameters $[2nm,k,dm]$.

▷ We use the notation (x,y) to denote the coordinates of the 2D grid the code is embedded in.

for each $b \in [n]$ **do**
  Embed the bit b at the coordinate $(1,b)$.
  Add a bit at every point $(2,b),(3,b),\dots(m,b)$.
  Add the checks $Z_{(1,b)}Z_{(2,b)}, Z_{(2,b)}Z_{(3,b)}, \dots Z_{(m-1,b)}Z_{(m,b)}$ to the set of stabilisers.
    ▷ At this point, we have essentially replaced the bit b with a repetition code spanning the coordinates $(1,b)$ to (m,b).
end for
for each $j \in [m]$ **do**
  Let $\{b_1,\dots,b_\delta\}$ be the bits in the supports of s_j.
  Move s_j to the j-th column of bits: $s_j = Z_{(b_1,m)}Z_{(b_2,m)}\dots Z_{(b_\delta,m)}$
  Localise s_j by using the gadget of Proposition 4.
end for

We accompany the formal description of the embedding procedure with a step-by-step walkthrough using the Hamming code as input. As a reminder, the Hamming code has parameters $[7,4,3]$, and has the following stabilisers:

$$Z_1Z_3Z_5Z_7, \quad Z_2Z_3Z_6Z_7, \quad Z_4Z_5Z_6Z_7$$

The first step of the algorithm requires that we embed each bit in a line along the $\hat{y}$ direction, and to then extent each bit into a repetition code of size $m = 3$:

[3] The simulation definition we use states that k is preserved, hence why it does not change. It also requires that $d' \geq d$.

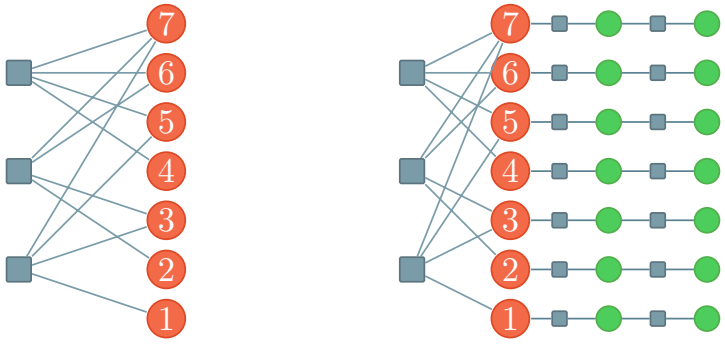


Fig. 4.7 The input code (on the left) is mapped to a new code, where each bit is extended by a repetition code (on the right).

We are then told to move the j-th check to the j-th column of bits. This is done to avoid adding many overlapping ancillary bits, when it will come to localise the checks. Finally, we use the gadget of Proposition 4 to localise each check in its respective column.

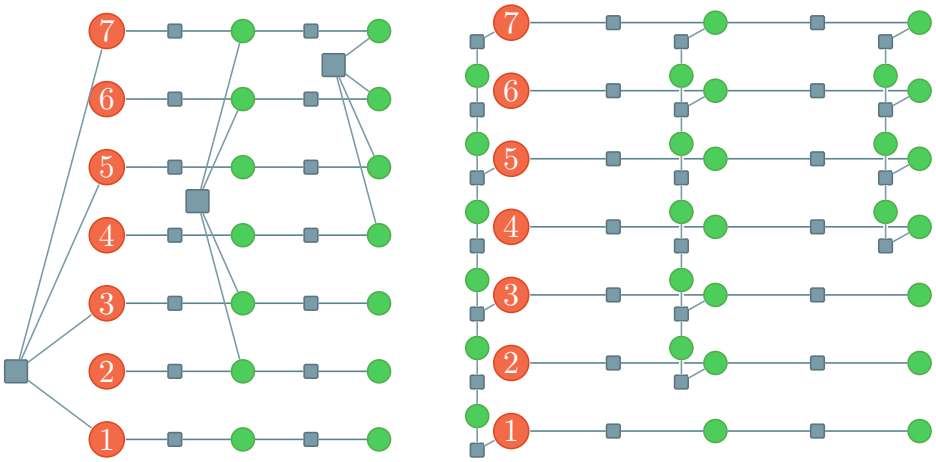


Fig. 4.8 After moving each check into a distinct column (on the left), we can make these checks local (on the right). For every check, its respective simulation only occupies its corresponding column.

Since the resulting code requires no long-range connections and constant bit density, we have successfully localised the Hamming code. The procedure we describe can be summarised by replacing every bit with a repetition code, and using the length of these to implement the required checks locally. After this illustration, the statement of our main result should not be surprising:

Theorem 10. Let $\mathcal{C}$ be a classical code with parameters $[n,k,d]$ and m stabilisers; then Algorithm 1 yields a 2D local code $\mathcal{C}'$ with parameters $[2nm,k,dm]$

Proof. By construction $\mathcal{C}'$ is 2D local; and the number of bit can be upper bounded by counting. Each bit is mapped to a m-long repetition code, yielding nm bits, and each check is mapped to a gadget using at most n ancillary bits, yielding $2nm$.

Regarding k and d, one can verify that $\mathcal{C}$ extended with repetition codes obeys $k' = k$, and $d' = dm$ [4]. The application of the localisation gadget does not decrease these parameters, per Proposition 4, which wraps up the proof. □

4.1.2 Saturating the classical BPT bound

We can obtain codes saturating the BPT bound as a simple corollary of Theorem 10. The input code $\mathcal{C}$ will be a family of LDPC codes with parameters $[n, \Theta(n), \Theta(n)]$, and $m \lesssim n$ – due to their LDPC-ness. Applying Theorem 10 to these families then yields codes with parameters $[n, \Theta(\sqrt{n}), \Theta(n)]$, see Figure 4.9.

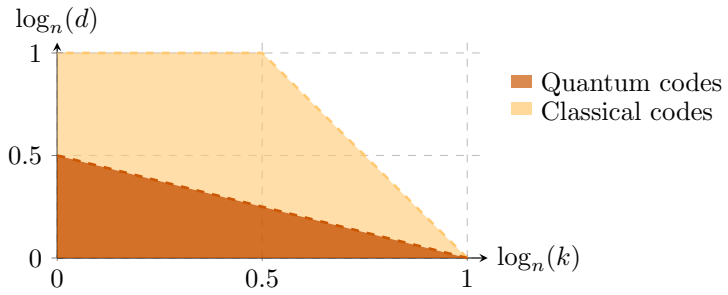


Fig. 4.9 BPT bounds, for classical codes and stabiliser codes. We should note that having access to a code situated at the extremal points of these bounds is enough to fully saturate the bound, see Corollary 1 of [49] .

Our simple argument was the first [37] to prove the exact achievability of this bound (see [44] for an earlier approximate result), and the existence of these codes. Given the accessibility of our argument and it relying on no elaborate mathematical tools, one might be curious as to why such a long time elapsed between the BPT paper [14] and these constructions. We would like to suggest that the answer lies behind the lack of translation symmetry: studying non-symmetrical structures is usually hard and rarely achievable. We leave it as an open question whether our construction can be made translationally invariant.

[4] In fact this corresponds to concatenating $\mathcal{C}$ with a repetition code.

4.2 Simulating stabiliser codes

With the classical BPT bound reached, our main concern becomes transmuting the proof to the quantum realm: how do we simulate an arbitrary stabiliser code with a 2D local stabiliser code? In this effort we are met with several obstacles. The definition of a 'simulation' that we relied on becomes obsolete due to the Cleaning Lemma: some logical operators could be pushed *into* a slate of ancillary qubits. In stabiliser codes, logicals are very fluid and mobile; it does not make sense to expect they assume a strict structure. What, then, should we taken 'simulation' to mean here? Despite our best efforts, we failed to come up with an alternative definition that would be satisfyingly general, *and* could ensure that the resulting code maintains high k and d. We decided to settle on a necessary condition that will guide us through the next section.

Let $\mathcal{C} \subset \mathcal{H}$ be a stabiliser code with stabiliser group $\mathcal{Q}$. Then $\mathcal{C}' \subset \mathcal{H} \otimes \mathcal{H}_{ancilla}$ is a simulation of $\mathcal{C}$ only if for every $Q \in \mathcal{Q}$, there exists $Q \otimes \mathbb{1}_{ancilla} \in \mathcal{Q}'$; where $\mathcal{Q}'$ is the stabiliser group of $\mathcal{C}'$.

This condition has the merit of clarifying how the simulation should work. We replace s with some different (presumably more local) stabilisers, *but* the value of s still remains accessible to us through the new stabilisers.

We should note that our embedding is restricted to CSS codes; equivalently, stabiliser codes whose stabiliser group obeys $\mathcal{Q} \subset \{\mathbb{1}, X\}^{\otimes n} \cup \{\mathbb{1}, Z\}^{\otimes n}$. We will frequently refer to Pauli in $\{\mathbb{1}, X\}^{\otimes n}$ as X-type operators, and similarly for Z. Although this does not affect our ability to saturate the BPT bound, we mention this limitation for the sake of completeness.

4.2.1 Surface codes

Our next challenge is posed to us by repetition codes. In the previous section these were crucial to our construction. By mapping bits to repetition codes, we 'stretched' them out in space, making the information they hold more widespread. One might then surmise that, to achieve an equivalent construction for quantum codes, we first need a 'quantum repetition code'. In particular, this new code needs to protect both the X and the Z information that a qubit holds. This challenge can naturally be overcome by choosing the surface code – which indeed can be interpreted as a 'product'[5] of two repetition codes, one for each of X and Z.

[5] In fact, a hypergraph product [50].

The surface code is defined on a 2D lattice of size $m_X \times m_Z$, where the qubits are located on the edges. To every vertex v is associated a stabiliser $A_v = \prod_{\text{edge}\in v} X_{\text{edge}}$, and to every plaquette p we associate a stabiliser $B_p = \prod_{\text{edge}\in p} Z_{\text{edge}}$. It encodes one logical qubit and has distance $d = \min m_X, m_Z$.

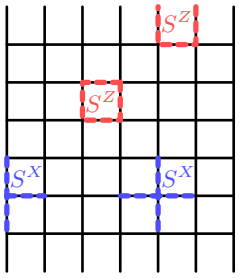


Fig. 4.10 The standard surface code with some stabilisers highlighted.

The syndrome of the surface code can be conveniently categorised. When an A_v anticommutes with an operator, we call that an e excitation, while a B_p stabiliser gives an m excitation. For example, a single Z operator in the middle of a surface code gives two e excitations. Operators of constant-sized support introduce an equivalence relation between sets of excitations: two sets are equivalent if a constant-sized operator can go from one to the other. The single Z operator then establishes the correspondence between a *pair* of e excitations, and no excitations.

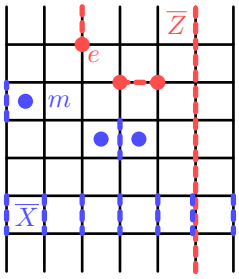


Fig. 4.11 The surface code and its condensation rules: a single-Pauli operator in the bulk of the code will generate a pair of excitations; while one on the boundary will only induce a single one.

When an operator P allows us to go from no excitations to a set of excitations, we say that P *condenses* this set of excitations. The rules dictating which excitations can and cannot be condensed are a defining feature of the surface code. In the bulk of the lattice, one can only condense pairs of e, or pairs of m excitations; however, at the boundaries things change entirely. The top/bottom boundaries allow for the condensation of a single e excitation, while the left/right boundaries generate single

m excitations. A region such as these boundaries where the condensation rules differ from that of the bulk are called *defects*.

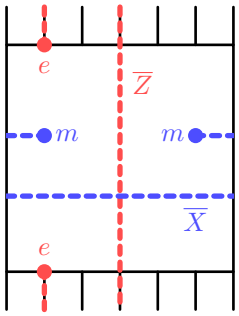


Fig. 4.12 The condensation dictates the structure of the logicals. The e excitation at the top can be 'dragged' all the way down to be paired with the e excitation at the bottom. The result is the $\overline{Z}$ logical operator.

4.2.2 Condensation rules as stabilisers

The condensation rules of the surface codes take a large role in the following section. They provide a singular proxy to the behaviour of the operators living on the code, and specifically, its logicals. A telling illustration is how these rules can be used to show that the surface code's distance is greater than $\min m_X, m_Z$.

Let L be a non-correctable Z logical. We write L_{top} and L_{bottom} the restriction of L to, respectively, the top boundary qubits, and the bottom boundary qubits. Since the bulk is correctable, then both L_{top} and L_{bottom} have to condense an odd number of e excitations at their respective boundary.

Because excitations are created in pairs in the bulk, the only way for L_{bulk} – the restriction of L to the bulk of the lattice – to cancel these excitations, is to span the distance between the top and the bottom layer, or equivalently, $|L| \geq m_Z$. By repeating this argument for X operators, we obtain $d \geq \min m_Z, m_X$. The core intuition we wish to highlight is the following:

Excitations are a proxy for operators. By restricting how they condense, we restrict what operators are logicals. More formally, an operator L belongs to the normaliser only if the truncated logical L_{defect} induces excitation that are condensable by said defect.

