LICENTIATE THESIS

Topological stabilizer code on a honeycomb lattice

BASUDHA SRIVASTAVA

Department of Physics
University of Gothenburg
Göteborg, Sweden 2022

Topological stabilizer code on a honeycomb lattice
Basudha Srivastava

ISBN 978-91-8009-845-8 (PRINT)
ISBN 978-91-8009-846-5 (PDF)

This thesis is electronically published, available at
https://hdl.handle.net/2077/71633

Department of Physics
University of Gothenburg
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-786 00 00

Printed by Kompendiet
Göteborg, Sweden 2022

ABSTRACT

Quantum systems are adversely affected by noise due to interactions with the environment. Quantum error correction is a technique that relies on the principle of redundancy to encode logical information in additional qubits to better protect the system against noise, and is required in order to design a viable quantum computer. One of the most popular classes of quantum error-correcting codes are topological stabilizer codes, which use repeated local measurements to detect and correct errors on the code.

In this thesis, we present a novel topological stabilizer code, the XYZ^2 code, which is implemented on a hexagonal grid of qubits and encodes a logical qubit with the help of weight-six and weight-two stabilizer measurements. This code has the advantage of having a quadratic distance ($2d^2$) for pure Z noise and pure Y noise, where d is the minimum distance of the code that utilizes $2d^2$ physical qubits. The code demonstrates high thresholds and reduced logical failure rates for biased noise error models simulated under perfect stabilizer measurement conditions.

We also present a maximum-likelihood decoder for stabilizer codes, called the effective weight and degeneracy (EWD) decoder. The EWD decoder uses Metropolis-based Monte Carlo sampling to find the most likely equivalence class for a given error syndrome, whose implementation depends on the bias of the noise model and is independent of the physical error rate of the qubits. The EWD decoder is a near-optimal decoder that is efficient, fast and can be easily modified to characterize new topological stabilizer codes, such as the XYZ^2 code.

List of papers

This thesis consists of an introductory text and the following two appended papers:

Paper A

Basudha Srivastava, Anton Frisk Kockum, and Mats Granath. "The XYZ^2 hexagonal stabilizer code". Quantum **6**, 698 (2022).

Paper B

Karl Hammar, Alexei Orekhov, Patrik Wallin Hybelius, Anna Katarina Wisakanto, Basudha Srivastava, Anton Frisk Kockum, and Mats Granath. "Error-rate-agnostic decoding of topological stabilizer codes". Physical Review A **105**, 042616 (2022).

Papers outside the scope of this thesis:

Paper C

Ingemar Bengtsson and Basudha Srivastava. "Dimension towers of SICS: II. Some constructions". Journal of Physics A: Mathematical and Theoretical **55**, 215302 (2022).

ACKNOWLEDGEMENTS

I would like to give my sincere thanks to my supervisor, Mats Granath, and my co-supervisor, Anton Frisk Kockum, for their guidance and invaluable insights during the thesis. Their constant support and encouragement have been instrumental throughout my studies. I would like to thank Karl Hammar, Alexei Orekhov, Patrik Wallin Hybelius and Anna Katariina Wisakanto for being wonderful collaborators on the paper, especially Karl for his support in running the EWD decoder. I would also like to thank Ingemar Bengtsson for interesting discussions on quantum information. I also thank Timo Hillmann and Ben Criger for valuable discussions.

I would like to thank my family and my friends for providing me with all the motivation and inspiration. Lastly, I thank Pulkit Chauhan, for the endless love and support.

CONTENTS

PART I
BACKGROUND

1 Introduction

Communication is an indispensable part of human life. Modern communication takes place through technologies that facilitate transmission of data over a medium, and storage of data, which can be thought of as transmission over time instead of space. While we rely on the storage and transmission of information in almost every aspect of our lives, the data is repeatedly subjected to adverse effects of noise that gets inevitably added by interactions with the environment. Thankfully, decades of research has resulted in a robust technique of dealing with noise, through what is known as error correction.

Reliable transmission of information is achieved through the use of error-correcting codes. As an example, modern communication systems use a class of error-correcting codes known as Low Density Parity Check (LDPC) codes for various techniques, such as digital video broadcast (DVB S2), wireless local-area networks (802.11n), and wireless metropolitan-area networks (802.16e) [1].

While the growth of classical technologies has been tremendous, there are fundamental limits to what can be achieved with a classical computer. The realization that even simple quantum systems cannot be simulated efficiently on the most powerful classical computers, led to the emergence of quantum computing in the early 1980s [2]. The field of quantum computing deals with utilizing quantum systems for information processing, and as such offers to provide solutions to problems far beyond the reach of classical computing, at least in a reasonable amount of time. Some examples of algorithms that promise this advantage are in the fields of cryptography [3], quantum simulation [4], machine learning [5, 6], etc. [7, 8].

While classical computers represent information as 0 and 1 bits, quantum computing deals with two-level quantum systems known as *qubits* (quantum bits). Similar to classical computing, however, quantum computing is also susceptible to noise during its implementation. In fact, compared to the error rates of modern classical computers ($\sim 10^{-17}$) [9], current Noisy Intermediate-Scale Quantum (NISQ) de-

vices [10] are highly vulnerable to interactions with the environment. In order for quantum processors to circumvent this issue of noise, and to make use of the inherent capacity of quantum computing, we require quantum error correction.

Naively replicating the methods of classical error correction does not work for quantum computing due to the underlying principles of quantum mechanics – arbitrary quantum states are prohibited from being duplicated due to the *no-cloning theorem* [11, 12], measurements on qubits destroy the information held by them, and errors are continuous, correcting which would require infinite precision. Fortunately, Peter Shor demonstrated in 1995 that quantum error correction is possible by introducing the first quantum error-correcting code [13]. Andrew Steane independently introduced another quantum error-correcting code [14] in 1996. Shor's nine-qubit code made a generalization of the classical repetition code to quantum computing, while Steane's seven-qubit code was inspired by the popular classical Hamming code. In the following years, further research into stabilizer codes and topological structures gave rise to various promising new frontiers of quantum error correction, that aim to reduce the effects of noise on quantum systems and make quantum advantage realistically possible.

One of the most studied and promising candidates of topological stabilizer codes is the surface code, introduced by Kitaev in 1997 [15, 16, 17]. The surface code boasts high performance for quantum computers with the added benefits of local measurements and low connectivity between qubits. Over the years, there have been many variants of the surface code, tailored to particular noise models or specific desired properties [18, 19, 20, 21, 22].

As of today, the field of quantum error correction is not only theoretical in nature, evidenced by the many successful experimental realizations of quantum error-correcting codes [23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35].

In this thesis, we provide the background for the two appended papers, which introduce a novel topological stabilizer code on the

honeycomb lattice, and a new maximum-likelihood decoder for topological stabilizer codes. We start the discussion by looking at the basics of classical error correction in Chapter 2. This provides the background knowledge necessary to understand how various classical error-correction techniques were adapted to quantum codes, at the same time serving as an analogy when discussing the difficulties of the adaptation.

In Chapter 3, we give the fundamental concepts of quantum computing, sticking to concepts necessary for understanding the next chapter on quantum error correction. Chapter 4 starts with the simplest example of a quantum repetition code, inspired by its classical counterpart. Using this example, we illustrate the challenges faced by quantum error correction, and how they are overcome with the help of clever constructions. In the next section, we give the motivation for relying on stabilizer codes for quantum error correction, and describe some noise models necessary for understanding the appended papers. Afterwards, we introduce topological stabilizer codes, and explain its basic concepts with the help of two examples. Finally, in the last section, we explain the principles of decoding as is used in quantum error correction, and describe a few metrics used in the evaluation of quantum error-correcting codes.

Chapter 5 gives a short summary on the two appended papers, while Chapter 6 provides the conclusion of the thesis and an outlook for future research.

2 Classical error correction

Most digital data in the world is coded into sequences of bits, which can take the values 0 and 1. These sequences can then be stored to preserve information, or transmitted to send and receive messages over a communication channel. Physically, the bits can be implemented by using the low and high voltage levels of an electrical signal, two electronic states of a flip-flop circuit, or (historically) punched and unpunched hole positions on a punched card. Any of these techniques is susceptible to faults in its implementation, e.g., the signal carrying a bit 1 (high voltage level) might be subjected to environmental noise during storage or transmission and change to a 0 (low level).

An error in data transmission occurs when the intended message is modified by the channel in such a way that the receiver receives an altered message. A simplified error model for classical bits is the binary symmetric noise channel (see Fig. 2.1), which in the case of error converts a transmitted bit 0(1) to 1(0) with probability p, specified by the message channel.

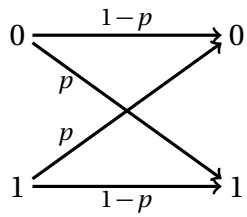


Figure 2.1: Binary symmetric noise channel

In order to prevent the effect of noise on classical data, we use error-correcting codes to protect the message by adding redundancy to the message bits. One of the simplest error-correcting codes that uses the principle of redundancy is the class of repetition codes.

2.1 Repetition codes

Repetition codes use the redundancy of sending duplicate information through the channel and decoding the incoming information using a majority rule. For example, instead of sending 0 or 1, we can encode the information by adding two redundant bits as

$$\bar{0} \longrightarrow 000 \tag{2.1}$$
$$\bar{1} \longrightarrow 111 \tag{2.2}$$

and send $\bar{0}$ or $\bar{1}$ instead. For a binary symmetric channel, a single bit-flip on the encoded bits will occur with probability $3p(1-p)^2$, and can easily be detected and corrected by using a majority-rule decoding. The three encoded bits in this example are called the *physical* bits, whereas $\bar{0}$ and $\bar{1}$ bits are the *logical* bits. 000 and 111 are called the *codewords* of the code.

Error correction fails, however, in the event of two or three simultaneous bit-flips. This occurs with a probability $3p^2(1-p)+p^3$, which is less than the physical error rate for error probabilities less than half. Sending a logical bit instead of a physical bit through the channel would then be beneficial only in the case of $p < 0.5$. The event of three bit flips takes the logical bit from $\bar{0}$ to $\bar{1}$ or vice versa, and is undetectable by the code.

The Hamming distance between the codewords, i.e., the minimum number of errors on the physical bits that translates to a logical error on the logical bits, is called the *distance* of the code. In the above example, the distance of the code is $d = 3$. The distance d of an error-correcting code is directly related to the maximum number of errors a code can detect, $d-1$, and can correct, $\lfloor (d-1)/2 \rfloor$. As such, increasing the distance of a code also improves its error-correcting properties. For repetition codes, the distance of a code is equal to the number of physical bits, and it is easy to see that for physical error rates less than 0.5, the logical error rate is reduced by increasing the number of encoded bits.

Repetition codes are a fairly simple example in the family of error-correcting codes called *linear block codes*.

2.2 Linear block codes

Linear block codes are defined as codes that encode blocks of messages into larger sequences and for which any linear combination of the codewords also forms a codeword. For any linear block code, the family of codes can be labelled by the parameters $[n,k,d]$, where n is the number of physical bits, k is the number of logical bits, and d is the distance of the code. For repetition codes, this family is given by $[n,1,n]$.