There is an alternative, more direct way to show that condensation rules map to stabilisers. A boundary that condenses an excitation m can condense it twice. These two excitations can be joined together, and cancel each other out. We obtain a

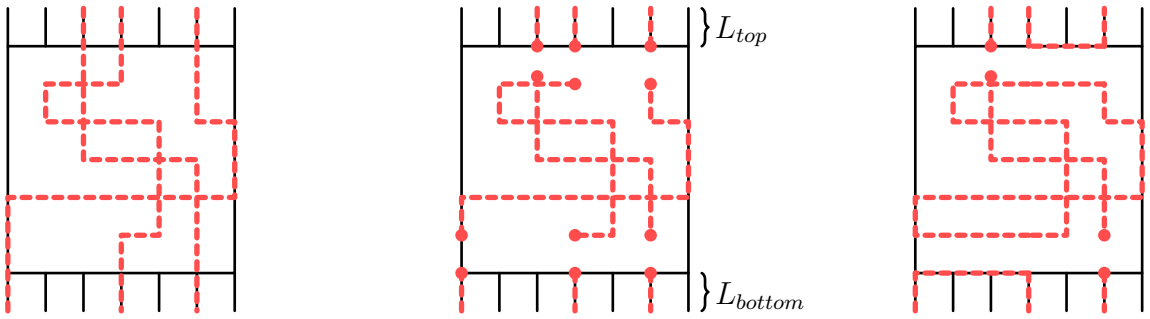


Fig. 4.13 An arbitrary logical (on the left). The induced operators L_{top} and L_{bottom} leave a set of excitations at their respective boundaries (middle). Using stabilisers these boundary excitations can be paired up, leaving either one, or zero excitations, depending on their parity (on the right). If zero excitations are left then these stabilisers disconnect L_{top}/L_{bottom} from the bulk.

syndrome-free operator that by definition belongs to the normaliser. Further, because the surface code has a macroscopic distance this constant size operator *has to* belong to the stabiliser. These stabilisers ensure that a Z logical *cannot* touch the boundary, and has to span the length of the surface code patch – reflecting the above argument regarding the distance. The careful reader might notice that these induced stabilisers are eerily close to what our simulation condition required. Indeed, we will make extensive use of condensation rules to retrieve $Q \otimes \mathbb{1}$ in the new stabilisers.

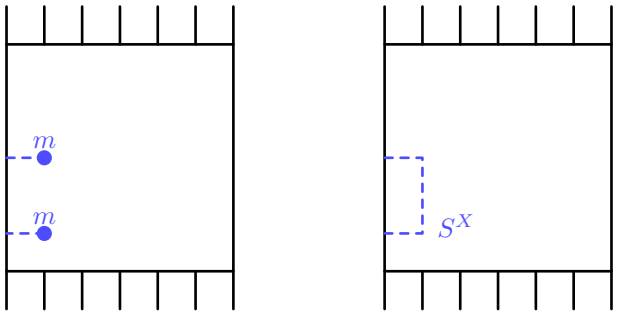


Fig. 4.14 A pair of excitations emerging from the same boundary can be joined to create a stabiliser.

4.2.3 Embedding checks

By comparison with the classical case, it should be clear by now that instead of replacing every bit by a repetition code, we intend to replace every qubit with a

surface code. It remains however to elucidate how to embed the checks of the original code.

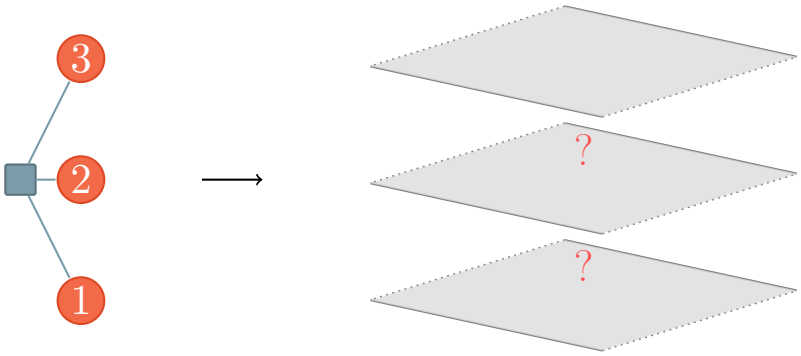


Fig. 4.15 When embedding the code on the left, our first step is to map each qubit to a surface code. We use the convention where the dotted boundaries correspond to the e condensing boundaries. It remains however to elucidate hot to emulate the ZZZ check on these surface codes.

Given that the qubits' data is now encoded in the logicals of each surface code, it is only evident that embedding the original checks now entails acting on the surface codes' logicals. In short, to embed ZZZ we need to enforce the operator $\bar{Z}\bar{Z}\bar{Z}$. This is a significant inconvenience: to the $\mathcal{O} = \mathcal{U}$ correspondence of quantum mechanics, each of these $\bar{Z}$ is an operator of size $d \propto \sqrt{n}$ and are consequently very non-local. We will show that, in fact, $\bar{Z}\bar{Z}\bar{Z}$ can be enforced through local stabilisers by introducing new condensation rules.

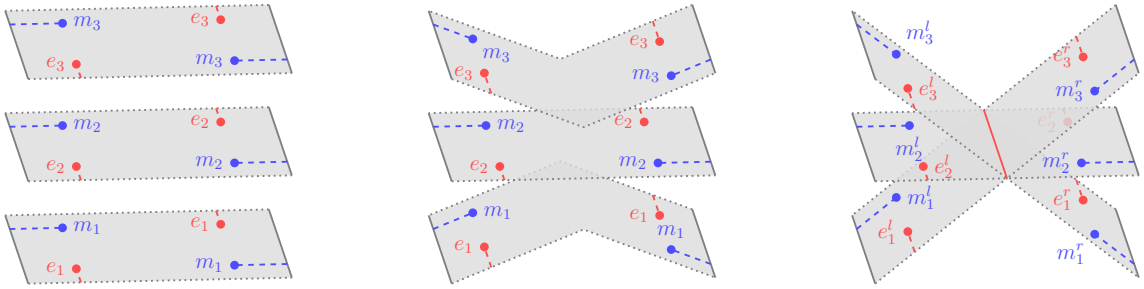
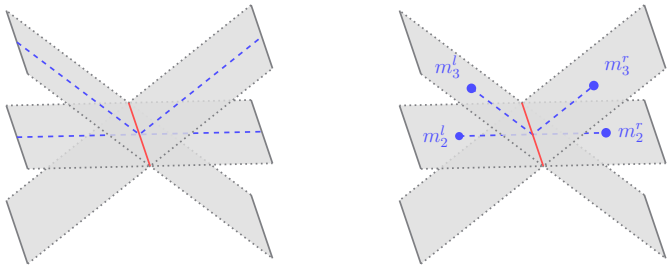


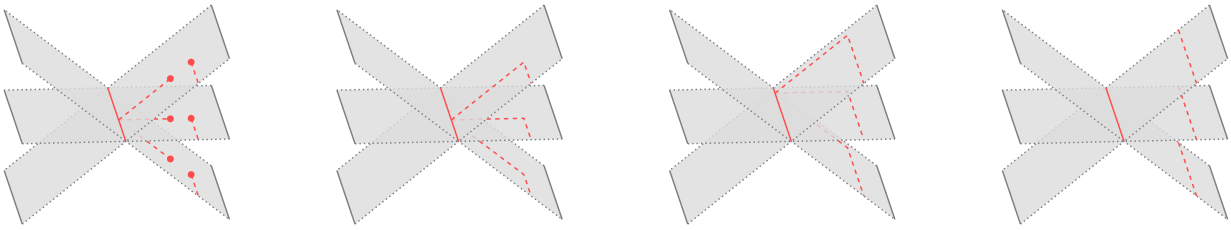
Fig. 4.16 Three surface codes being joined together along a (red) line defect. This line defect introduces a split between the excitations, hence the labelling e_i^l/e_i^r, for 'left' and 'right'.

The only m excitations that we wish to allow are the ones that correspond to operators commuting with $\bar{Z}\bar{Z}\bar{Z}$: the defect will condense $\{m_1^l m_2^l m_1^r m_2^r, m_2^l m_3^l m_2^r m_3^r, m_1^l m_3^l m_1^r m_3^r\}$. Additionally, we should include $\{e_1^l e_2^l e_3^l, e_1^r e_2^r e_3^r\}$ as $\bar{Z}\bar{Z}\bar{Z}$ should introduce new relations between the Z logicals; these are precisely the excitations that will allow us to

recover $\bar{Z}\bar{Z}\bar{Z}$ in the new stabilisers. Lastly, this defect should allow Z operators to cross the boundary unaffected, thus $e_i^r e_i^l$ should also be included. These conditions are illustrated in Figure 4.17.



(a) The defect condensing $m_2^l m_3^l m_2^r m_3^r$ is a logical consequence of $\bar{X}\bar{X}$ being an allowed logical: truncating it gives precisely $m_2^l m_3^l m_2^r m_3^r$ near the defect, and therefore it has to be condenseable.



(b) Once condensed, the excitation $e_1^r e_2^r e_3^r$ can be stretched into the corresponding stabiliser $\bar{Z}\bar{Z}\bar{Z}$. This ensures that only logicals commuting with $\bar{Z}\bar{Z}\bar{Z}$ are allowed in the new code.

Fig. 4.17

The new defect now only allows X operators that commute with $\bar{Z}\bar{Z}\bar{Z}$, and we can recover the value of $\bar{Z}\bar{Z}\bar{Z}$ in the new stabilisers: the desired simulation is achieved. Despite this success, the new defect is local only as long as the three surface code patches are close to each other. In order to mirror the classical embedding, we need to find an embedding that remains local regardless of the distance between the target surface codes.

As a reminder, the classical construction made use of 'data repetition codes', corresponding to the bits of the input code, and 'check repetition codes'. These latter were used to transport the bits' values and make them interact with that held in other data repetition codes. This interaction designed to implement the checks from the input code.

Considering the quantum case it is then reasonable to use 'check surface codes' in order to transport value between different 'data surface codes'. Nevertheless *interactions* between the two remains to be elucidated. Here we decide to first re-interpret the classical case from a condensation perspective, which we then use to specify the

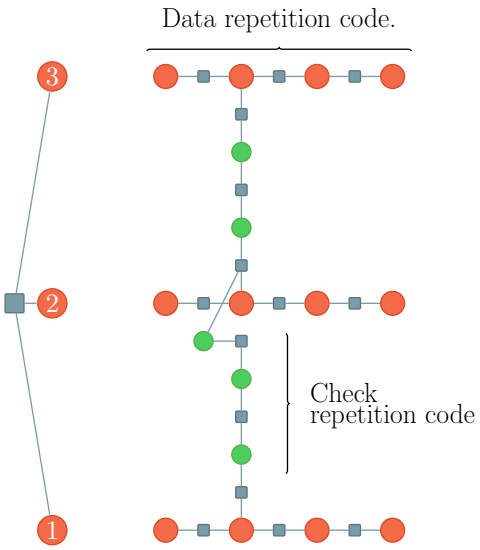


Fig. 4.18 Reminder of how the classical embedding works on ZZZ. The data repetition codes are joined by check repetition codes, and checks located at their junctions obey specific rules.

condensation rules at the junctions between data surface codes and check surface codes. Our task will be alleviated by introducing some additional notation.

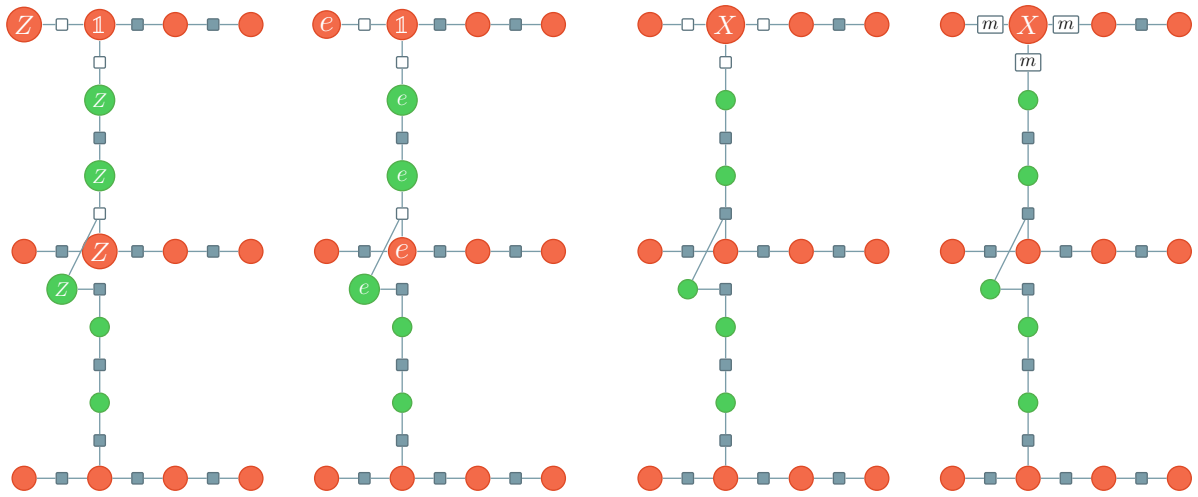
(a) Dot defect and associated condensation rules.