For a binary $[n,k,d]$ linear code, there exist k linearly independent codewords, $\mathbf{g}_0,\mathbf{g}_1,\ldots\mathbf{g}_{k-1}$, such that all possible codewords are linear combinations of these [1]. Given a length-k initial message to be encoded,

$$\mathbf{u} = (u_0, u_1, \ldots, u_{k-1}), \tag{2.3}$$

the length-n encoded message is given by

$$\mathbf{v} = u_0\mathbf{g}_0 + u_1\mathbf{g}_1 + \ldots u_{k-1}\mathbf{g}_{k-1}. \tag{2.4}$$

As $u_i \in \{0,1\}$, there are 2^k distinct codewords that can be generated. Arranging the n-bit long, k linearly independent codewords, $\mathbf{g}_0,\mathbf{g}_1,\ldots\mathbf{g}_{k-1}$, as the rows of a $k\times n$ matrix gives us the *generator matrix* $\mathbf{G}$ of the code.

A logical data sequence $\mathbf{u}$ which is encoded into a vector $\mathbf{v}$ can then be expressed using the generator matrix as:

$$\mathbf{v} = \mathbf{u}\cdot\mathbf{G} \tag{2.5}$$

Another important matrix associated with a linear block code is the $(n-k)\times n$ *parity-check matrix* $\mathbf{H}$, which is related to the codewords as:

$$\mathbf{v}\cdot\mathbf{H}^{\mathrm{T}} = \mathbf{0} \tag{2.6}$$

The encoded codewords **v** are transmitted through the message channel which introduces an error sequence **e** for the code. This error **e** can be expressed as a vector of length-n and modifies the codeword to become

$$\mathbf{v'} = \mathbf{v} + \mathbf{e} \tag{2.7}$$

Mapping the transmitted message back into the allowed codewords of the code to find the original message is called *decoding*.

Given **v'**, maximum-likelihood (ML) decoding finds the codeword **v** that maximizes the conditional probability $P(\mathbf{v'}|\mathbf{v})$. For the binary symmetric channel, ML decoding is equivalent to minimum-distance decoding, that is, finding the codeword **v** with the minimum Hamming distance to **v'**. While ML decoding is an optimal decoding technique, it becomes highly complex for large code sizes, which is the case for practical purposes. There are various sub-optimal decoding algorithms that can be used in this case, which give sufficiently good performance with reduced complexity.

Using the parity-check matrix, we can check the transmitted sequence **v'** for detectable errors by calculating:

$$\begin{aligned} \mathbf{s} &= \mathbf{v'} \cdot \mathbf{H}^{\mathrm{T}} \\ &= \mathbf{v} \cdot \mathbf{H}^{\mathrm{T}} + \mathbf{e} \cdot \mathbf{H}^{\mathrm{T}} \\ &= \mathbf{e} \cdot \mathbf{H}^{\mathrm{T}} \end{aligned} \tag{2.8}$$

If **e** is a valid codeword, **s** turns out to be a length-$(n-k)$ zero vector; otherwise, the errors are indicated as non-zero values in **s**, called the *syndrome* of the parity checks. With the help of the syndrome, we can find out the most-likely codeword that was transmitted to perform *syndrome decoding*.

As an example, for the $[3,1,3]$ repetition code illustrated in Section 2.1, the generator matrix is given by the 1×3 matrix

$$G = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \tag{2.9}$$

and the parity-check matrix is the 2×3 matrix

$$H = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \tag{2.10}$$

For binary codes, all operations are performed modulo 2:

$$\begin{array}{cccc} 0\pm 0=0 & 1\pm 1=0 & 0\pm 1=1 & 1\pm 0=1 \\ 0\cdot 0=0 & 1\cdot 1=1 & 0\cdot 1=0 & 1\cdot 0=0 \end{array} \tag{2.11}$$

Hence, the $2^k = 2$ codewords, $\bar{0}$ and $\bar{1}$ are encoded as

$$\bar{0}: \quad \mathbf{v} = 0\cdot\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \tag{2.12}$$

$$\bar{1}: \quad \mathbf{v} = 1\cdot\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \tag{2.13}$$

If an error corresponding to a bit-flip on the first bit occurs, $\mathbf{e} = [1\,0\,0]$, the transmitted message is modified to

$$\bar{0}: \quad \mathbf{v'} = \mathbf{v} + \mathbf{e} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{2.14}$$

$$\bar{1}: \quad \mathbf{v'} = \mathbf{v} + \mathbf{e} = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \tag{2.15}$$

Now, to detect errors on the transmitted sequence, we can calculate the syndrome of the code. In the case the original information was 0, the syndrome is given by

$$\mathbf{s} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{2.16}$$

which indicates that an error has occurred during transmission. For repetition codes, the syndrome checks are equivalent to checking the parity of consecutive pairs of bits, and so, it is easy to create a simple look-up table to decode the error. In this example, we can easily decode that a bit-flip error occurred on the first bit.

Decoding would have failed, however, if the error sequence added by the message channel was $\mathbf{e} = [1\,1\,1]$, as this changes the codewords from $\bar{0}$ to $\bar{1}$ or vice versa, and is undetectable by the syndrome. For the $[3,1,3]$ repetition code, the maximum number of bit-flip errors the code can detect are $d-1=2$ and the maximum number of errors it can correctly decode and correct are $(d-1)/2 = 1$.

There exist better linear codes than repetition codes that, for example, are tailored for different noise channels and message characteristics, increase the distance, optimize the code rate k/n, or reduce the logical failure rates during transmission. Nevertheless, for our purpose, simple repetition codes are enough to explain the link to quantum error correction. Before delving into that, however, we will look at the basics of quantum computing.

3 Quantum computing

Continuing the analogy with Classical Computing, where the fundamental unit of information is a bit, Quantum Computing uses quantum bits, or qubits, as their basis. As these basic units have widely different properties, the laws governing classical and quantum computing differ greatly, both for better and for worse.

The advantage that quantum computing possesses over its classical counterpart came into the picture in the early 1980s, when Paul Benioff [36], Richard Feynman [2] and Yuri Manin [37] put forward the idea of quantum computers (Benioff proposed a quantum mechanical model of the Turing machine; Feynman and Manin proposed quantum computing as a solution to the problems faced when simulating quantum systems). It was in 1994, however, when Peter Shor devised his ingenious factoring algorithm [3], that a quantitative and concrete advantage of quantum computing came into view. Since then, there has been a lot of research into algorithms that could improve upon the computations of a classical computer [9, 38, 39, 40, 41, 42]. It would be worthwhile to note, however, that these advantages are still mostly theoretical in nature. The restrictions that we face in implementing these algorithms can be understood by looking at the properties of qubits.

3.1 Quantum states

A qubit is a two-level system that follows the postulates of quantum mechanics. Physical realization of a qubit can be the spins of an electron, polarization of photons, trapped ions, etc. For the current work, we will only focus on the theoretical framework that defines a qubit.

Similar to the 0 and 1 states of a classical bit, a qubit has orthogonal computational states $|0\rangle$ and $|1\rangle$ (where the *ket* $|\cdot\rangle$ notation is used for quantum states), which can be represented as vectors in the com-

plex Hilbert space. Any arbitrary quantum state in the 2-dimensional Hilbert space can then be written as a "superposition" of the basis states as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{3.1}$$

$$\text{where } |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{3.2}$$

and $\alpha, \beta \in \mathbb{C}$. Measuring the state in the $|0\rangle / |1\rangle$ basis would destroy any superposition with the outcome being $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$.

As probabilities sum to one, $|\alpha|^2 + |\beta|^2 = 1$, and Eq. (3.1) can be re-written (up to irrelevant global phase factors) as

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\phi}\sin\frac{\theta}{2}\,|1\rangle\,. \tag{3.3}$$

As a consequence, a qubit state can be represented graphically as a point on a 3-dimensional sphere, called the *Bloch sphere*, with the point specified by the parameters θ and ϕ (see Fig. 3.1). The computational basis states $|0\rangle$ and $|1\rangle$ lie on the positive and negative z-axes, respectively.

Multiple isolated qubits can be written as tensor products of pure states (individual vectors) as

$$|\psi\rangle \otimes |\phi\rangle = |\psi\rangle\,|\phi\rangle \tag{3.4}$$

For example,

$$|0\rangle \otimes |0\rangle = |0\rangle\,|0\rangle = |00\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.5}$$

For two-qubit systems, the computational basis states are $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ and any two-qubit state can be written as a linear superposition of these states.

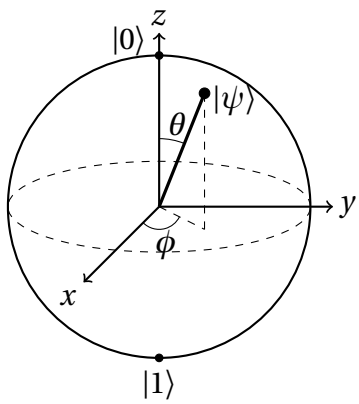


Figure 3.1: Bloch sphere representation of a qubit.

In addition to tensor product states, there also exist multiple qubit states that are *entangled*, and hence cannot be written as separable vectors under tensor product. One common example is the two-qubit Bell state.

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{3.6}$$

It is easy to see that such states cannot be written as a tensor product of multiple pure quantum states. Measurements on an entangled state are not as straightforward as in the case of pure states, as measuring an entangled qubit disentangles it from the rest and also projects the state of the other qubits based on the measurement outcome. For example, for the given Bell-state $|\Phi^+\rangle$, measuring the second qubit in the state $|0\rangle$ also projects the first qubit to $|0\rangle$.

Another formulation for describing quantum systems is through density matrices, ρ. In addition to describing pure states, density matrices can be used to represent *open systems*, i.e., quantum states which are not necessarily isolated and are allowed to interact with the environment. Density matrices can be written as

$$\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i|, \tag{3.7}$$

where the vectors $|\psi_i\rangle$ are pure states with corresponding probabilities p_i and the *bras* $\langle\psi_i|$ are the conjugate transposed vectors of their *ket* counterparts. These density matrices which are a mixture of numerous pure quantum states are also called *mixed states*. For pure states, the quantum system can exist in only a single state $|\psi\rangle$ with probability $p = 1$, and the density matrix of a pure state is

$$\rho = |\psi\rangle\langle\psi|. \tag{3.8}$$

Density matrices follow the following conditions,

$$\rho = \rho^\dagger \tag{3.9}$$

$$\rho \geq 0 \tag{3.10}$$

$$\mathrm{Tr}\rho = 1, \tag{3.11}$$

i.e., they are positive semi-definite matrices with trace 1. Here, $\rho \geq 0$ signifies that the matrix has non-negative eigenvalues, and $\mathrm{Tr}\rho = 1$ follows as probabilities sum to 1:

$$\mathrm{Tr}\rho = \sum_i p_i \mathrm{Tr}\left(|\psi_i\rangle\langle\psi_i|\right) = \sum_i p_i = 1. \tag{3.12}$$

3.2 Quantum operators

In order to create a working quantum computer, we need to be able to initialize and modify the information carried by qubits. Similar to initializing classical bits as 0 or 1 and acting on them with classical gates to perform operations, we can (in principle) initialize qubits into arbitrary quantum states and act on them with quantum operators. In quantum computing, any unitary matrix forms a valid quantum gate.

In analogy with the classical NOT gate that takes a 0 to 1 and vice versa, we have a quantum NOT gate, X, that takes state $|0\rangle$ to $|1\rangle$ and $|1\rangle$ to $|0\rangle$. In matrix representation, the X gate can be written as

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{3.13}$$

We can show the evolution of a quantum state graphically with the help of a quantum circuit diagram, with single "lines" (quantum wires) representing single qubits, and "blocks" representing quantum operators. The action of the quantum NOT gate can then be visualized as in Circuit 3.1.

$$\alpha|0\rangle + \beta|1\rangle \quad \text{---}\boxed{X}\text{---} \quad \alpha|1\rangle + \beta|0\rangle$$

Circuit 3.1: Pauli-X gate

While the classical NOT gate is the only non-trivial single-bit gate in classical computing, for a single qubit, valid quantum gates include the group of all 2-dimensional unitary matrices. For the sake of simplicity, however, we can first consider the most important gates – the Pauli matrices. These include the Pauli-X gate which we have seen above in Eq. (3.13), and the following Pauli-Y and Pauli-Z gates, represented graphically in Circuit 3.2.

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{3.14}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3.15}$$

Pauli-Z is also called the *phase-flip* gate, as it flips the phase of $|1\rangle$ to $-|1\rangle$ and leaves $|0\rangle$ unchanged. The three Pauli matrices, in addition to the Identity matrix I which acts trivially on all qubits, form a *nice error basis* [43], i.e., any unitary matrix can be expressed as a linear combination of these four. It is sufficient to look at the action of the Pauli matrices on the orthogonal basis states $|0\rangle$ and $|1\rangle$, as they span the complex vector space.

The Z, X and Y matrices all have eigenvalues $+1$ and -1, and have

$$\alpha|0\rangle+\beta|1\rangle \;\text{---}\boxed{Y}\text{---}\; i\alpha|1\rangle-i\beta|0\rangle$$

$$\alpha|0\rangle+\beta|1\rangle \;\text{---}\boxed{Z}\text{---}\; \alpha|0\rangle-\beta|1\rangle$$

Circuit 3.2: Pauli-Y and Pauli-Z gates

respective eigenvectors $|0\rangle\,\&\,|1\rangle$, $|+\rangle\,\&\,|-\rangle$ and $|i\rangle\,\&\,|-i\rangle$, where

$$|+\rangle=\frac{|0\rangle+|1\rangle}{\sqrt{2}} \qquad |-\rangle=\frac{|0\rangle-|1\rangle}{\sqrt{2}} \tag{3.16}$$

$$|i\rangle=\frac{|0\rangle+i|1\rangle}{\sqrt{2}} \qquad |-i\rangle=\frac{|0\rangle-i|1\rangle}{\sqrt{2}} \tag{3.17}$$

In the context of quantum error correction, it is also important to note that the Pauli matrices anticommute with each other, and are related to each other as,

$$\begin{aligned} ZX&=-XZ=iY\\ YZ&=-ZY=iX\\ XY&=-YX=iZ \end{aligned} \tag{3.18}$$

Here, the operators are acting on the same single qubit. Pauli matrices can also act on multiple qubits at the same time and their operation is defined with the help of a tensor product. For example, if Pauli-X acts on qubits 1 and 2, we can denote the operation by

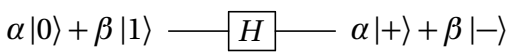
$$(X_1\otimes X_2)(|0\rangle_1\otimes|0\rangle_2)=X_1X_2|00\rangle=|11\rangle \tag{3.19}$$

In the sections pertaining to quantum error correction, we will often write such a tensor product simply as $X_1X_2=XX$. The distinction between Pauli operators acting on single or multiple qubits will be clear from their usage in the discussion.

Another important single-qubit gate is the Hadamard gate (see Circuit 3.3), which maps the eigenvectors of Pauli-Z to the corresponding

eigenvectors of Pauli-X:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{3.20}$$



Circuit 3.3: Hadamard gate

Unitary matrices can also act as quantum operators on multiple qubits, with the most common two-qubit gate being the CNOT (Controlled-X) gate. The action of the CNOT gate corresponds to performing a bit-flip (X gate) on the second qubit, conditioned on the first qubit being in state $|1\rangle$.

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{3.21}$$

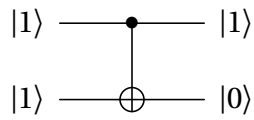
The action of the CNOT gate on an arbitrary two-qubit state is (where the first qubit is the control qubit and the second is the target qubit):

$$\begin{gathered} \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle \\ \Big\downarrow \text{CNOT} \\ \alpha|00\rangle + \beta|01\rangle + \gamma|11\rangle + \delta|10\rangle. \end{gathered} \tag{3.22}$$

Circuit 3.4 shows the circuit representation of the CNOT gate.

The CNOT gate is used to introduce entanglement between isolated qubits. Other two-qubit gates that operate on the same principle are Controlled-Z and Controlled-Y, which perform a Pauli gate on the target qubit conditioned on the state of the control qubit.

Similar to the single-qubit case, the group of all 4-dimensional unitary matrices make up the set of valid quantum gates acting on two qubits. However, in practice, we only need a subset of these gates in



Circuit 3.4: CNOT gate acting on two qubits with the control and target states set to $|1\rangle$

order to achieve universal quantum computation. Specifically, we only need an entangling gate, such as the CNOT, together with single-qubit gates, to create any arbitrary multiple-qubit quantum gate.

3.3 Quantum measurements

Measurements in quantum computing are carried out by interacting with the system to gain knowledge of its properties. While there are philosophical differences in the interpretation of measuring a quantum system, the common ground rests upon the fact that measurements give us some information about the state of the system. In general, measurements can be defined using POVMs (Positive-Operator Valued Measures), which are a set of elements E_m following

$$E_m = E_m^\dagger \tag{3.23}$$

$$E_m \geq 0 \tag{3.24}$$

$$\sum_m E_m = 1. \tag{3.25}$$

The last restriction is required in order for the measurement probabilities to sum to 1, as for a given density matrix ρ,

$$p_m = \mathrm{Tr}(E_m \rho), \tag{3.26}$$

where p_m is the probability of observing an outcome m. A special case of POVM, and the one most widely used in the context of quantum computing, is the Projection-Valued Measure (PVM). In this case,

the POVM elements are an orthogonal set of projectors, P_m's, such that $P_m P_{m'} = \delta_{mm'} P_m$. For a d-dimensional system, there are d such elements which can be defined by the set of eigenvectors $|e_m\rangle$ of an observable M as $P_m = |e_m\rangle \langle e_m|$.

Measuring a state in the PVM basis gives an outcome as the corresponding eigenvalue m with probability

$$p(m) = \langle \psi | P_m | \psi \rangle \tag{3.27}$$

As an example, consider the observable Z, with eigenvectors $|0\rangle$ and $|1\rangle$ and the corresponding eigenvalues $+1$ and -1, respectively. A Z measurement on an unknown quantum state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ will give the outcome $+1$ with probability

$$p_{+1} = \langle \psi | 0 \rangle \langle 0 | \psi \rangle = |\alpha|^2 \tag{3.28}$$

and outcome -1 with probability

$$p_{-1} = \langle \psi | 1 \rangle \langle 1 | \psi \rangle = |\beta|^2 \tag{3.29}$$

Projective measurements project the state of the qubit to one of its eigenvectors. Measuring an observable M on a qubit will destroy the superposition, and also disentangles it from the environment. Projective measurements perform an important role in the theory of quantum error correction, as we will see in the next Chapter.

4 Quantum error correction

Quantum computation requires devices that can generate, store, and perform calculations on quantum systems. Research in the direction of achieving this has looked at realizations of qubits using superconducting circuits [44, 45, 46], trapped-ion devices [47, 48], photons [49], etc [50, 51]. Different architectures use different techniques to implement the fundamental blocks of quantum computing – initialization of qubits, evolution of states through quantum gates, and measurement of quantum systems. All of these, however, are susceptible to noise by interaction with the environment. While there is an ongoing effort to reduce the effects of noise as much as possible, we need additional protection against noise in order to fully harness the potential advantages of a quantum computer. Quantum error correction is a technique that introduces redundancy in order to protect quantum information from the adverse effects of noise.

Looking back at the case of a simple three-bit repetition code used in classical computing, one might naively try to repeat the same procedure. However, the principles of quantum mechanics restrict us from using the same technique for quantum systems:

1. No-cloning theorem
   The basic idea behind a repetition code is to duplicate the data numerous times so as to send multiple bits containing the same information over the same channel. In quantum computing, however, the *no-cloning theorem* [11, 12] prohibits us from doing this. The no-cloning theorem states that it is impossible to create a single cloning device that can copy an arbitrary unknown quantum state.

2. Continuous quantum errors
   While it is possible to describe any error operation through a linear combination of the "*nice error basis*" operators, the set of errors that actually occur on qubits are continuous.

3. Measurement destroys superposition
   Decoding of a classical linear code happens after taking into account the "measured" received messages. For example, in the case of a repetition code, a simple decoding technique is to use the majority rule on the incoming messages. This is not possible in quantum computing as measuring a quantum state in a given basis destroys its superposition and hence the knowledge held by the qubit.

4.1 Quantum repetition code

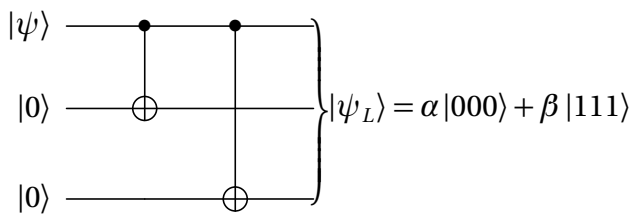
Although the situation looks dire, there is a known solution of overcoming the challenges posed by quantum computing. Given an arbitrary single-qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, it is prohibited by the no-cloning theorem to create a logical state as copies of $|\psi\rangle$:

$$|\psi_L\rangle \neq |\psi\rangle|\psi\rangle|\psi\rangle. \tag{4.1}$$

Nevertheless, we can instead create a state made up of three constituent qubits which creates a logical state

$$|\psi_L\rangle = \alpha|000\rangle + \beta|111\rangle, \tag{4.2}$$

with a simple encoding circuit, shown in Circuit 4.1.



Circuit 4.1: Encoding circuit for the three-qubit repetition code

Instead of simply repeating the state a few times, we can generate logical states of the computational basis states and encode $|\psi\rangle$ in these

three physical qubits.

$$|0_L\rangle = |000\rangle \tag{4.3}$$
$$|1_L\rangle = |111\rangle . \tag{4.4}$$

Now, for simplicity and analogy with the classical repetition code, suppose we have a single bit-flip (X) error on one of the physical qubits:

$$|\psi_L\rangle \xrightarrow{IIX} \alpha|001\rangle + \beta|110\rangle . \tag{4.5}$$

The error operator acting on the state is $I_1 \otimes I_2 \otimes X_3 = IIX$. This operator flips the state of the third qubit. Instead of measuring the qubits to decode with the majority rule as in the classical case, in this case we can simply look at the parity of the Z_1Z_2 and Z_2Z_3 observables on the state. In the case of no error, the parity measurements give eigenvalues of $(+1,+1)$ and for a single bit-flip error on the third qubit, the parity checks give the outcome $(+1,-1)$.