(b) Cross defect and associated condensation rules.

Fig. 4.19

We will see that, in a sense, the classical gadget implements a sort of combination of these defects, which we illustrate in Figure 4.20. On one hand, m excitations arise from truncated logicals, which corresponds to the previously introduced definition of excitation. On the other, there are no X-type stabilisers in a classical code, and therefore no proper e excitations to speak of. This issue can be resolved by having e excitations correspond to combinations of stabilisers. As a result, the resulting excitations of either types mutually commute; a property inherited from the operators that are derived from. Keeping an eye towards the projection to the quantum case, because of the $\mathcal{O} = \mathcal{U}$ correspondence the distinction between the two types of excitations will not be relevant.

The condensation rules derived from the gadget can be summarized by the network illustrated in Figure 4.21.



(a) Products of stabilisers leave certain operators at the proximity of the data/check junctions. These operators can be understood as e excitations.

(b) Truncating a logical to a single bit at a junction triggers certain stabilisers. These triggered stabilisers can be understood as em excitations.

Fig. 4.20 Condensation rules in the classical embedding.

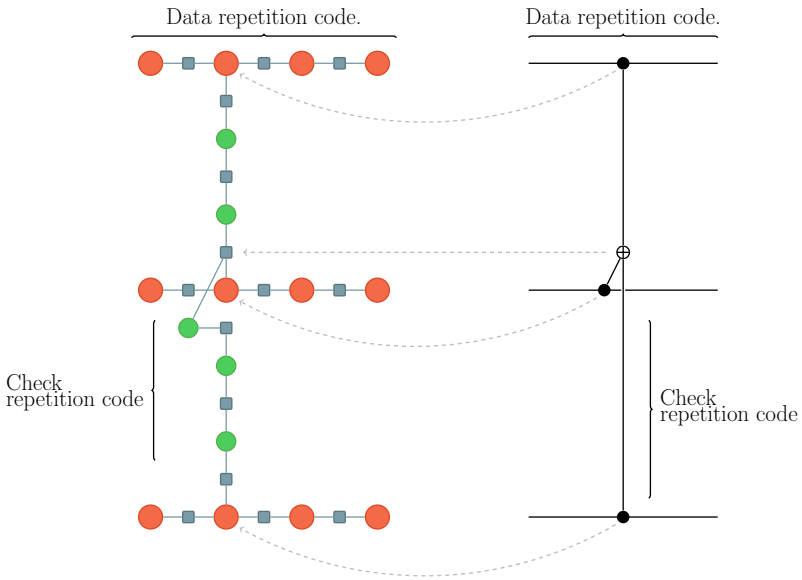


Fig. 4.21 Abstracting away the various repetition codes, and focusing on how they are joined, the classical embedding (on the left) can be expressed as a defect network (on the right).

Once the defect network is elaborated, as in Figure 4.21, we can then project these condensation rules into the quantum realm. Once we replace the repetition codes with surface codes, the result is depicted in Figure 4.22.

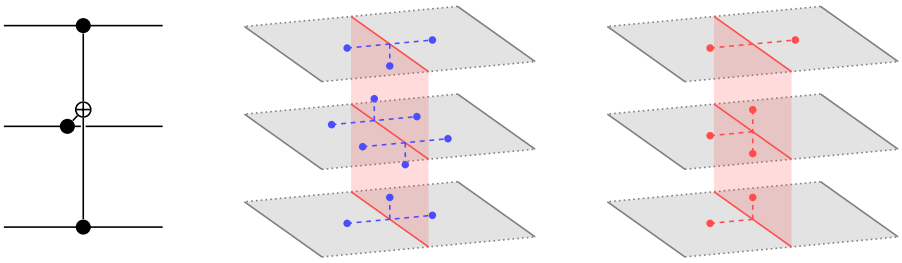


Fig. 4.22 Example of excitations condensed by the defect network.

Since the condensation rules were derived from the classical simulation, the logical operators and stabilisers follow the same algebra see Figure 4.23. The logical operators obey the same structure, while the newly introduced stabilisers can similarly be combined to obtain $Q \otimes \mathbb{1}_{\text{ancilla}}$.

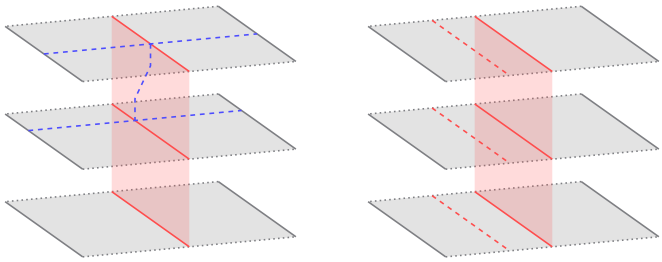


Fig. 4.23 The excitations condensed by the defect network can also be stretched into X logicals (on the left) and the stabiliser $\bar{Z}\bar{Z}\bar{Z}$ (on the right).

We refer to Section 4.3 as to the realisation of these defects, and all future ones. Generalising this simulation to ω-body checks becomes remarkably simple through the language of defect networks. As a reminder, the process for the classical simulation is summarised in Figure 4.24.

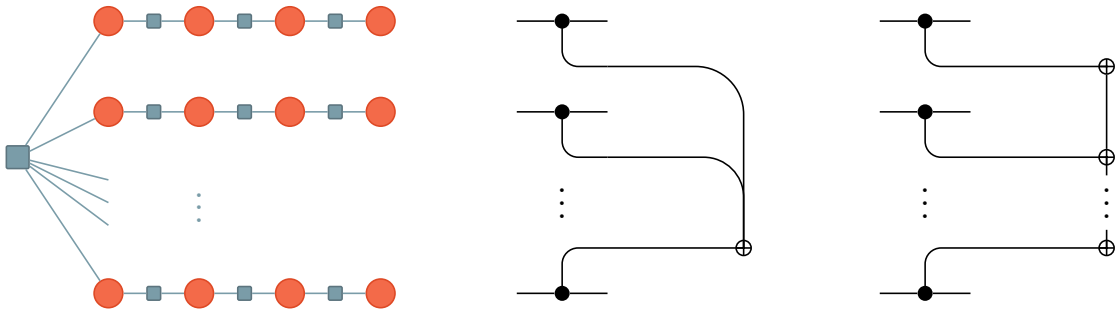


Fig. 4.24 An ω-body check (left) can similarly be expressed as a defect network (middle). That network can be broken down into a different network corresponding to the ω-body gadget of Figure 4.6 (right) .

Once we have the defect network, we can unravel the quantum equivalent, and find the condensation rules, see 4.25.

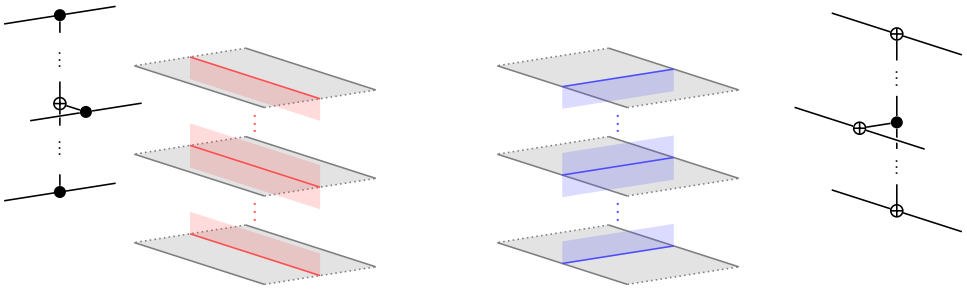


Fig. 4.25 The defect networks, translated into surface codes.

The junctions between data and check surface code make for some confusing language. Unlike the classical case, the stabilisers of the surface codes are modified in a way that leaves something distinctly different from a surface code proper. To avoid future misunderstandings, we introduce the language of data/check *layers*, see Figure 4.26.

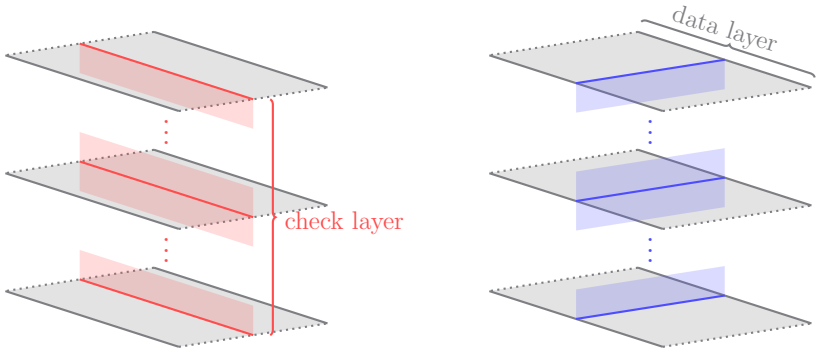


Fig. 4.26 Data and check layers.

4.2.4 Intersecting check layers

So far, we have outline a way to embed a code that relies heavily on the condensation picture of topological codes. Crucially, the condensation rules we describe only make sense if they can be realised by commuting stabilisers. Although, up to this point, this has been the case for all the examples we discussed; there remains a notable exception that ought to be addressed. When two stabilisers overlap in the original code, they yield check layers that intersect on some of the same data layers. In the case where these stabilisers are of different types (X and Z), the stabilisers on the check layers end up not commuting where they meet.

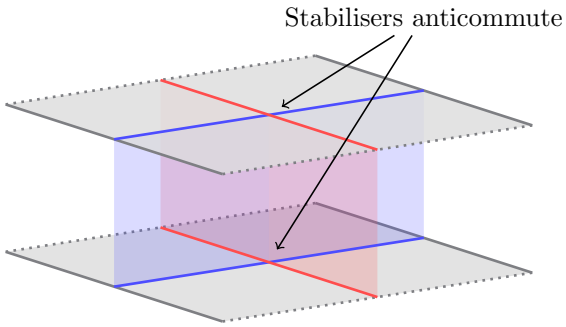


Fig. 4.27 Example of a faulty embedding of a ZZ check and an XX check.

This issue can be remediated by introducing a new defect along these intersections. Remember that the two stabilisers need to overlap on an even number of qubits, which we label $(q_1, q_2, \ldots, q_t)$. Then the new defect will run along the intersection of the check layers between the data layers corresponding to every pair $(q_1, q_2), (q_3, q_4), \ldots, (q_{t-3}, q_{t-2}), (q_{t-1}, q_t)$.

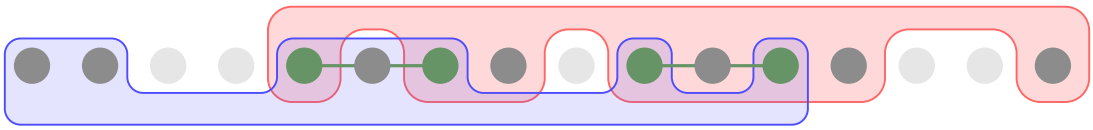


Fig. 4.28 Example of two overlapping checks of different types. The support of this overlap dictates the pairing. The first and second qubit are paired together, respectively the same applies to the third and fourth qubits.

The condensation rules corresponding to that defect are specified in the following figure.

(a) Condensation rules along two paired layers. (b) Condensation rules for two uninteracting layers.

Fig. 4.29

The embedding procedure can now be summarized as follows.

Algorithm 2 Layer code construction

Require: A CSS code $\mathcal{C}$, with parameters $[[n,k,d]]$, n_X X-type checks, n_Z Z-type checks, and maximum check weight ω.

Ensure: A CSS code that is local in 3D with parameters $[[\Theta(nn_Xn_Z), k, \frac{1}{\omega-1}d\min(n_X,n_Z)]]$.

for each qubit $i \in [0,\dots,n-1]$ **do**

Create an data layer $\mathcal{D}_i$, with the boundary conditions specified in Section 4.2.1.

end for

for each Q_j in the Z-type checks of $\mathcal{C}$ **do**

Let $i_1,\dots,i_T$ be the qubits in the support of Q_j. Create a Z-type check layer $\mathcal{Z}_j$ starting from $\mathcal{D}_{i_1}$, going through each of the $\mathcal{D}_{i_t}$ until $\mathcal{D}_{i_T}$, with the specifications given in Section 4.2.3.

end for

for each $\tilde{Q}_k$ in the X-type checks of $\mathcal{C}$ **do**

Let $i_1,\dots,i_T$ be the qubits in the support of $\tilde{Q}_k$. Create an X-type check layer $\mathcal{X}_k$ starting from $\mathcal{D}_{i_1}$, going through each of the $\mathcal{D}_{i_t}$ until $\mathcal{D}_{i_T}$, with the specifications given in Section 4.2.3.

end for

for each Q_j in the Z-type checks of $\mathcal{C}$ **do**

for each $\tilde{Q}_k$ in the X-type checks of $\mathcal{C}$ **do**

If the support of Q_j and $\tilde{Q}_k$ overlap, introduce line and point defects between the layers $\mathcal{Z}_j$ and $\mathcal{X}_k$ following the specifications of Section4.2.4.

end for

end for

4.2.5 Properties of the embedded code

There remains the question of ascertaining what the parameters of the embedded code are. Preserving our previous convention, these new parameters will be denoted $[[n',k',d']]$. The input code is denoted $\mathcal{C}$, while the new code is denoted $\mathcal{C}'$. Let L be a logical of $\mathcal{C}$, we write $\overline{L}$ the concatenation of L with surface code logicals on the data layers, see Figure 4.30.