In fact, for a single bit-flip error and perfect parity check measurements, we can uniquely identify the error configuration applied to the initial state by the parity measurement outcomes, which are called the *syndrome* of the code, and apply an identical recovery operation on the same qubit (as $X^2 = I$):

$$\begin{array}{cc|c||c} +1 & +1 & \alpha|000\rangle + \beta|111\rangle & III \\ +1 & -1 & \alpha|001\rangle + \beta|110\rangle & IIX \\ -1 & +1 & \alpha|100\rangle + \beta|011\rangle & XII \\ -1 & -1 & \alpha|010\rangle + \beta|101\rangle & IXI \end{array} \tag{4.6}$$

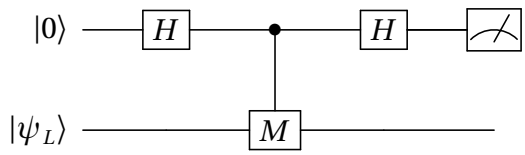
The above look-up table will work only when we are certain a single bit-flip error has occurred. In the event of two or more bit-flips, the correction given by the look-up table fails and the state $|\psi_L\rangle$ is transformed to another $(+1,+1)$ eigenstate of the observables ZZI and IZZ. We define the event of an error correction failure as a logical error:

$$\alpha|000\rangle + \beta|111\rangle \xrightarrow{IXX} \alpha|011\rangle + \beta|100\rangle \xrightarrow[\text{(Correction)}]{XII} \alpha|111\rangle + \beta|000\rangle . \tag{4.7}$$

Under the restriction of bit-flip errors, this code works completely like the classical repetition code, and is often called the quantum repetition code. It is important to note that this repetition code is unable to protect the state against phase-flip errors, which will go unnoticed by the parity checks measurements. For a repetition code that protects against only phase-flip errors instead, we can encode the basis states as $|0_L\rangle = |+++\rangle$ and $|1_L\rangle = |---\rangle$ and equivalently measure the parities of the observables XXI and IXX.

Measuring parity checks on the qubits as observables to gain information about the eigenvalues does not affect the superposition of the logical state. In fact, during the whole process we are unaware of the values of α and β and can detect and correct errors with only the knowledge of the syndrome.

A method of measuring the parity of the observables in the three-qubit repetition code without measuring the actual physical qubits is through the help of extra *ancilla* qubits which can be made to interact with the logical state to determine the parity measurements. Given a single-qubit operator M which is both Hermitian and unitary, with eigenvalues ± 1, we can construct a circuit which measures the observable M with the help of a Controlled-M gate and single-qubit Hadamard gates [9] (see Circuit 4.2).



Circuit 4.2: Circuit for measuring an observable M with eigenvalues ± 1 using an ancilla qubit (top).

The ancilla qubit is initialized in the $|0\rangle$ state and made to interact with the logical state $|\psi_L\rangle$. It is then measured after the end of the circuit to give the eigenvalue of the observable as the outcome. This circuit implements a projective measurement of observable M without destroying the superposition of the initial logical state.

After addressing the issues of the cloning and measurement problems, we are still left with the fact that errors in quantum computing are continuous. This issue is also addressed by the parity measurement scheme. Given a particular code with a logical state $|\psi_L\rangle$, we can assume that instead of a single bit-flip error on a qubit, a small rotation around the x-axis occurs. As the parity check operators are projective measurements, they project the state of the measured qubits into an eigenvector of the projector, giving the corresponding eigenvalue as the outcome. Then, measuring the observable after the application of a small rotation either projects the state into one with no error, or an X applied on the qubit, which, for example, can be corrected by a bit-flip repetition code.

This argument can be generalized to any single-qubit error matrix which is projected to either an Identity operation I, a bit-flip error X, a phase-flip error Z, or (up to global phase factors) a bit-phase-flip-error Y. Hence, instead of keeping track of the infinite continuous error possibilities, we can correct any arbitrary single-qubit error, if we can correct X, Y, and Z errors on the code.

4.2 Stabilizer formalism

Analogous to the linear codes of classical computing, stabilizer codes form a set of error-correcting codes for quantum computing. For the design of stabilizer codes, the Pauli operators take a very important role. The n-qubit Pauli group is defined as the tensor product of single-qubit Pauli operators acting on n qubits $\mathcal{P}_n = \langle I, X, Y, Z\rangle^{\otimes n}$.

Any unitary operator U that acts on a pure quantum state $|\psi\rangle$ without modifying it is called a *stabilizer* of the state.

$$U|\psi\rangle = |\psi\rangle \tag{4.8}$$

In other words, a unitary U stabilizes all states which are its eigenvectors with eigenvalue $+1$. For example, the stabilizer of $|0\rangle$ is Z, the stabilizer of $|+\rangle$ is X, $|00\rangle$ is stabilized by ZI and IZ, and $(|00\rangle + |11\rangle)/\sqrt{2}$ is stabilized by XX and ZZ.

Taking a subgroup S of $\mathcal{P}_n$ such that all its elements commute with each other, and ensuring that $-I \notin S$, we can define a set V_S of qubit states as being stabilized under S, if for all $|\psi\rangle \in V_S$:

$$g|\psi\rangle = |\psi\rangle \ \forall \ g \in S \tag{4.9}$$

These vectors form the codewords for the stabilizer code and span the logical subspace. The condition that the elements of the stabilizer group commute with each other and do not contain $-I$ is required to ensure a non-trivial vector subspace V_S.

In the stabilizer formalism, instead of describing quantum processes through the evolution of quantum states, we use stabilizers that specify the codespace, and operators that describe the evolution of the states stabilized under the stabilizer group. A stabilizer code which encodes k logical qubits into n physical qubits is uniquely identified by the generators of its stabilizer group $\langle g_1, g_2, \ldots, g_{n-k} \rangle$.

Such a code can be used for its error-correcting properties, under conditions specified by a theorem [9] which states that: A set of errors $\{E_j\} \in \mathcal{P}_n$ are correctable by a stabilizer code if

$$E_j^\dagger E_k \notin \mathcal{N}(S) \backslash S \ \forall \ j,k. \tag{4.10}$$

For a stabilizer group S, $\mathcal{N}(S)$ defines the normalizer of the group:

$$\mathcal{N}(S) = \{N \in \mathcal{P}_n | SNS^\dagger \in S\}. \tag{4.11}$$

In order for a code to detect these errors, the generators of the stabilizer code acting on the n physical qubits are measured with the help of ancilla qubits, as specified in Circuit 4.2. For any generator g_l which acts on a qubit i, an error $E_i \in \{X, Y, Z\}$ acting on qubit i will flip the eigenvalue of the measured stabilizer if g_l anticommutes with E_i. Hence, measuring all the generators $\langle g_1, g_2, \ldots, g_{n-k} \rangle$, give the outcome $+1$ or -1 on the ancilla qubits, depending on the error configuration on the physical qubits. In other words, if a stabilizer commutes with the error chain, its eigenvalue remains $+1$, otherwise it is flipped to -1. The latter stabilizers signify that errors exist on the code and are called *syndromes*.

The logical operators of the stabilizer code S, that is, the Pauli operators that can create a logical error in the code, are defined as $\mathcal{C}(S)\backslash S$, where $\mathcal{C}(S)$ is the centralizer of the stabilizer group:

$$\mathcal{C}(S) = \{c \in \mathcal{P}_n | c\,g = g\,c \ \forall\ g \in S\}. \tag{4.12}$$

In other words, the logical operators of a stabilizer code are the operators in $\mathcal{P}_n$ that commute with each stabilizer but are not in S, and as a result are not detectable by the stabilizer code.

The logical operators also follow the properties and commutation rules for single-qubit Pauli operators:

$$X_L^2 = Z_L^2 = I \tag{4.13}$$

$$X_L Z_L = -Z_L X_L. \tag{4.14}$$

For quantum error-correcting codes, we can define the distance of the code as the minimum number of single-qubit Pauli errors that can create a logical error. In the stabilizer formalism, the distance translates to the minimum weight of the logical operators. A family of quantum error-correcting codes can be labelled as an $[[n,k,d]]$ code, which encodes k logical qubits into n physical qubits and has a minimum distance of d. Here, the double brackets signify that the code is quantum.

Similar to the classical coding formulation, a quantum code of distance d can detect $d-1$ and can correct $\lfloor (d-1)/2 \rfloor$ single-qubit Pauli errors.

Taking the three-qubit repetition code as an example again, recall that the computational basis states $|0_L\rangle$ and $|1_L\rangle$ were encoded into logical states $|000\rangle$ and $|111\rangle$ respectively. As we saw earlier, these two states are common eigenvectors of the observables ZZI and IZZ with eigenvalues $+1$.

Hence, for the three-qubit repetition code, the stabilizers of the logical states are $S = \{III, ZZI, IZZ, ZIZ\}$, where we can choose the set of generators to be $\langle ZZI, IZZ\rangle$.

The logical operators for this code are (XXX) and (ZII). As the operator (XXX) takes $|0_L\rangle$ to $|1_L\rangle$, we can refer to it as the logical X,

and similarly (ZII) as the logical Z operator:

$$X_L = XXX \tag{4.15}$$

$$Z_L = ZII. \tag{4.16}$$

The minimum weight of logical operators is 1 ($Z_L = ZII$), and so the code has a minimum distance of 1 and can be labelled as a $[[3,1,1]]$ code. It has an error-detecting capacity of $d-1=0$, which is evident as it cannot detect any phase-flip errors.

However, the code can detect and correct bit-flip errors. In fact, the set of X errors which anticommute with at least one of the generators $\langle ZZI, IZZ\rangle$ is

$$\{XII, IXI, IIX, XXI, IXX, XIX\}. \tag{4.17}$$

Considering the case of only bit-flip errors, which we can call pure X noise, the only logical operator allowed is $X_L = XXX$, and the distance of the code for pure X noise becomes $d_X = 3$. Consequently, the three-bit repetition code can detect $d_X - 1 = 2$ X errors, and can correct $(d_X - 1)/2 = 1$ X error.

4.3 Noise models

Errors in quantum computing are any unwanted operations that result from the interaction between the quantum system and the environment. While the experimental effects of noise on the qubits depend on the physical realization of the quantum system among other things, in theory, a simplified error model is sufficient for the analysis of quantum error-correcting codes. In this work, we will assume that errors occur on qubits independently as single-qubit Pauli operations and are described by the process

$$\rho \longrightarrow \mathcal{E}(\rho) = (1-p)\rho + p_X X\rho X + p_Z Z\rho Z + p_Y Y\rho Y \tag{4.18}$$

where p_X, p_Y, and p_Z are respectively the probabilities of an X, Y, or Z error occurring on the qubit, and $(1-p) = (1-p_X-p_Y-p_Z)$ is the probability that no errors occur.

A general error model widely used for simulating quantum error-correcting codes is depolarizing noise. In this case, X, Y, and Z errors are equally likely to occur, i.e., $p_X = p_Y = p_Z = p/3$.

For this work, we are also interested in looking at biased noise, that is, when a single Pauli error is more likely than the other two. Biased noise error models are of relevance for several physical qubit systems [52, 53, 54]. Considering only X errors, such that $p_X = p$ and $p_Z = p_Y = 0$ creates the bit-flip channel, also known as pure X noise. Similar models exist for pure Z noise (phase-flip noise) and for pure Y noise. These models are important when considering the error-correcting properties of many stabilizer codes, as we will see in Section 4.4.2.

A possible choice for parameterizing noise bias is through η, which is used commonly in many works exploring biased noise [21, 22]. For phase-biased noise, as an example, η is defined as

$$\eta = \frac{p_Z}{p_X + p_Y} \qquad (p_X = p_Y). \tag{4.19}$$

Taking $\eta = 10$ would signify that a Z error is 10 times more likely to occur than an X or Y error. For $\eta = 0.5$, $p_Z = p_X = p_Y$ and the channel becomes depolarizing noise. Similarly, to create a pure Z noise model, η can simply be set to ∞.

4.4 Topological stabilizer codes

Topological codes are usually implemented on a 2D lattice of physical qubits, which encode quantum information in global degrees of freedom, but with geometrically local parity check operators [16, 55, 56]. Stabilizer codes that incorporate a topological structure in their parity checks possess the desirable quality of having low-weight and local stabilizer measurements. Local stabilizer measurements are needed to keep the physical design of the chip simple, as higher connectivity leads to complicated fabrication design. In addition, low-weight checks prevent the propagation of local errors to cascade throughout

the grid of physical qubits with subsequent parity check operations. Topologically inspired stabilizer codes can also assist in characterizing the properties of stabilizer codes by looking at the corresponding statistical physics model and finding out the transition characteristics, although we will not explore this property of topological stabilizer codes in this work.

The archetypical topological code is the toric code [16, 17, 57], a stabilizer code that is formed using two classical error correcting codes. It is an $[[O(d^2), 2, d]]$ code with periodic boundary conditions. It demonstrates desirable error-correcting properties with a distance that scales with the number of physical qubits, and has been one of the most studied codes in quantum error correction research. Here, we will focus on a variant of the toric code, called the rotated surface code.

4.4.1 The rotated surface code

The rotated surface code [18, 19, 21] with boundaries is a stabilizer code placed on a square grid of physical qubits, with code parameters $[[d^2, 1, d]]$. It consists of alternating weight-four $XXXX$ and $ZZZZ$ stabilizers on the square faces of the grid, called *plaquettes*. On the boundaries, it contains weight-two half-stabilizers XX and ZZ (see Fig. 4.1).

$X(Z)$ errors on the rotated surface code are detected through the $Z(X)$ stabilizers; Y errors are detected by both X and Z stabilizers. The code has a minimum distance d which translates to the linear dimension of the code. Hence, while the rotated surface code will protect a single logical qubit, the code distance increases by increasing the size of the code block.

The smallest logical operators on the rotated surface code are horizontal and vertical chains of Pauli-Z and Pauli-X operators respectively, as shown in Fig. 4.2.

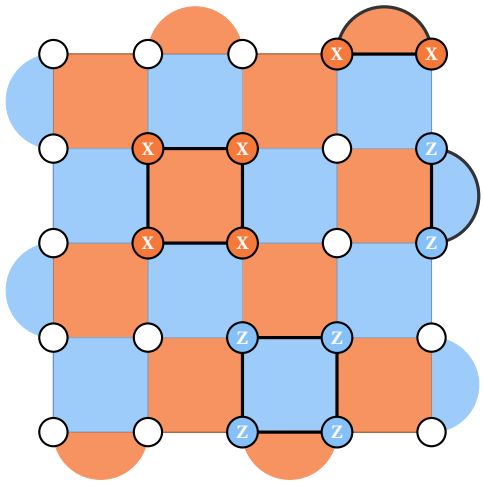


Figure 4.1: Rotated surface code for distance $d = 5$. Each vertex of the square grid represents a physical qubit. Orange and blue plaquettes denote the weight-four $XXXX$ and $ZZZZ$ stabilizers, respectively. Weight-two half-stabilizers XX and ZZ are shown on the boundaries.

4.4.2 The XZZX surface code

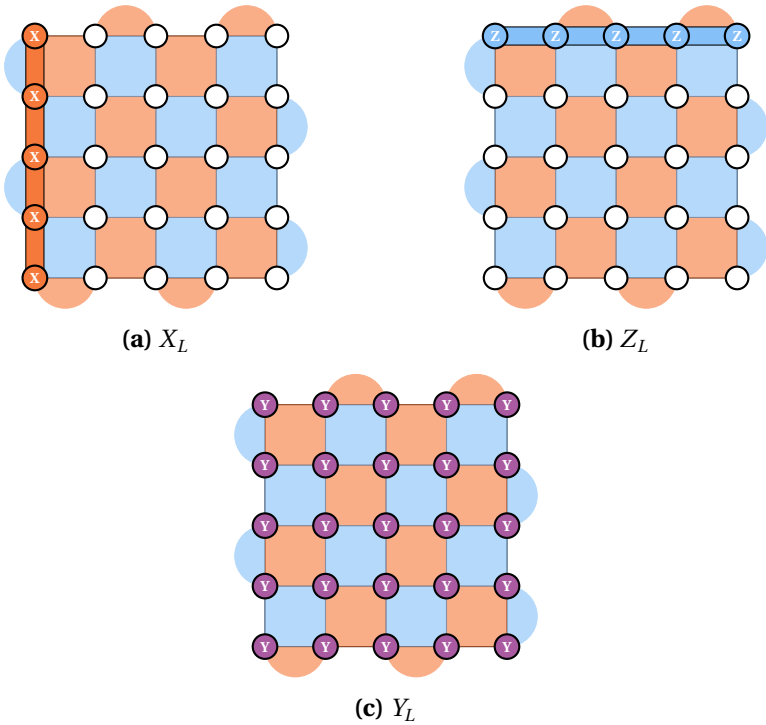
The XZZX surface code [22] is a variant of the rotated surface code defined similarly on a square lattice with code parameters $[[d^2, 1, d]]$. This code can be obtained from the rotated surface code, by applying a Hadamard transformation on every other qubit. The operation of a Hadamard gate on X transforms it into a Z, and similarly transforms a Z to an X:

$$HZH^\dagger = X \tag{4.20}$$

$$HXH^\dagger = Z. \tag{4.21}$$

So, the alternating $XXXX$ and $ZZZZ$ plaquette stabilizers of the rotated surface code get transformed to uniform $XZZX$ plaquette stabilizers on each square (see Fig. 4.3). This simple rotation leads to many desirable properties in comparison to the rotated surface code.

The logical operators on the XZZX code get transformed into vertical and horizontal chains of mixed XZ operators. While the minimum



(a) X_L **(b)** Z_L

(c) Y_L

Figure 4.2: Logical operators on the rotated surface code. (a) Weight-d X_L formed by a vertical chain of pure X's. (b) Weight-d Z_L formed by a horizontal chain of pure Z's. (c) Weight-d^2 pure Y_L formed by applying a Y on all qubits.

distance of the code remains d, $d_X = d_Z = d$, and $d_Y = d^2$, similar to the rotated surface code, this transformation reduces the number of smallest logical operators for pure noise. The only pure chain of X_L and Z_L on the code are the operators lying on the two diagonals of the grid (see Fig. 4.4). As a consequence, for any error chain E under pure X noise, there is only a single corresponding chain E' with the same syndrome pattern, where E' is given by $X_L E$ and X_L is the pure X logical operator. A similar behaviour is shown by error chains under pure Z noise.

Due to the mixed nature of the stabilizers, and the fact that the

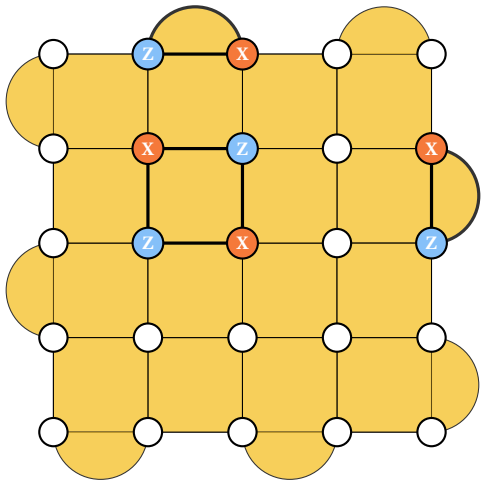
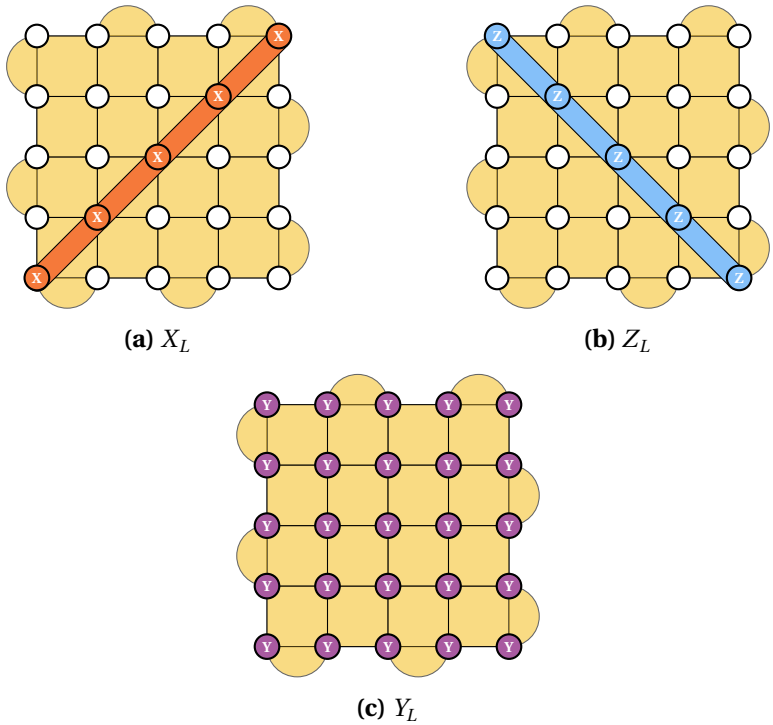


Figure 4.3: XZZX surface code for distance $d = 5$. Each vertex of the square grid represents a physical qubit. Yellow plaquettes denote the weight-four $XZZX$ stabilizers. Weight-two half-stabilizers XZ and ZX are shown on the boundaries.

stabilizers are uniform on all plaquettes, the XZZX code has an interesting syndrome property. For isolated X or Z errors on any physical qubit, the syndromes appear only in a unique diagonal direction, depending on whether the error is an X or a Z. This creates the potential to use the uni-directional syndrome information while decoding the error, as done in [22]. However, Y errors do not follow this property, and an isolated Y error on a qubit generates syndromes on all four neighbouring plaquettes.



(a) X_L **(b)** Z_L

(c) Y_L

Figure 4.4: Logical operators on the XZZX surface code. (a) Weight-d X_L formed by a diagonal chain of pure X's. (b) Weight-d Z_L formed by a diagonal chain of pure Z's. (c) Weight-d^2 pure Y_L formed by applying a Y on all qubits.