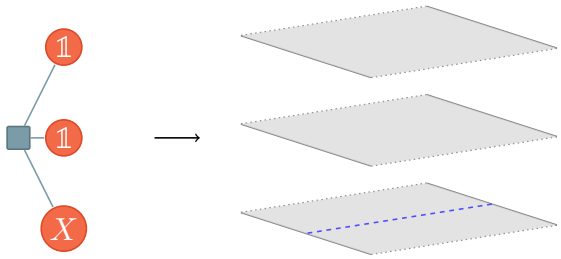


Fig. 4.30 The X_1 operator on the left, vs. $\bar{X}_1$ operator on the right. The operator $\bar{X}_1$ acts on the logical space of the surface code.

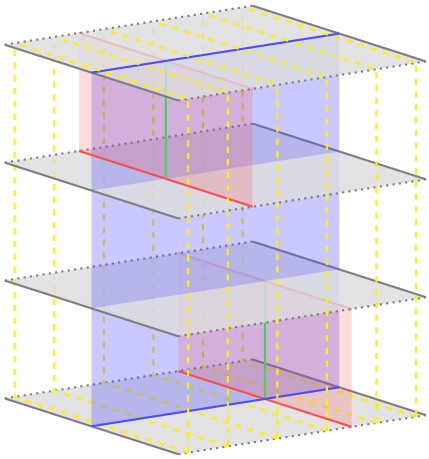
The guarantee $k' \geq k$ is straightforward to establish. The m-excitations that can be condensed by the check layers induced by Z stabilisers correspond precisely to those that commute with the e excitations these same defects condense. Our analysis of the classical case then carries over: for every X-type logical L in $\mathcal{C}$, there exists a logical $\overline{L} \otimes L_{\text{check}}$ in $\mathcal{C}'$, see Figure 4.23.

The same can be done for Z-type logical. These X- and Z-type logicals only overlap on the data layers, which guarantees that they commute if and only if they overlap on an even number of layers. This happens if and only if the original logicals overlap on an even number of qubit, and therefore they commute. In other words, given $\overline{L} \otimes L_{\text{check}}$, $\overline{L}' \otimes L'_{\text{check}}$, we have $[\overline{L} \otimes L_{\text{check}}, \overline{L}' \otimes L'_{\text{check}}] = [\overline{L}, \overline{L}'] = [L, L']$; the commutation relations are preserved. This gives us a set of logicals isomorphic to $\mathcal{P}^k$, hence $k' \geq k$.

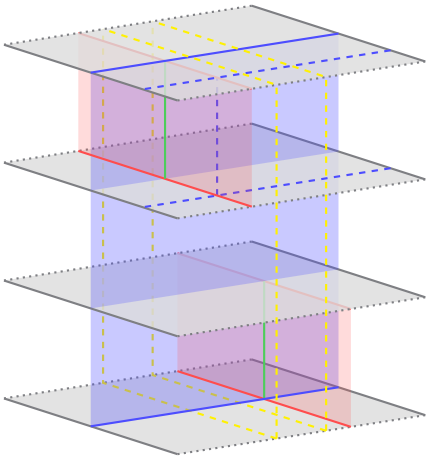
We can in fact show that $k' = k$ precisely, which will be very helpful to guarantee that d' is large. This upper bound on k' is established through counting the constraints added by the condensation rules induced by each check of $\mathcal{C}$. These rules will only condense X operators that commute with the Z check. The set of Z-type check layers quotient out as many operators off the n data layers as the set of Z checks do from the n qubits. The same argument holds for X-type checks, and yields $k' = k$.

The logicals we have found of the form $\overline{L} \otimes L_{\text{check}}$ in fact form a basis for *all* the logicals, modulo stabilisers. We will show that the weight of the operators in that basis cannot be reduced *too much*. Part of these operators can be pushed into check layers but the resulting reduction is not significant. For the rest of this section, we will focus on X-type logical operators. The code can be divided into *slabs* between Z check layers – consider the example of the $[4, 1, 2]$ code, with an X logical highlighted in blue in Figure 4.31.

Focusing on one of these slabs, we illustrated how the X check layer introduces new relations that allow for reducing the size of the logical. Understanding these new relations will be the core of our argument.

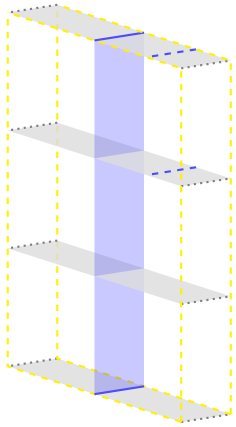


(a) An embedding dissected into 'slabs', highlighted in yellow. Each slabs contains qubits strictly between two Z-type check layers.

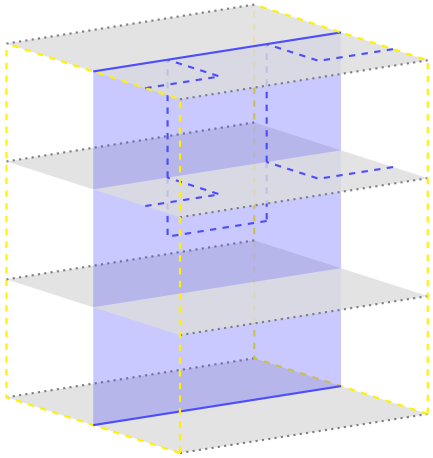


(b) A single slab, with an X operator highlighted.

Fig. 4.31



(a) Detail of a single slab.



(b) Within a slab, a logical can be pushed into the X-type check layer, potentially reducing its weight.

Fig. 4.32

Our ability to push the logical into the slab and reduce its size is a direct consequence of the condensation rules at the defect lines. Understanding exactly which rules are used will be crucial to our argument, and we illustrate them in Figure 4.33.

Each of these rules can be interpreted as a Pauli operator. For example $m_1^{anc} m_1^{data}$ corresponds to $X_1^{anc} X_1^{data}$ and so on. The existence of these rules should be understood as follows: replacing the original stabiliser $X_1X_2X_3X_4$ came at the cost of introducing

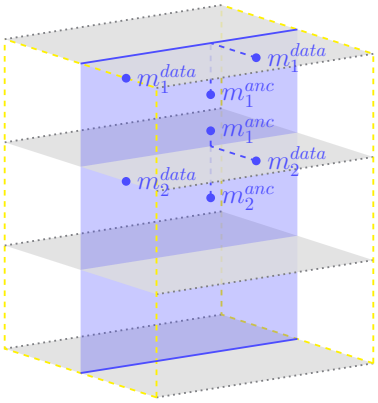


Fig. 4.33 The weight reduction of the logical observed in Figure 4.32 is due to the existence of certain condensation rules, which we make explicit here.

many smaller 'excitation stabilisers' like $X_1^{anc} X_1^{data}$ that might drive the number of excitations down – and therefore the weight of the associated logical. Nevertheless, these 'excitation stabilisers' have a an interesting property where, either they act as the original stabiliser on the data layers, or they leave at least one remaining excitation on the check layer. In order to demonstrate this property, we will leverage a projection that maps excitations to Pauli operators, see Figure 4.34.

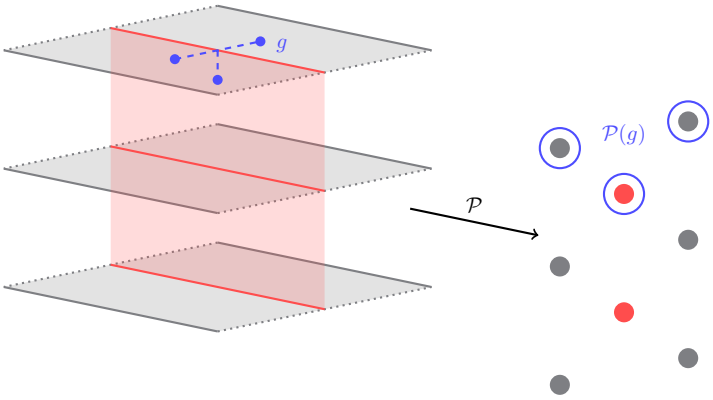


Fig. 4.34 For any excitation g condensed by a defect, we can project it down to a Pauli operator $g \to \mathcal{P}(g)$.

Lemma 11. Let $\mathcal{W}$ be the group of excitations condensed by a check layer corresponding to a stabiliser Q. Then either $g \in \mathcal{W}$ acts as Q on the data layers, or gg' has weight at least one; for any excitation g' on the data layers.

Proof. We assume that g is an m-type excitation and has no support on the check layer. Further, assume that g does not have any element of the form $m_i^l m_i^r$ following

the notation of Figure 4.16, otherwise the result follows. We write $\mathcal{P}(g)$ the Pauli operator corresponding to g, see Figure 4.34. By construction, $\mathcal{P}(g)$ commutes with P' if $[Q,P']=1$. However, if $\mathcal{P}(g)\neq Q$, then there exists P_* such that $[Q,P_*]=1$, yet $[\mathcal{P}(g),P_*]=-1$. □

Put it simply, these smaller stabilisers cannot have a dramatic effect on the logicals. They can either act as a stabiliser of the original code, or *somewhat* reduce their size. At the worse, each check layer takes $|\mathrm{supp}(s)|-1$ excitations and maps them to one. We conclude that in the worst case an operator L can be reduced to L' where $|\mathrm{supp}(L')|\geq\frac{|\mathrm{supp}(L)|}{\omega-1}$.

Lemma 12. The distance d' of the output code satisfies $d'\geq\frac{d\cdot\min(m_Z,m_X)}{\omega-1}$.

Proof. In the selected basis of logicals $\bar{L}_i\otimes L_i^{ancilla}$, each operator has size at least $\mathrm{supp}(\bar{L}_i)\geq d\cdot\min(m_Z,m_X)$. And this size can be reduced by a factor at most $\frac{1}{\omega-1}$, as argued above. □

This last lemma can be understood as the logical being 'shackled' to every slab. This intuitively makes sense given that in every slab there exists a Z-type logical that anti-commutes with the given X-type logical. Because of this presence, the X-type logical cannot leave the slab, lest it stops anti-commuting. This situation is very similar to the logicals of the surface code. The logical $\overline{Z}$ has support on every row of the lattice, because there is a representative of $\overline{X}$ contained strictly on that row. See Figure 4.33.

4.3 Open Questions

1. In this chapter we frequently referred to the idea of a simulation to motivate our construction; disappointingly however, we never came up with a substantive definition that carried over to the quantum case. One might find more success exploring exotic definitions, like the 'soundness/completeness' conditions of [51].

2. The difference between the scaling of the classical versus the quantum BPT bound seems to be entirely a consequence of the repetition code being a one-dimensional code, version the surface code being a two-dimensional code. This could hide a deeper connection. For example, we need at least two dimensions to protect the two of X and Z [52]. Does there exist a non-embedding construction that saturates the BPT bound?

3. The surface code features prominently in our result. However, there exist 2D codes with better parameters [53, 54], could these codes yield more efficient embeddings?

4. The relationship between condensation rules and induced stabilisers is a somewhat novel topic, which asks for more careful inquiries. For example the group $\mathcal{L}$ of condensed excitations seems to correspond to its own normaliser, i.e. if it can be condensed, it commutes with the elements of $\mathcal{L}$ and vice versa. In other words, since condensation rules induce new stabilisers, they have to commute with the other condensed excitations.

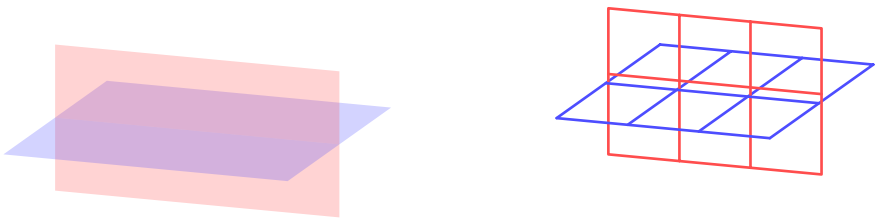
Appendix 4.A Microscopic detail of the defects

In this section, we enumerate the defects, and their realisation as stabilisers.