4.5 Decoding and characterization of quantum error-correcting codes

4.5.1 Decoding

So far, we have looked at how stabilizer codes can be used for encoding logical information in a number of physical qubits. We have also mentioned some popular topological stabilizer codes which have desirable properties. A complete error-correction procedure, however,

requires the detection of errors on the code, and a successful recovery applied to take the system back to its logical codespace.

Consider an $[[n,k,d]]$ stabilizer code with stabilizer group S and generators $\langle g_1, g_2, \ldots, g_{n-k}\rangle$. An error chain $E \in \mathcal{P}_n$ that anti-commutes with at least one stabilizer takes the code out of its logical subspace. In order to correct the error, we need to find a recovery operation $R \in \mathcal{P}_n$ such that $RE \in S$. Note that the recovery is not a unique operation for stabilizer codes; instead, any chain that is equivalent to E up to some stabilizers results in successful error correction.

A failure in error correction occurs if the recovery operation takes the code to a logical subspace different from the initial. That is, if $RE \in LS$, where L is the set of possible logical operators $\{X_L, Z_L, Y_L\}$. After applying the recovery, no detectable errors would remain on the code, as each element in L commutes with S. However, the logical information incurs a logical error.

As we cannot measure the final states of the physical or logical qubits without losing the quantum information contained in the system, we rely on syndrome measurements to correct errors on stabilizer codes. Syndrome decoding using a look-up table is an optimal but computationally complex method as it requires checking $2^{(n-k)}$ entries for an $[[n,k,d]]$ code, which becomes impractical for large code sizes.

Another optimal decoding technique is called *maximum-likelihood* (ML) decoding. Similar to the the classical case, the ML decoder maximizes the probability of an error chain given a measured syndrome. Given a particular syndrome configuration, we can divide the set of error chains possible for the syndrome into *equivalence classes* based on their action on the logical subspace. For instance, given an error E which leads to the syndrome, the set of all possible error chains can be divided into cosets as

$$\{EIS\} \cup \{EX_LS\} \cup \{EZ_LS\} \cup \{EY_LS\}. \tag{4.22}$$

Due to the degeneracy of solutions for quantum error-correcting codes, it is sufficient to find the equivalence class of the error chain with the maximum probability. Then, any error configuration in this equivalence class gives the optimal recovery operator.

While evaluating the complete set of possible error solutions has an exponential complexity in the number of physical qubits, there are approximate ML methods that make use of clever sampling techniques to find out the distribution of possible error chains. Although this reduces the computational complexity of implementing the decoder, dynamic error correction schemes would require even faster decoding techniques in practice.

Another popular and widely used decoder is the minimum-weight perfect matching (MWPM) decoder [57, 58, 59]. The MWPM is a graph-theory algorithm introduced by Jack Edmonds in 1965 [60, 61]. Given a graph G with a set of vertices V and edges E, a *matching* refers to pairing the vertices by their edges, such that no two pairs have any vertices in common. A perfect matching would correspond to having paired all vertices in V. Now, considering that the edges in E have a weight assigned to them through some specific criteria, the minimum-weight problem is to find a perfect matching of the all vertices in V, that minimize the total weight of the chosen edges.

Decoding a stabilizer code using the syndrome information can be mapped to an MWPM problem, and solved efficiently in polynomial time [62, 63]. In the given context, the syndromes of a code can be thought of as vertices on a graph, connected by edges weighted by the distance between these vertices, or a modified procedure that takes into account the bias levels of the noise. Assuming X and Z errors to act independently on the qubits, decoding is achieved by perfectly matching the two graphs separately, and applying the correction operators along the edges chosen in the matching that result in minimum-weight (as $Y = XZ$).

MWPM used for topological stabilizer codes is a sub-optimal decoder. Nevertheless, it is an efficient and fast decoding method that performs very close to optimal decoders for surface codes. Other examples of decoders used for stabilizer codes are [64, 65, 66, 67, 68].

4.5.2 Characterization

Quantum error-correcting codes are labelled by the parameters $[[n, k, d]]$, which characterize the code family to an extent. As mentioned earlier, the distance d of the code quantitatively determines the number of single-qubit errors that can be be corrected without the possibility of a logical error. Hence, for a given number of physical qubits n, the code with a higher d should perform better. However, there are other factors that should be taken into consideration before comparing the performance of stabilizer codes.

An important metric in the evaluation of quantum error-correcting codes is the *error threshold*. The *threshold theorem* [15, 69, 70] states that if the physical error rate of a code is below a certain threshold value, p_{th}, the logical failure rate can be made arbitrarily small by increasing the code size. Here, the logical failure rate refers to the probability of a logical error occurring on a code, given that the qubits are subjected to a particular noise model. Clearly, a code with a high error threshold is advantageous as scaling up the number of qubits will give better performance. The threshold error rate depends upon the code under consideration, the inherent noise model, and the decoder being used for the error-correction scheme.

For instance, considering depolarizing noise on the physical qubits, and simulating under the assumption of perfect and noiseless stabilizer measurements, the rotated surface code gives an error threshold of about 18.8(2)% [21]. A more realistic estimation of the actual threshold can be carried out by allowing noisy encoding and measurement circuits, also called circuit-level noise. While the threshold decreases down to about 1% [71] when including the effect of circuit-level noise, it is still high enough to make the surface code a popular and efficient candidate for quantum error correction. For perfect stabilizer measurements, The XZZX code also exhibits similar behaviour for depolarizing noise, with a threshold of 18.7(1)% [22]. Looking at pure noise for the XZZX code, taking pure X noise as an example, the threshold observed is 50%, a consequence of the lack of degeneracy in the logical operators for pure X noise. In contrast, the rotated surface

code shows a threshold of 10.8(1)% for pure X noise, which is far worse than that for depolarizing noise.

The threshold error rate alone is not sufficient to evaluate the performance of a code. For practical purposes, it may be beneficial to compare codes by the logical failure rate they exhibit at a given physical error rate. Taking the XZZX code as an example again, pure X noise, pure Z noise, and pure Y noise all show a threshold of 50%. Nevertheless, the logical failure rate is highly suppressed for pure Y noise owing to the quadratic distance $d_Y = d^2$, compared to the other two noise channels, $d_X = d_Z = d$.

Finally, the logical failure rates and the threshold error rates depend on the decoder being implemented. For the rotated surface code, the optimal maximum-likelihood decoder gives an error threshold of 18.8(2)% for depolarizing noise; however, using a sub-optimal decoder, such as the MWPM decoder, demonstrates a threshold of about 14.88% [59].

Part II
Current and future work

5 Summary of research papers

5.1 Research paper A

In paper A [72], we introduce a new quantum error-correcting code, the XYZ^2 hexagonal stabilizer code, which is based on the class of "matching codes" introduced by Wootton [73]. By introducing boundaries to a particular example of a matching code, we show that a single logical qubit can be encoded in a hexagonal grid of physical qubits, to create a quantum error-correcting code with parameters $[[2d^2, 1, d]]$.

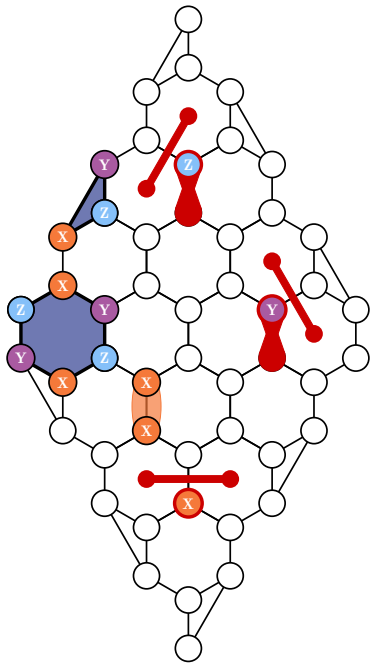


Figure 5.1: The XYZ^2 code for distance $d = 5$. The stabilizers of the code are weight-six $XYZXYZ$ on each hexagonal plaquette, weight-two XX on each vertical link, and weight-three XYZ on the boundary. Also shown are the unidirectional syndromes (red) for isolated X, Y, and Z errors.

The XYZ^2 hexagonal stabilizer code is a topological code with mixed weight-six $XYZXYZ$ plaquette stabilizers on each hexagonal face, and weight-two XX stabilizers on each vertical pair (or link) of qubits (see Figure 5.1). Following the reasoning of Sec. 4.4.2, the mixed stabilizers of the XYZ^2 code give rise to uni-directional syndromes for isolated single-qubit Pauli errors; the XYZ^2 code extends the directional syndrome property shown by Pauli-X and Pauli-Z in the XZZX code, to include Pauli-Y errors as well.

We give a theoretical basis of the novel stabilizer code, and show that the XYZ^2 code exhibits a quadratic distance ($2d^2$) for both pure Z and pure Y noise, given a grid consisting of $2d^2$ physical qubits. We also show how the rotated surface code can be transformed into the XYZ^2 code by applying rotations to create a YZZY surface code and replacing each physical qubit of the resulting code by an effective qubit, i.e., a logical qubit made up of two physical qubits stabilized under an XX stabilizer.

Finally, we compare the properties of the XYZ^2 code with the XZZX code, focusing mainly on the error thresholds and the logical failure rates for different noise models. For pure X, pure Y, and pure Z noise, the threshold and exact logical failure rate can be calculated analytically. The XYZ^2 code exhibits a threshold of 50% for all three pure noise models, equivalent to that of the XZZX code. However, the logical failure rate is much higher for pure X noise, a disadvantage for the XYZ^2 code, while the rates are highly suppressed for pure Z noise. This advantage can be controlled by changing the link stabilizers to get another variation of the XYZ^2 code. For example, if we measure ZZ stabilizers on the links instead (with an appropriate rotation of the plaquette stabilizers), the logical failure rate will be highly suppressed for pure X noise.

For depolarizing noise and biased noise, we simulate the performance of the code using maximum-likelihood decoding under the assumption of perfect stabilizer measurements. We use the EWD decoder of paper B for the calculations on the XYZ^2 code, and the MPS [74, 75] decoder for the XZZX code.

For depolarizing noise, we observe a threshold of about 18% and see that the logical failure rates depend largely on the code distance below the threshold. For biased noise parameterized by η, and taking $\eta = 10$, we observe error thresholds around 28% for all three biases, matching those shown by the XZZX code. Similar to the observations of the pure noise model, the XYZ^2 code performs worse for X-biased noise, but better for Z-biased noise when comparing by the number of physical qubits. For Y-biased noise, the logical failure rates depend on the number of physical qubits for both the XYZ^2 code and the XZZX code. As stated above, the XYZ^2 code can be tailored to the inherent noise model. For example, for a system with X-biased noise, we would measure the link stabilizers as ZZ, to get a better performance under X-biased noise.

For each of the noise models, we observe that the thresholds of the XYZ^2 code matches those of the XZZX code. The advantage of the XYZ^2 code reveals itself in the logical failure rates, which are highly suppressed for Z-biased noise on the XYZ^2 code, compared to the XZZX code, when comparing a similar number of data qubits. Another benefit that posits itself arises from the hexagonal nature of the plaquettes, suggesting that the XYZ^2 code might be more suitable to certain qubit architectures.

5.2 Research paper B

In paper B [76], we introduce a new approximate ML decoder for topological stabilizer codes, called the "effective weight and degeneracy" (EWD) decoder. This decoder is based on counting the most likely error chains in each equivalence class to find the most likely equivalence class corresponding to an error syndrome.

Similar to the ML Markov-chain Monte Carlo (MCMC) decoder [77, 78], the EWD decoder is based on the Metropolis algorithm for sampling error chains in each equivalence class using the degeneracy in possible error chains caused by the stabilizers of the code. However, in contrast to the MCMC decoder, the EWD decoder is more efficient

as it is "error-rate agnostic", i.e., it works independently of the physical error rate, and only requires information about the bias of the noise model. Hence, it can sample chains at a higher error rate than the physical error rate. In this sense, the EWD decoder is an intermediate approach between ML decoders and the MWPM decoder, as the latter does not account for error probabilities in its implementation.

We apply the EWD decoder to the rotated surface code and the XZZX code, for different noise models and compare with the MCMC decoder, the MPS decoder [74], and exact analytical expressions. We observe that the performance of the EWD decoder matches that of maximum-likelihood decoders for moderate code sizes or low error rates. In addition to this, the EWD decoder is a flexible decoder that can be used to evaluate the performance of novel stabilizer codes, as is evidenced by its usage in paper A.

6 Conclusions and Outlook

In this work, we have given the background for studying quantum error-correcting codes. Quantum error correction is necessary to protect quantum systems against noise and to harness the computational power of current NISQ devices. We focus mainly on topological stabilizer codes, as they possess desirable qualities of local and low-weight stabilizer measurements.

We have introduced a novel $[[2d^2, 1, d]]$ topological stabilizer code, the XYZ^2 hexagonal code. This code is inspired by the Kitaev honeycomb model and is implemented on a hexagonal grid containing $2d^2$ physical qubits, where the distance of the code is d. For pure Z and pure Y noise, we demonstrate that the code has a quadratic distance of $2d^2$, which leads to significantly reduced logical failure rates. The XYZ^2 code demonstrates high threshold rates that match those of the XZZX code, for depolarizing as well as biased noise.

We also present the EWD decoder, an approximate ML decoder, whose implementation depends on the bias of the error model and is independent of the actual physical error rate. It is an efficient and flexible decoder which matches the performance of ML decoders for moderate code sizes or low error rates.

The threshold values for the XYZ^2 code were obtained by simulating the performance of the code using the EWD decoder, under the assumption of perfect stabilizer measurement conditions. In order for the benefits of the XYZ^2 code to be of practical value, it is necessary to consider circuit-level noise as well to get a better picture of the actual threshold rates.

Compared to the weight-four plaquette stabilizers of the rotated surface code and the XZZX code, the XYZ^2 code has weight-six plaquette stabilizers, which in general might perform worse under circuit-level noise [79]. However, the XYZ^2 code also has weight-two link stabilizers, which make up about half of the stabilizers on the code. The effect of these stabilizers on circuit-level noise is also something to be explored in the future. This would require the development of a decoder that

is reliable and efficient while handling circuit-level noise, which can also take into account the fact that the link stabilizers do not occur in pairs, and hence cannot be matched using decoders such as MWPM.

Additionally, while we illustrate the tri-directional property of syndromes under X, Y, and Z noise, an efficient decoder needs to be implemented that can exploit this property. The XYZ^2 code can be represented as a concatenation of numerous two-qubit error detection codes and the YZZY surface code. A way of handling the decoding problem could be to use this fact to optimally decode the concatenated code block using a message-passing algorithm [80].

A future approach could be to modify the EWD decoder to include weights corresponding to faulty stabilizer measurements as well to evaluate circuit-level noise. Another direction for future work could be to combine the EWD decoder with supervised deep-learning methods to create a fast decoder which is flexible to the overall error rate, or to provide an accurate reward function for deep reinforcement learning approaches [67, 68].

The XYZ^2 code has already shown a promising performance, on par with other prominent stabilizer codes in recent research that aim to deal with biased noise. The current work opens the potential for a deeper analysis of the performance of the code, and creates a path for the development of more efficient, flexible, and reliable decoders.

References

[1] William Ryan and Shu Lin. “Channel codes”. Cambridge University Press. Cambridge, England (2009).

[2] Richard P Feynman. “Simulating Physics with Computers”. International Journal of Theoretical Physics **21** (1982).

[3] Peter W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In Proceedings 35th Annual Symposium on Foundations of Computer Science. Pages 124–134. (1994).

[4] I.M. Georgescu, S. Ashhab, and Franco Nori. “Quantum simulation”. Reviews of Modern Physics **86**, 153–185 (2014).

[5] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. “Quantum machine learning”. Nature **549**, 195–202 (2017).

[6] Sankar Das Sarma, Dong-Ling Deng, and Lu-Ming Duan. “Machine learning meets quantum physics”. Physics Today **72**, 48–54 (2019).

[7] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven. “Characterizing quantum supremacy in near-term devices”. Nature Physics **14**, 595–600 (2018).

[8] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, Peng Hu, Xiao-Yan Yang, Wei-Jun Zhang, Hao Li, Yuxuan Li, Xiao Jiang, Lin Gan, Guangwen Yang, Lixing You, Zhen Wang, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. “Quantum computational advantage using photons”. Science **370**, 1460–1463 (2020).

[9] Michael A. Nielsen and Isaac L. Chuang. “Quantum Computation and Quantum Information”. Cambridge University Press. Cambridge (2010).

[10] John Preskill. “Quantum computing in the NISQ era and beyond”. Quantum **2**, 79 (2018).

[11] D. Dieks. “Communication by EPR devices”. Physics Letters A **92**, 271–272 (1982).

[12] W. K. Wootters and W. H. Zurek. “A single quantum cannot be cloned”. Nature **299**, 802–803 (1982).

[13] Peter W. Shor. “Scheme for reducing decoherence in quantum computer memory”. Physical Review A **52**, R2493–R2496 (1995).

[14] A. M. Steane. “Error correcting codes in quantum theory”. Physical Review Letters **77**, 793–797 (1996).

[15] A. Yu. Kitaev. “Quantum error correction with imperfect gates”. In Quantum Communication, Computing, and Measurement. Pages 181–188. Springer US (1997).

[16] A. Yu. Kitaev. “Fault-tolerant quantum computation by anyons”. Annals of Physics **303**, 2–30 (2003).

[17] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. “Surface codes: Towards practical large-scale quantum computation”. Physical Review A **86**, 032324 (2012).

[18] H. Bombin and M. A. Martin-Delgado. “Optimal resources for topological two-dimensional stabilizer codes: Comparative study”. Physical Review A **76**, 012305 (2007).

[19] Yu Tomita and Krysta M. Svore. “Low-distance surface codes under realistic quantum noise”. Physical Review A **90**, 062320 (2014).

[20] Theodore J. Yoder and Isaac H. Kim. "The surface code with a twist". Quantum **1**, 2 (2017).

[21] David K. Tuckett, Andrew S. Darmawan, Christopher T. Chubb, Sergey Bravyi, Stephen D. Bartlett, and Steven T. Flammia. "Tailoring surface codes for highly biased noise". Physical Review X **9**, 041031 (2019).

[22] J. Pablo Bonilla Ataides, David K. Tuckett, Stephen D. Bartlett, Steven T. Flammia, and Benjamin J. Brown. "The XZZX surface code". Nature Communications **12**, 2172 (2021).

[23] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Yu Chen, Z. Chen, B. Chiaro, A. Dunsworth, I.-C. Hoi, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, A. Vainsencher, J. Wenner, A. N. Cleland, and John M. Martinis. "State preservation by repetitive error detection in a superconducting quantum circuit". Nature **519**, 66–69 (2015).

[24] Maika Takita, Andrew W. Cross, A. D. Córcoles, Jerry M. Chow, and Jay M. Gambetta. "Experimental Demonstration of Fault-Tolerant State Preparation with Superconducting Qubits". Physical Review Letters **119**, 180501 (2017).

[25] Christian Kraglund Andersen, Ants Remm, Stefania Lazar, Sebastian Krinner, Nathan Lacroix, Graham J. Norris, Mihai Gabureac, Christopher Eichler, and Andreas Wallraff. "Repeated quantum error detection in a surface code". Nature Physics **16**, 875–880 (2020).

[26] K. J. Satzinger, Y.-J Liu, A. Smith, C. Knapp, M. Newman, C. Jones, Z. Chen, C. Quintana, X. Mi, A. Dunsworth, C. Gidney, I. Aleiner, F. Arute, K. Arya, J. Atalaya, R. Babbush, J. C. Bardin, R. Barends, J. Basso, A. Bengtsson, A. Bilmes, M. Broughton, B. B. Buckley, D. A. Buell, B. Burkett, N. Bushnell, B. Chiaro, R. Collins, W. Courtney, S. Demura, A. R. Derk, D. Eppens, C. Erickson, L. Faoro,

E. Farhi, A. G. Fowler, B. Foxen, M. Giustina, A. Greene, J. A. Gross, M. P. Harrigan, S. D. Harrington, J. Hilton, S. Hong, T. Huang, W. J. Huggins, L. B. Ioffe, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, T. Khattar, S. Kim, P. V. Klimov, A. N. Korotkov, F. Kostritsa, D. Landhuis, P. Laptev, A. Locharla, E. Lucero, O. Martin, J. R. McClean, M. McEwen, K. C. Miao, M. Mohseni, S. Montazeri, W. Mruczkiewicz, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, T. E. O'Brien, A. Opremcak, B. Pató, A. Petukhov, N. C. Rubin, D. Sank, V. Shvarts, D. Strain, M. Szalay, B. Villalonga, T. C. White, Z. Yao, P. Yeh, J. Yoo, A. Zalcman, H. Neven, S. Boixo, A. Megrant, Y. Chen, J. Kelly, V. Smelyanskiy, A. Kitaev, M. Knap, F. Pollmann, and P. Roushan. "Realizing topologically ordered states on a quantum processor". Science **374**, 1237–1241 (2021).

[27] Laird Egan, Dripto M. Debroy, Crystal Noel, Andrew Risinger, Daiwei Zhu, Debopriyo Biswas, Michael Newman, Muyuan Li, Kenneth R. Brown, Marko Cetina, and Christopher Monroe. "Fault-tolerant control of an error-corrected qubit". Nature **598**, 281–286 (2021).