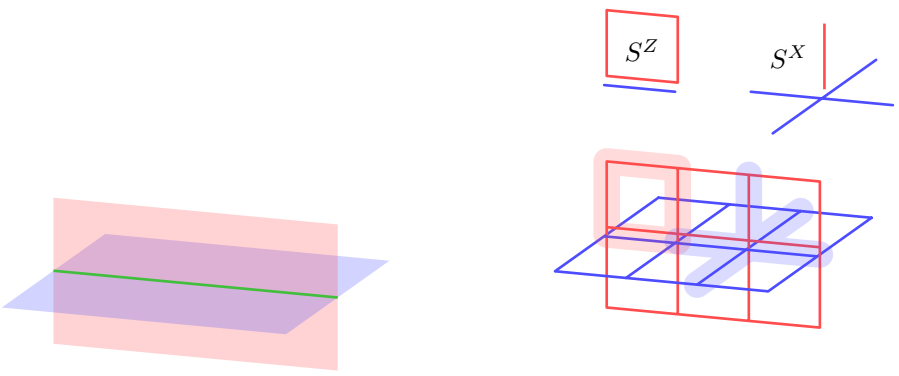
4.A.1 Line Defect Checks

There exist 8 different types of line bulk defects.

1. Two unpaired check layers intersect trivially: their stabilisers are unchanged.

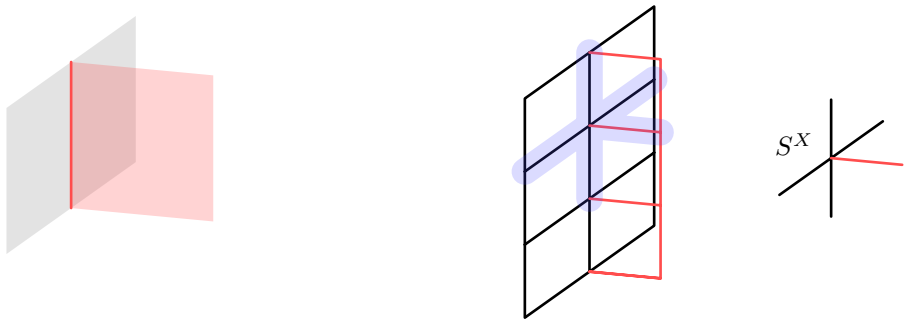


2. When two check layers are paired, the green line defect is realised by the following modified stabilisers.

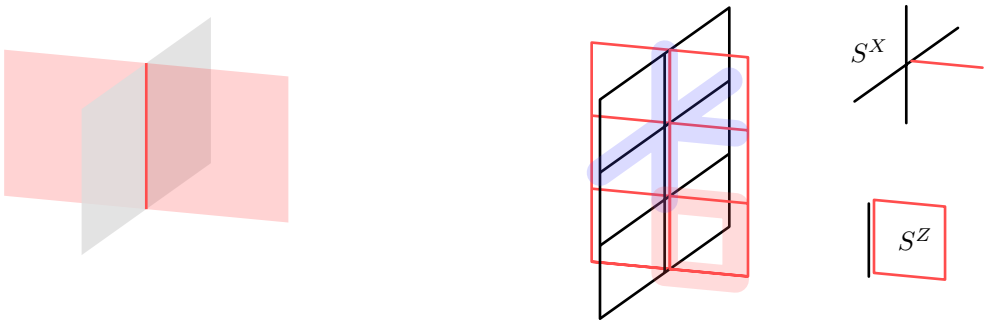


In this figure, and those below, the S^Z stabilizer on the region highlighted in red has support on the edges that are depicted adjacent to it. The S^X stabilizer on the region highlighted in blue is specified similarly.

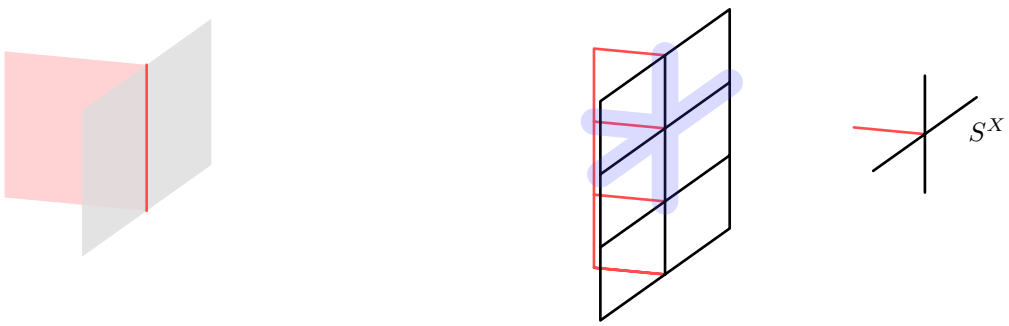
3. A Z-type check surface code beginning at a data surface code.



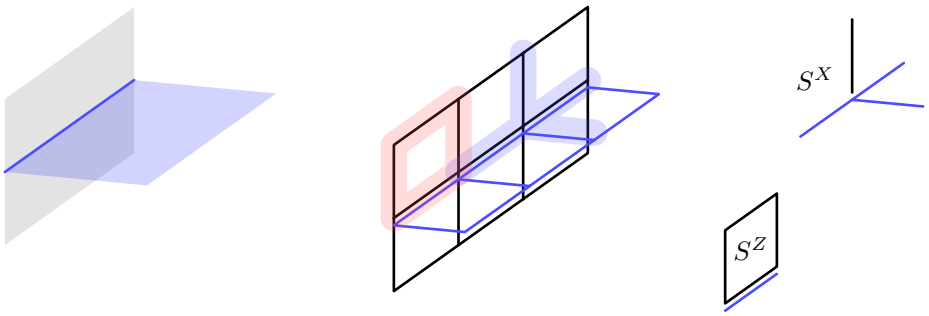
4. A Z-type check surface code intersecting with a data surface code.



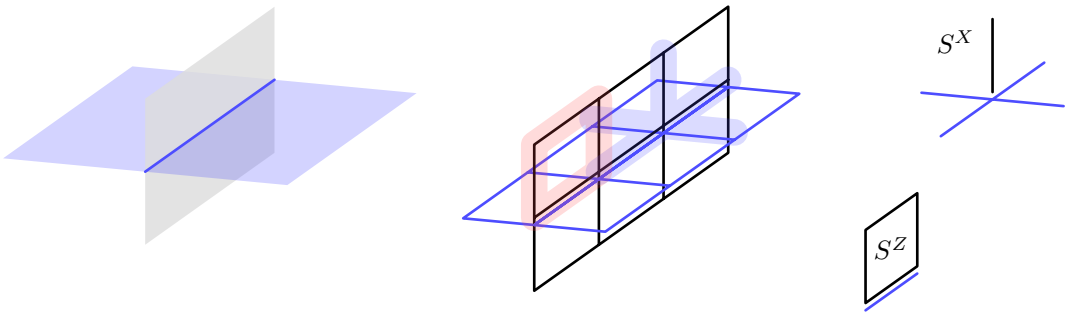
5. A Z-type check surface code terminating at a data surface code.



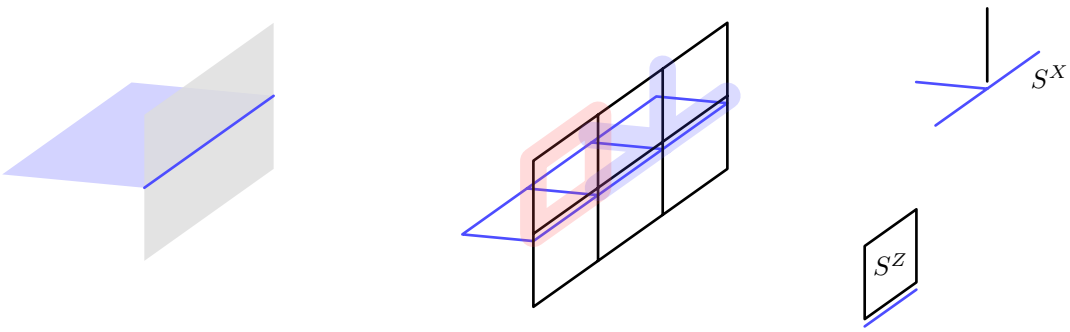
6. An X-type check surface code beginning at a data surface code.



7. An X-type check surface code intersecting with a data surface code.



8. An X-type check surface code terminating at a data surface code.



This concludes the enumeration of line defect lattice checks.

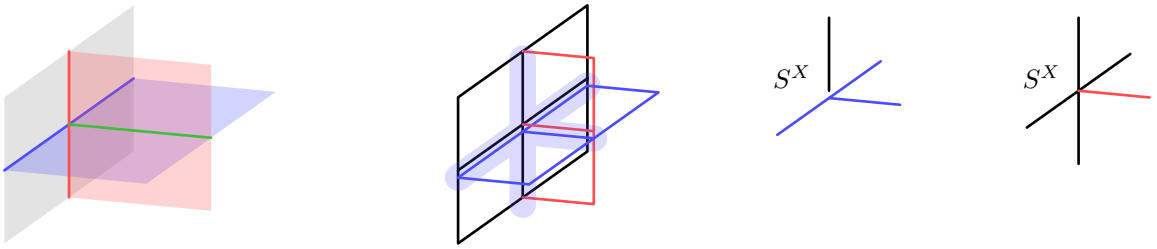
4.A.2 Point Defect Checks

In addition to the above line defects, there are 10 types of point defects in the bulk and 12 types of point defects on the boundary of the cuboid.

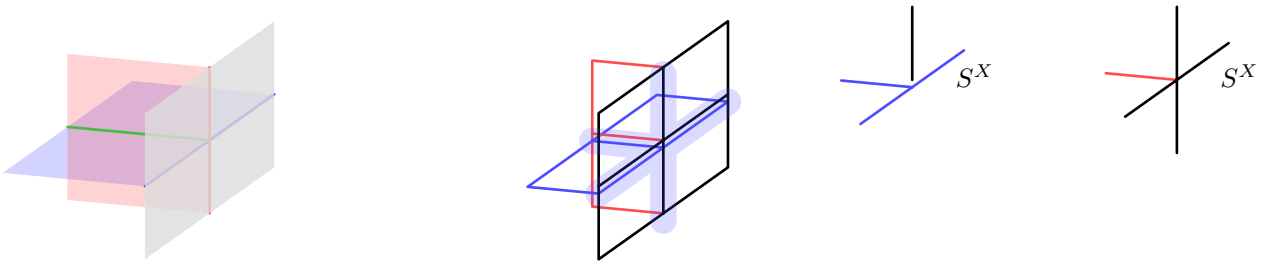
Bulk Point Defects

The 10 point defects in the bulk are listed below.

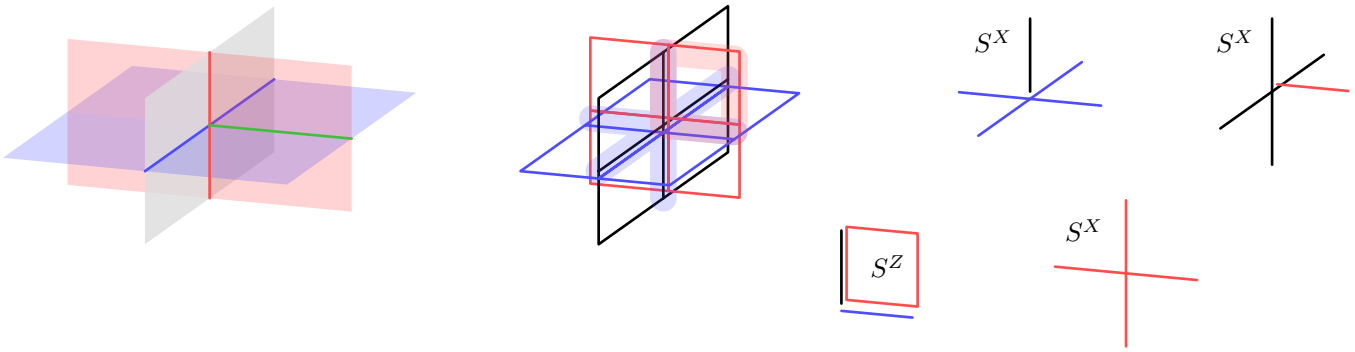
1. The first type of bulk point defect occurs when two pared check layers begin at a data surface code.



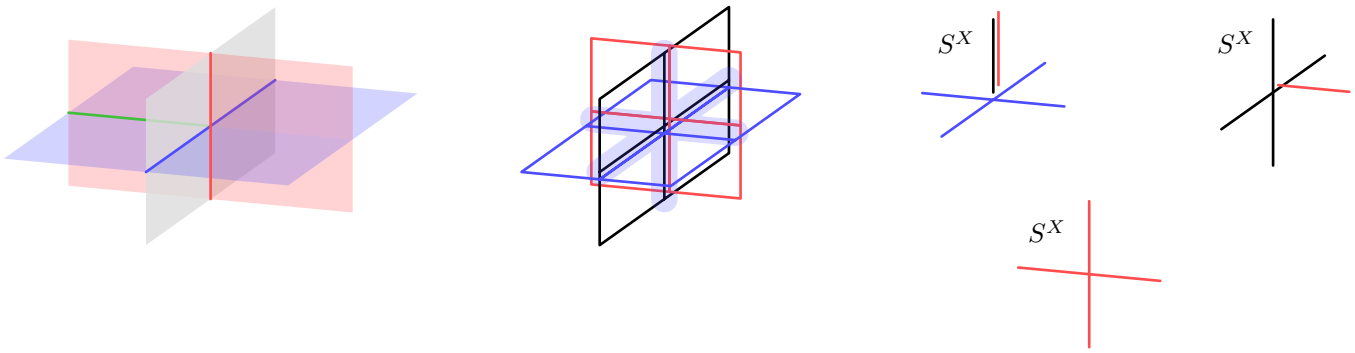
2. The first type of bulk point defect occurs when two pared check layers terminates at a data surface code.



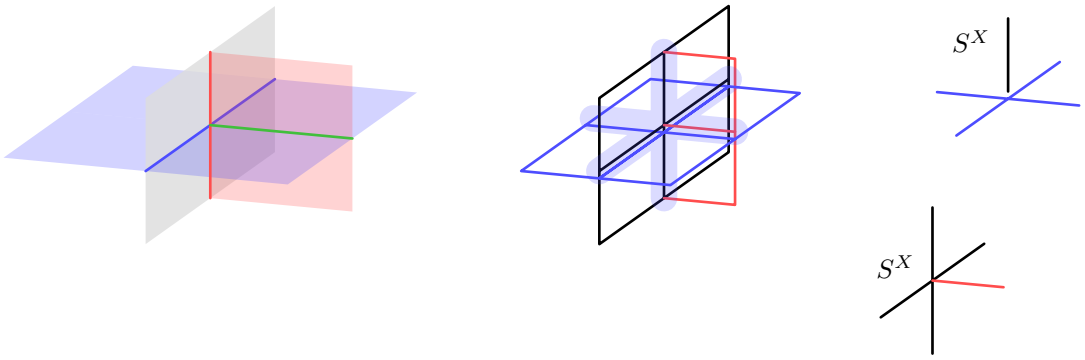
3. The third type of bulk point defect occurs when two check layers become paired after intersecting with a data surface code.



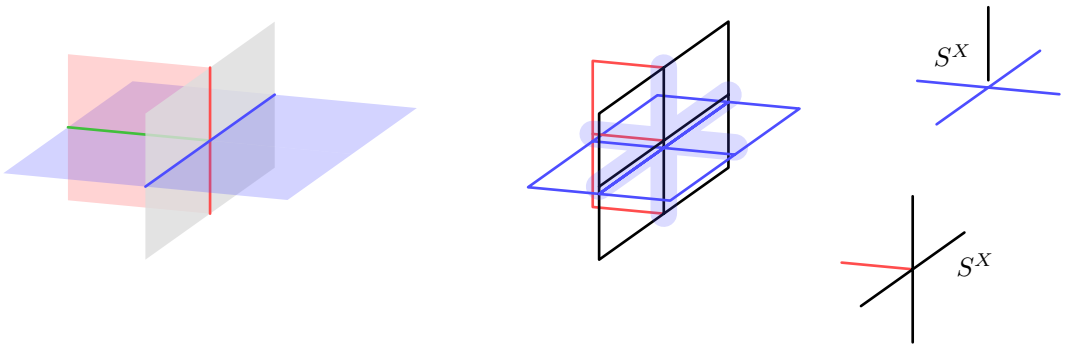
4. The fourth type of bulk point defect occurs when two check layers stop being paired after intersecting with a data surface code.



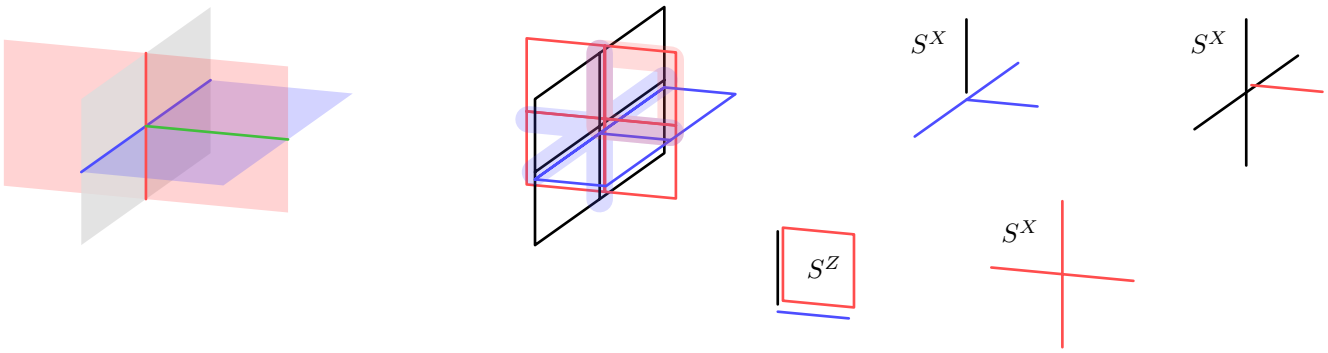
5. The fourth type of bulk point defect occurs when an X layer starts being paired with a Z layer that begins on the surface code layer.



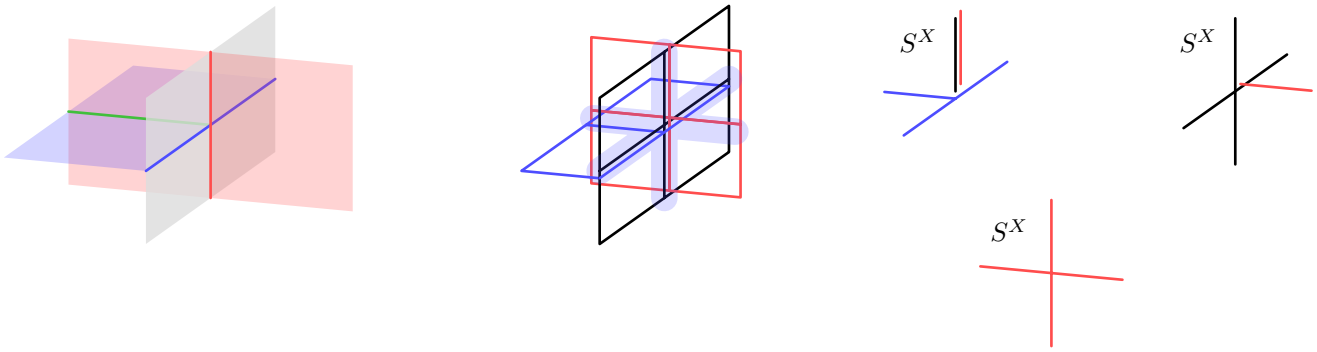
6. The sixth type of bulk point defect is the reflection of the above defect.



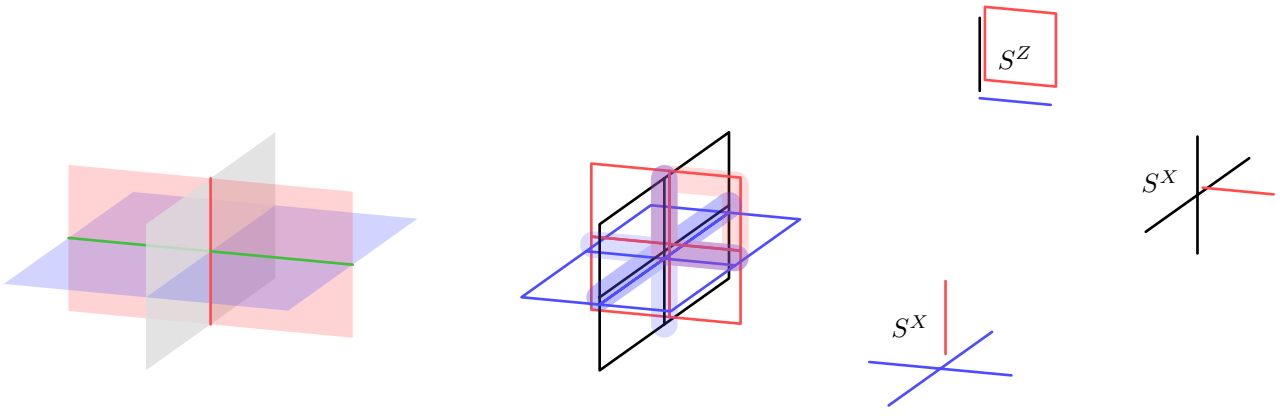
7. The seventh type of bulk point defect occurs when an Z layer starts being paired with a X layer that begins on the surface code layer.



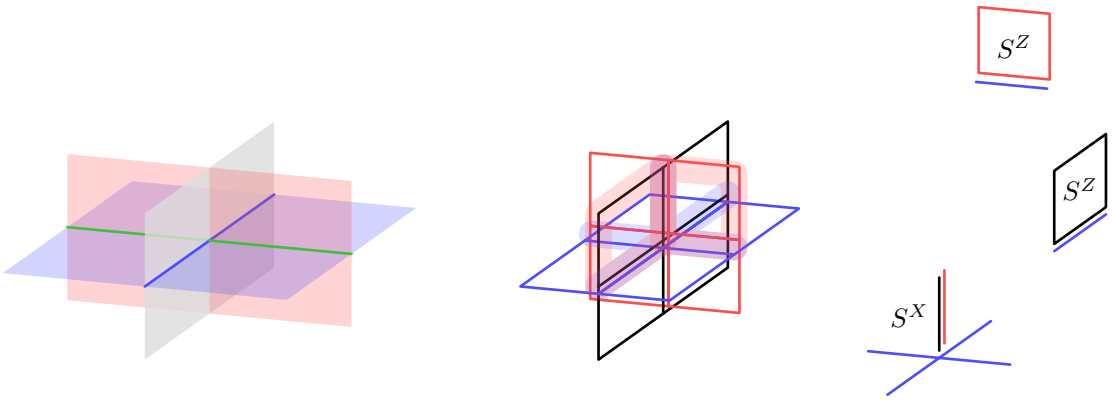
8. The eighth type of bulk point defect is the reflection of the above defect.



9. The ninth type of bulk point defect occurs when an Z layer intersects with a data surface code, while being paired with an X check layer that does not intersect with that surface code.



10. The tenth type of bulk point defect occurs when an X layer intersects with a data surface code, while being paired with an Z check layer that does not intersect with that surface code.

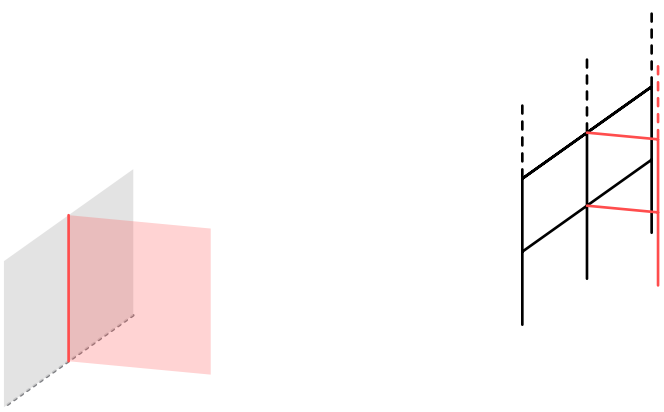


4.A.3 Boundary Point Defects

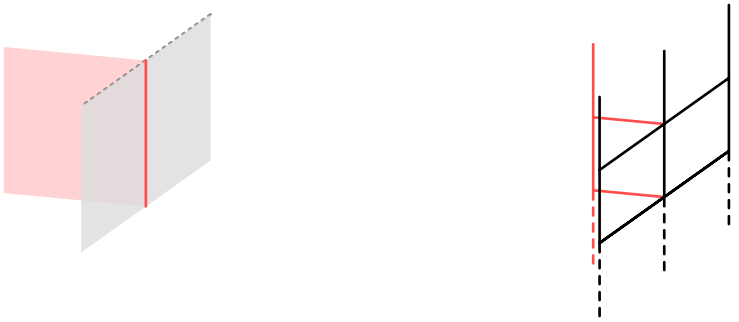
The 12 boundary point defects are listed below.

1. The first type of boundary point defect happens at the top of the intersection of a data surface code and a Z check layer beginning on that surface code.

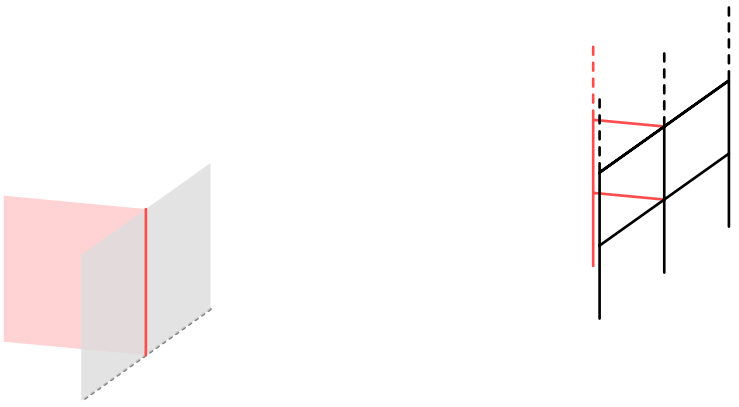
2. The second type of boundary point defect is the reflection of the previous defect.



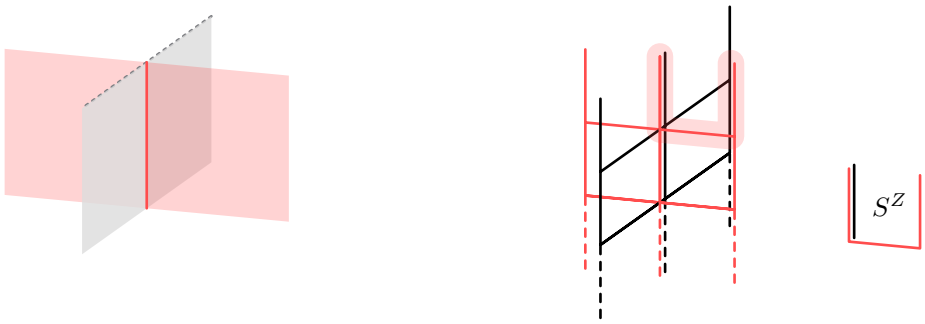
3. The third type of boundary point defect happens at the top of the intersection of a data surface code and a Z check layer terminating on that surface code.



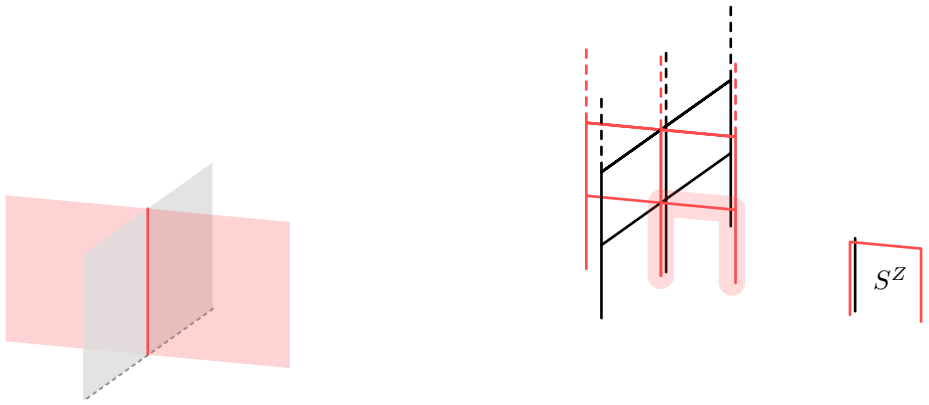
4. The fourth type of boundary point defect is the reflection of the previous defect.



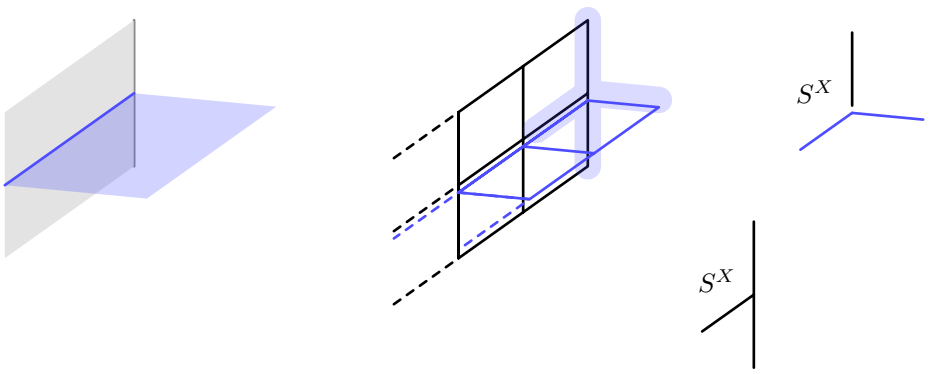
5. The fifth type of boundary point defect happens at the top of the intersection of a data surface code and a Z check layer intersecting with that surface code.



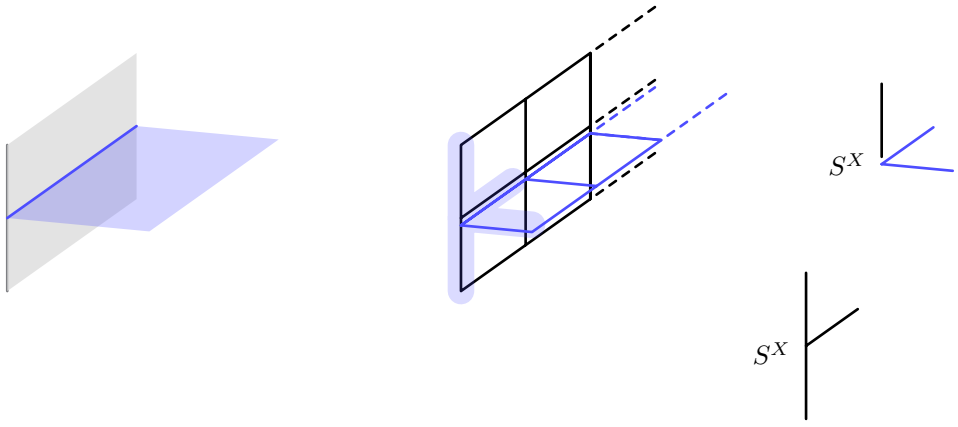
6. The sixth type of boundary point defect is the reflection of the previous defect.



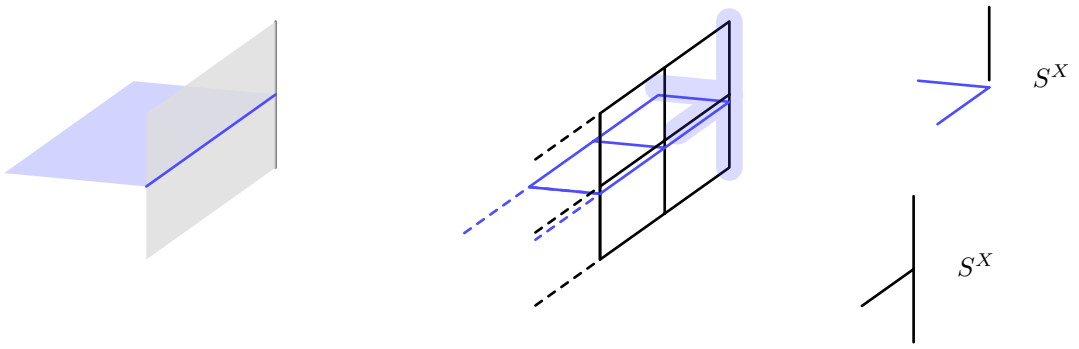
7. The seventh type of boundary point happens at the top of the intersection of a data surface code and an X check layer beginning on that surface code.



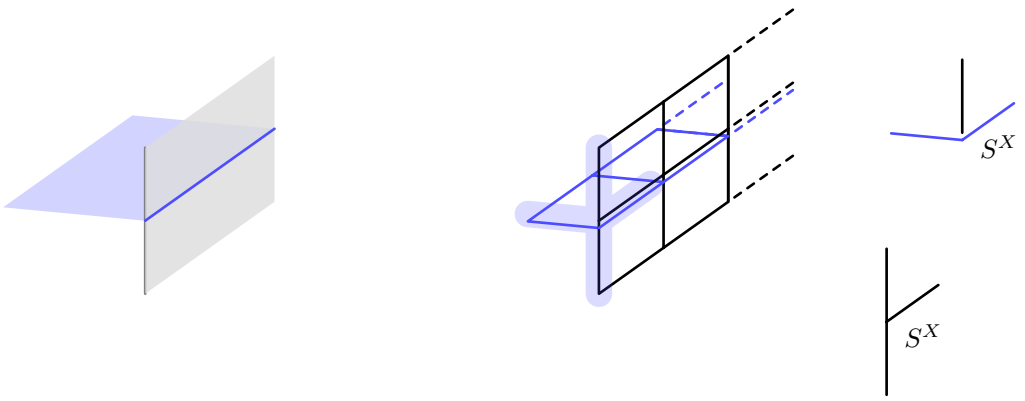
8. The eighth type of boundary point defect is the reflection of the previous defect.



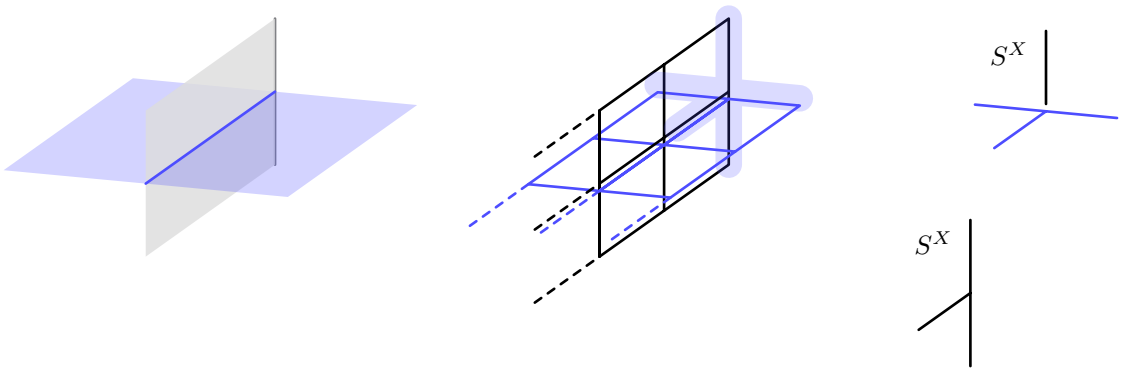
9. The ninth type of boundary point defect happens at the top of the intersection of a data surface code and an X check layer terminating on that surface code.



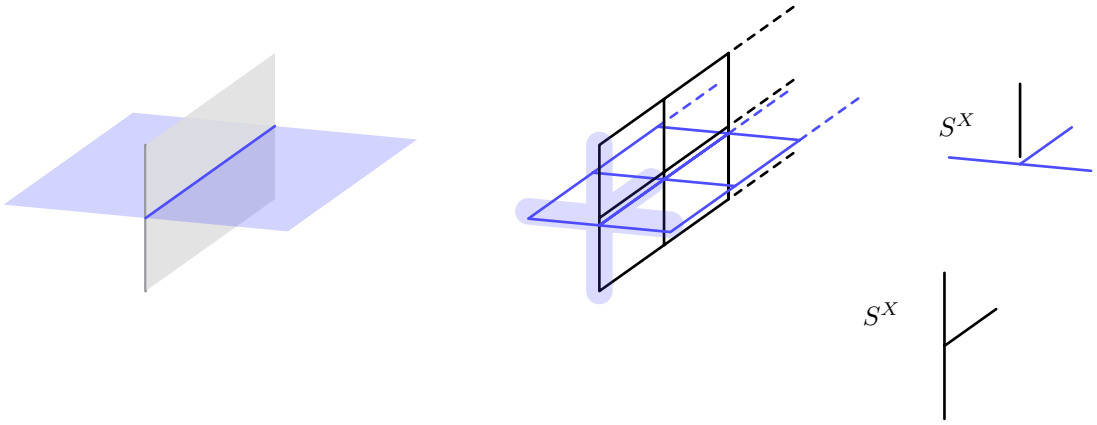
10. The tenth type of boundary point defect is the reflection of the previous defect.



11. The eleventh type of boundary point defect happens at the top of the intersection of a data surface code and an X check layer intersecting with that surface code.



12. The twelfth type of boundary point defect is the reflection of the previous defect.


S^X
S^X

5 | Conclusion

In this thesis we have investigated the interplay between locality and (quantum) error correction. We have demonstrated this interaction is multifaceted, and naturally spans at least multiple subfields of maths and physics. After 20 years since the birth of QEC, this topic is suddenly moving at an elevated pace, and results like [55] are becoming more frequent.

There is yet much to explore an we hope the open question we have sketched throughout this thesis will be useful jumping off points for future research.

Bibliography

[1] David Poulin. "Stabilizer Formalism for Operator Quantum Error Correction". In: *Phys. Rev. Lett.* 95 (23 2005), p. 230504. DOI: 10.1103/PhysRevLett.95.230504. URL: https://link.aps.org/doi/10.1103/PhysRevLett.95.230504.

[2] David Aasen et al. "Measurement Quantum Cellular Automata and Anomalies in Floquet Codes". 2023. DOI: 10.48550/ARXIV.2304.01277. URL: https://arxiv.org/abs/2304.01277.

[3] Beni Yoshida and Isaac L. Chuang. "Framework for classifying logical operators in stabilizer codes". In: *Phys. Rev. A* 81 (5 2010), p. 052302. DOI: 10.1103/PhysRevA.81.052302. URL: https://link.aps.org/doi/10.1103/PhysRevA.81.052302.

[4] Emanuel Knill, Raymond Laflamme, and Lorenza Viola. "Theory of Quantum Error Correction for General Noise". In: *Phys. Rev. Lett.* 84 (11 2000), pp. 2525–2528. DOI: 10.1103/PhysRevLett.84.2525. URL: https://link.aps.org/doi/10.1103/PhysRevLett.84.2525.

[5] Michael A Nielsen and Isaac Chuang. "Quantum computation and quantum information". 2002.

[6] Charles H. Bennett et al. "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels". In: *Phys. Rev. Lett.* 70 (13 1993), pp. 1895–1899. DOI: 10.1103/PhysRevLett.70.1895. URL: https://link.aps.org/doi/10.1103/PhysRevLett.70.1895.

[7] David Fattal et al. “Entanglement in the stabilizer formalism”. 2004. DOI: 10.48550/ARXIV.QUANT-PH/0406168. URL: https://arxiv.org/abs/quant-ph/0406168.

[8] Marco Tomamichel, Roger Colbeck, and Renato Renner. “A Fully Quantum Asymptotic Equipartition Property”. In: *IEEE Transactions on Information Theory* 55.12 (Dec. 2009), 5840–5847. ISSN: 1557-9654. DOI: 10.1109/tit.2009.2032797. URL: http://dx.doi.org/10.1109/TIT.2009.2032797.

[9] John C. Baez. “What is Entropy?” 2024. arXiv: 2409.09232 [cond-mat.stat-mech]. URL: https://arxiv.org/abs/2409.09232.

[10] Benjamin Schumacher. “Quantum coding”. In: *Phys. Rev. A* 51 (4 1995), pp. 2738–2747. DOI: 10.1103/PhysRevA.51.2738. URL: https://link.aps.org/doi/10.1103/PhysRevA.51.2738.

[11] Robert König, Renato Renner, and Christian Schaffner. “The operational meaning of min- and max-entropy”. In: *IEEE Trans. Inf. Theor.* 55.9 (Sept. 2009), 4337–4347. ISSN: 0018-9448. DOI: 10.1109/TIT.2009.2025545. URL: https://doi.org/10.1109/TIT.2009.2025545.

[12] Josh Cadney et al. “Inequalities for the ranks of multipartite quantum states”. In: *Linear Algebra and its Applications* 452 (2014), pp. 153–171. ISSN: 0024-3795. DOI: https://doi.org/10.1016/j.laa.2014.03.035. URL: https://www.sciencedirect.com/science/article/pii/S0024379514001797.

[13] Felix Leditzky, Nilanjana Datta, and Graeme Smith. “Useful States and Entanglement Distillation”. In: *IEEE Transactions on Information Theory* 64.7 (2018), pp. 4689–4708. DOI: 10.1109/TIT.2017.2776907.

[14] Sergey Bravyi, David Poulin, and Barbara Terhal. “Tradeoffs for reliable quantum information storage in 2D systems”. In: *Physical Review Letters* 104.5 (2010), p. 050503.

[15] Mark M. Wilde. *Quantum Information Theory*. Cambridge University Press, 2013.

[16] Ainesh Bakshi et al. "High-Temperature Gibbs States are Unentangled and Efficiently Preparable". 2024. arXiv: 2403.16850 [quant-ph]. URL: https://arxiv.org/abs/2403.16850.

[17] Anurag Anshu and Chinmay Nirkhe. "Circuit Lower Bounds for Low-Energy States of Quantum Code Hamiltonians". In: *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Ed. by Mark Braverman. Vol. 215. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 6:1–6:22. ISBN: 978-3-95977-217-4. DOI: 10.4230/LIPIcs.ITCS.2022.6. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2022.6.

[18] Shachar Lovett and Emanuele Viola. "Bounded-Depth Circuits Cannot Sample Good Codes". In: *computational complexity* 21.2 (Mar. 2012), 245–266. ISSN: 1420-8954. DOI: 10.1007/s00037-012-0039-3. URL: http://dx.doi.org/10.1007/s00037-012-0039-3.

[19] Sergey Bravyi and Alexei Yu Kitaev. "Quantum codes on a lattice with boundary". In: *arXiv preprint quant-ph/9811052* (1998).

[20] A Yu Kitaev. "Fault-tolerant quantum computation by anyons". In: *Annals of Physics* 303.1 (2003), pp. 2–30.

[21] Eric Chitambar et al. "Everything You Always Wanted to Know About LOCC (But Were Afraid to Ask)". In: *Communications in Mathematical Physics* 328.1 (Mar. 2014), 303–326. ISSN: 1432-0916. DOI: 10.1007/s00220-014-1953-9. URL: http://dx.doi.org/10.1007/s00220-014-1953-9.

[22] Fernando G. S. L. Brandão, Matthias Christandl, and Jon Yard. "Faithful Squashed Entanglement". In: *Communications in Mathematical Physics* 306.3 (Aug. 2011), 805–830. ISSN: 1432-0916. DOI: 10.1007/s00220-011-1302-1. URL: http://dx.doi.org/10.1007/s00220-011-1302-1.

[23] Eric Chitambar, Wei Cui, and Hoi-Kwong Lo. "Increasing Entanglement Monotones by Separable Operations". In: *Phys. Rev. Lett.* 108 (24 2012), p. 240504. DOI: 10.1103/PhysRevLett.108.240504. URL: https://link.aps.org/doi/10.1103/PhysRevLett.108.240504.

[24] T. Ogawa and H. Nagaoka. "Strong converse and Stein's lemma in quantum hypothesis testing". In: *IEEE Transactions on Information Theory* 46.7 (2000), pp. 2428–2433. DOI: 10.1109/18.887855.

[25] Nouédyn Baspin, Omar Fawzi, and Ala Shayeghi. "A lower bound on the overhead of quantum error correction in low dimensions". 2023. arXiv: 2302.04317 [quant-ph]. URL: https://arxiv.org/abs/2302.04317.

[26] Ángela Capel, Angelo Lucia, and David Pérez-García. "Superadditivity of Quantum Relative Entropy for General States". In: *IEEE Transactions on Information Theory* 64.7 (2018), pp. 4758–4765. DOI: 10.1109/TIT.2017.2772800.

[27] S. Bravyi, M. B. Hastings, and F. Verstraete. "Lieb-Robinson Bounds and the Generation of Correlations and Topological Quantum Order". In: *Phys. Rev. Lett.* 97 (5 2006), p. 050401. DOI: 10.1103/PhysRevLett.97.050401. URL: https://link.aps.org/doi/10.1103/PhysRevLett.97.050401.

[28] Miguel Aguado and Guifré Vidal. "Entanglement Renormalization and Topological Order". In: *Phys. Rev. Lett.* 100 (7 2008), p. 070404. DOI: 10.1103/PhysRevLett.100.070404. URL: https://link.aps.org/doi/10.1103/PhysRevLett.100.070404.

[29] Nicolas Delfosse, Michael E. Beverland, and Maxime A. Tremblay. "Bounds on stabilizer measurement circuits and obstructions to local implementations of quantum LDPC codes". 2021. arXiv: 2109.14599 [quant-ph]. URL: https://arxiv.org/abs/2109.14599.

[30] Nicolas Delfosse and Gilles Zémor. "Upper bounds on the rate of low density stabilizer codes for the quantum erasure channel". In: *Quantum Info. Comput.* 13.9–10 (Sept. 2013), 793–826. ISSN: 1533-7146.

[31] Nouédyn Baspin and Anirudh Krishna. "Connectivity constrains quantum codes". In: *Quantum* 6 (2022), p. 711. DOI: 10.22331/q-2022-05-13-711. URL: https://doi.org/10.22331/q-2022-05-13-711.

[32] Nouédyn Baspin and Anirudh Krishna. "Quantifying Nonlocality: How Outperforming Local Quantum Codes Is Expensive". In: *Physical Review Letters* 129.5 (July 2022). DOI: 10.1103/physrevlett.129.050505. URL: https://doi.org/10.1103/physrevlett.129.050505.

[33] Nouédyn Baspin et al. "Improved rate-distance trade-offs for quantum codes with restricted connectivity". In: *Quantum Science and Technology* 10.1 (2024), p. 015021. DOI: 10.1088/2058-9565/ad8370. URL: https://dx.doi.org/10.1088/2058-9565/ad8370.

[34] Julia Chuzhoy. "Routing in Undirected Graphs with Constant Congestion". In: *SIAM Journal on Computing* 45.4 (2016), pp. 1490–1532. DOI: 10.1137/130910464. eprint: https://doi.org/10.1137/130910464. URL: https://doi.org/10.1137/130910464.

[35] Julia Böttcher et al. "Bandwidth, expansion, treewidth, separators and universality for bounded-degree graphs". In: *European Journal of Combinatorics* 31.5 (2010), pp. 1217–1227.

[36] Shayan Oveis Gharan and Luca Trevisan. " Approximating the Expansion Profile and Almost Optimal Local Graph Clustering ". In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2012, pp. 187–196. DOI: 10.1109/FOCS.2012.85. URL: https://doi.ieeecomputersociety.org/10.1109/FOCS.2012.85.

[37] Nouédyn Baspin. "On combinatorial structures in linear codes". 2023. arXiv: 2309.16411 [cs.IT]. URL: https://arxiv.org/abs/2309.16411.

[38] Nouédyn Baspin and Dominic Williamson. "Wire Codes". 2024. arXiv: 2410.10194 [quant-ph]. URL: https://arxiv.org/abs/2410.10194.

[39] Jeongwan Haah. "Local stabilizer codes in three dimensions without string logical operators". In: *Physical Review A* 83.4 (2011). ISSN: 1094-1622. DOI: 10.1103/physreva.83.042330. URL: http://dx.doi.org/10.1103/PhysRevA.83.042330.

[40] Kamil Michnicki. "3-d quantum stabilizer codes with a power law energy barrier". 2012. arXiv: 1208.3496 [quant-ph]. URL: https://arxiv.org/abs/1208.3496.

[41] Anthony Leverrier and Gilles Zémor. "Quantum Tanner codes". In: *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. 2022, pp. 872–883. DOI: 10.1109/FOCS54457.2022.00117.

[42] Pavel Panteleev and Gleb Kalachev. "Asymptotically good Quantum and locally testable classical LDPC codes". In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2022. Association for Computing Machinery, 2022, 375–388. ISBN: 9781450392648. DOI: 10.1145/3519935.3520017. URL: https://doi.org/10.1145/3519935.3520017.

[43] Beni Yoshida. "Information storage capacity of discrete spin systems". In: *Annals of Physics* 338 (2013), pp. 134–166. ISSN: 0003-4916. DOI: https://doi.org/10.1016/j.aop.2013.07.009. URL: https://www.sciencedirect.com/science/article/pii/S0003491613001693.

[44] Elia Portnoy. "Local Quantum Codes from Subdivided Manifolds". 2023. arXiv: 2303.06755 [quant-ph]. URL: https://arxiv.org/abs/2303.06755.

[45] Ting-Chun Lin, Adam Wills, and Min-Hsiu Hsieh. "Geometrically Local Quantum and Classical Codes from Subdivision". 2024. arXiv: 2309.16104 [quant-ph]. URL: https://arxiv.org/abs/2309.16104.

[46] Xingjian Li, Ting-Chun Lin, and Min-Hsiu Hsieh. "Transform Arbitrary Good Quantum LDPC Codes into Good Geometrically Local Codes in Any Dimension". 2024. arXiv: 2408.01769 [quant-ph]. URL: https://arxiv.org/abs/2408.01769.

[47] Andrew C. Yuan. "Unified Framework for Quantum Code Embedding". 2025. arXiv: 2507.05361 [quant-ph]. URL: https://arxiv.org/abs/2507.05361.

[48] Dave Bacon et al. "Sparse quantum codes from quantum circuits". In: *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*. 2015, pp. 327–334.

[49] Dominic J. Williamson and Nouédyn Baspin. "Layer codes". In: *Nature Communications* 15.1 (Nov. 2024). ISSN: 2041-1723. DOI: 10.1038/s41467-024-53881-3. URL: http://dx.doi.org/10.1038/s41467-024-53881-3.

[50] Jean-Pierre Tillich and Gilles Zémor. "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 (2014), pp. 1193–1202.

[51] Harriet Apel and Nouédyn Baspin. "Simulating Sparse Hamiltonians on 2D Lattices". In: *Phys. Rev. Lett.* 134 (17), p. 170602. DOI: 10.1103/PhysRevLett.134.170602. URL: https://link.aps.org/doi/10.1103/PhysRevLett.134.170602.

[52] Nouédyn Baspin. "Stabilizer codes of less than two dimensions have constant distance". 2025. arXiv: 2503.17655 [quant-ph]. URL: https://arxiv.org/abs/2503.17655.

[53] Zijian Liang, Jens Niklas Eberhardt, and Yu-An Chen. "Planar quantum low-density parity-check codes with open boundaries". 2025. arXiv: 2504.08887 [quant-ph]. URL: https://arxiv.org/abs/2504.08887.

[54] Vincent Steffan et al. "Tile Codes: High-Efficiency Quantum Codes on a Lattice with Boundary". 2025. arXiv: 2504.09171 [quant-ph]. URL: https://arxiv.org/abs/2504.09171.

[55] Samuel Dai, Ray Li, and Eugene Tang. "Optimal Locality and Parameter Tradeoffs for Subsystem Codes". 2025. arXiv: 2503.22651 [quant-ph]. URL: https://arxiv.org/abs/2503.22651.