[28] Zijun Chen, Kevin J. Satzinger, Juan Atalaya, Alexander N. Korotkov, Andrew Dunsworth, Daniel Sank, Chris Quintana, Matt McEwen, Rami Barends, Paul V. Klimov, Sabrina Hong, Cody Jones, Andre Petukhov, Dvir Kafri, Sean Demura, Brian Burkett, Craig Gidney, Austin G. Fowler, Alexandru Paler, Harald Putterman, Igor Aleiner, Frank Arute, Kunal Arya, Ryan Babbush, Joseph C. Bardin, Andreas Bengtsson, Alexandre Bourassa, Michael Broughton, Bob B. Buckley, David A. Buell, Nicholas Bushnell, Benjamin Chiaro, Roberto Collins, William Courtney, Alan R. Derk, Daniel Eppens, Catherine Erickson, Edward Farhi, Brooks Foxen, Marissa Giustina, Ami Greene, Jonathan A. Gross, Matthew P. Harrigan, Sean D. Harrington, Jeremy Hilton, Alan Ho, Trent Huang, William J. Huggins, L. B. Ioffe, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Kostyantyn Kechedzhi, Seon Kim, Alexei Kitaev, Fedor Kostritsa, David Landhuis, Pavel Laptev, Erik

Lucero, Orion Martin, Jarrod R. McClean, Trevor McCourt, Xiao Mi, Kevin C. Miao, Masoud Mohseni, Shirin Montazeri, Wojciech Mruczkiewicz, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Michael Newman, Murphy Yuezhen Niu, Thomas E. O'Brien, Alex Opremcak, Eric Ostby, Bálint Pató, Nicholas Redd, Pedram Roushan, Nicholas C. Rubin, Vladimir Shvarts, Doug Strain, Marco Szalay, Matthew D. Trevithick, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Juhwan Yoo, Adam Zalcman, Hartmut Neven, Sergio Boixo, Vadim Smelyanskiy, Yu Chen, Anthony Megrant, and Julian Kelly. "Exponential suppression of bit or phase errors with cyclic error correction". Nature **595**, 383–387 (2021).

[29] Alexander Erhard, Hendrik Poulsen Nautrup, Michael Meth, Lukas Postler, Roman Stricker, Martin Stadler, Vlad Negnevitsky, Martin Ringbauer, Philipp Schindler, Hans J. Briegel, Rainer Blatt, Nicolai Friis, and Thomas Monz. "Entangling logical qubits with lattice surgery". Nature **589**, 220–224 (2021).

[30] Ming Gong, Xiao Yuan, Shiyu Wang, Yulin Wu, Youwei Zhao, Chen Zha, Shaowei Li, Zhen Zhang, Qi Zhao, Yunchao Liu, Futian Liang, Jin Lin, Yu Xu, Hui Deng, Hao Rong, He Lu, Simon C Benjamin, Cheng-Zhi Peng, Xiongfeng Ma, Yu-Ao Chen, Xiaobo Zhu, and Jian-Wei Pan. "Experimental exploration of five-qubit quantum error correcting code with superconducting qubits". National Science ReviewPage nwab011 (2021).

[31] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, N. C. Brown, T. M. Gatterman, S. K. Halit, K. Gilmore, J. A. Gerber, B. Neyenhuis, D. Hayes, and R. P. Stutz. "Realization of real-time fault-tolerant quantum error correction". Physical Review X **11**, 041058 (2021).

[32] J. F. Marques, B. M. Varbanov, M. S. Moreira, H. Ali, N. Muthusubramanian, C. Zachariadis, F. Battistel, M. Beekman, N. Haider,

W. Vlothuizen, A. Bruno, B. M. Terhal, and L. DiCarlo. "Logical-qubit operations in an error-detecting surface code". Nature Physics **18**, 80–86 (2021).

[33] Lukas Postler, Sascha Heußen, Ivan Pogorelov, Manuel Rispler, Thomas Feldker, Michael Meth, Christian D. Marciniak, Roman Stricker, Martin Ringbauer, Rainer Blatt, Philipp Schindler, Markus Müller, and Thomas Monz. "Demonstration of fault-tolerant universal quantum gate operations" (2021). arXiv:2111.12654.

[34] Sebastian Krinner, Nathan Lacroix, Ants Remm, Agustin Di Paolo, Elie Genois, Catherine Leroux, Christoph Hellings, Stefania Lazar, Francois Swiadek, Johannes Herrmann, Graham J. Norris, Christian Kraglund Andersen, Markus Müller, Alexandre Blais, Christopher Eichler, and Andreas Wallraff. "Realizing Repeated Quantum Error Correction in a Distance-Three Surface Code" (2021). arXiv:2112.03708.

[35] Dolev Bluvstein, Harry Levine, Giulia Semeghini, Tout T. Wang, Sepehr Ebadi, Marcin Kalinowski, Alexander Keesling, Nishad Maskara, Hannes Pichler, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. "A quantum processor based on coherent transport of entangled atom arrays". Nature **604**, 451–456 (2022).

[36] Paul Benioff. "The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines". Journal of Statistical Physics **22**, 563–591 (1980).

[37] Yuri Manin. "Computable and uncomputable (in russian)". Sovetskoye Radio, Moscow (1980).

[38] Román Orús, Samuel Mugel, and Enrique Lizaso. "Quantum computing for finance: Overview and prospects". Reviews in Physics **4**, 100028 (2019).

[39] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan. "Quantum Algorithms for Quantum Chemistry and Quantum Materials Science". Chemical Reviews **120**, 12685–12717 (2020).

[40] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, A. Aspuru-Guzik, and A. G. White. "Towards quantum chemistry on a quantum computer". Nature Chemistry **2**, 106–111 (2010).

[41] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. "Quantum computational chemistry". Reviews of Modern Physics **92**, 015003 (2020).

[42] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. "Variational quantum algorithms". Nature Reviews Physics **3**, 625–644 (2021).

[43] E. Knill. "Non-binary unitary error bases and quantum codes" (1996). arXiv:quant-ph/9608048.

[44] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver. "A quantum engineer's guide to superconducting qubits". Applied Physics Reviews **6**, 021318 (2019).

[45] G Wendin. "Quantum information processing with superconducting circuits: a review". Reports on Progress in Physics **80**, 106001 (2017).

[46] Anton Frisk Kockum and Franco Nori. "Quantum bits with josephson junctions". In Fundamentals and Frontiers of the Josephson Effect. Pages 703–741. Springer International Publishing (2019).

[47] D. Leibfried, R. Blatt, C. Monroe, and D. Wineland. "Quantum dynamics of single trapped ions". Reviews of Modern Physics **75**, 281–324 (2003).

[48] R. Blatt and C. F. Roos. "Quantum simulations with trapped ions". Nature Physics **8**, 277–284 (2012).

[49] Xi-Lin Wang, Yi-Han Luo, He-Liang Huang, Ming-Cheng Chen, Zu-En Su, Chang Liu, Chao Chen, Wei Li, Yu-Qiang Fang, Xiao Jiang, Jun Zhang, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. "18-qubit entanglement with six photons' three degrees of freedom". Physical Review Letters**120** (2018).

[50] Daniel Loss and David P. DiVincenzo. "Quantum computation with quantum dots". Physical Review A **57**, 120–126 (1998).

[51] B. E. Kane. "A silicon-based nuclear spin quantum computer". Nature **393**, 133–137 (1998).

[52] Ioan M. Pop, Kurtis Geerlings, Gianluigi Catelani, Robert J. Schoelkopf, Leonid I. Glazman, and Michel H. Devoret. "Coherent suppression of electromagnetic dissipation due to superconducting quasiparticles". Nature **508**, 369–372 (2014).

[53] P Aliferis, F Brito, D P DiVincenzo, J Preskill, M Steffen, and B M Terhal. "Fault-tolerant computing with biased-noise superconducting qubits: a case study". New Journal of Physics **11**, 013061 (2009).

[54] M. D. Shulman, O. E. Dial, S. P. Harvey, H. Bluhm, V. Umansky, and A. Yacoby. "Demonstration of entanglement of electrostatically coupled singlet-triplet qubits". Science **336**, 202–205 (2012).

[55] H. Bombin. "An introduction to topological quantum codes" (2013). arXiv:1311.0277.

[56] S. B. Bravyi and A. Yu. Kitaev. "Quantum codes on a lattice with boundary" (1998). arXiv:quant-ph/9811052.

[57] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. "Topological quantum memory". Journal of Mathematical Physics **43**, 4452–4505 (2002).

[58] Austin G. Fowler. "Optimal complexity correction of correlated errors in the surface code" (2013). arXiv:1310.0863.

[59] Ben Criger and Imran Ashraf. "Multi-path summation for decoding 2D topological codes". Quantum **2**, 102 (2018).

[60] Jack Edmonds. "Maximum matching and a polyhedron with 0, 1-vertices". Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics **69B**, 125 (1965).

[61] Jack Edmonds. "Paths, trees, and flowers". Canadian Journal of Mathematics **17**, 449–467 (1965).

[62] Austin G. Fowler. "Minimum weight perfect matching of fault-tolerant topological quantum error correction in average $O(1)$ parallel time". Quantum Information and Computation **15**, 145–158 (2013).

[63] Oscar Higgott. "Pymatching: A python package for decoding quantum codes with minimum-weight perfect matching" (2021). arXiv:2105.13082.

[64] Guillaume Duclos-Cianci and David Poulin. "Fast decoders for topological quantum codes". Physical Review Letters **104**, 050504 (2010).

[65] Nicolas Delfosse and Naomi H. Nickerson. "Almost-linear time decoding algorithm for topological codes". Quantum **5**, 595 (2021).

[66] Shahnawaz Ahmed, Carlos Sánchez Muñoz, Franco Nori, and Anton Frisk Kockum. "Quantum state tomography with conditional generative adversarial networks". Physical Review Letters **127**, 140502 (2021).

[67] Philip Andreasson, Joel Johansson, Simon Liljestrand, and Mats Granath. "Quantum error correction for the toric code using deep reinforcement learning". Quantum **3**, 183 (2019).

[68] David Fitzek, Mattias Eliasson, Anton Frisk Kockum, and Mats Granath. "Deep Q-learning decoder for depolarizing noise on the toric code". Physical Review Research **2**, 023230 (2020).

[69] Emanuel Knill, Raymond Laflamme, and Wojciech H. Zurek. "Resilient quantum computation". Science **279**, 342–345 (1998).

[70] Dorit Aharonov and Michael Ben-Or. "Fault-tolerant quantum computation with constant error rate". SIAM Journal on Computing **38**, 1207–1282 (2008).

[71] Ashley M. Stephens. "Fault-tolerant thresholds for quantum error correction with the surface code". Physical Review A **89**, 022321 (2014).

[72] Basudha Srivastava, Anton Frisk Kockum, and Mats Granath. "The XYZ^2 hexagonal stabilizer code". Quantum **6**, 698 (2022).

[73] James R Wootton. "A family of stabilizer codes for $D(\mathbb{Z}_2)$ anyons and Majorana modes". Journal of Physics A: Mathematical and Theoretical **48**, 215302 (2015).

[74] Sergey Bravyi, Martin Suchara, and Alexander Vargo. "Efficient algorithms for maximum likelihood decoding in the surface code". Physical Review A **90**, 032326 (2014).

[75] David Kingsley Tuckett. "Tailoring surface codes: Improvements in quantum error correction with biased noise". PhD thesis. University of Sydney. (2020).

[76] Karl Hammar, Alexei Orekhov, Patrik Wallin Hybelius, Anna Katariina Wisakanto, Basudha Srivastava, Anton Frisk

Kockum, and Mats Granath. "Error-rate-agnostic decoding of topological stabilizer codes". Physical Review A **105**, 042616 (2022).

[77] James R. Wootton and Daniel Loss. "High threshold error correction for the surface code". Physical Review Letters **109**, 160503 (2012).

[78] Adrian Hutter, James R. Wootton, and Daniel Loss. "Efficient markov chain monte carlo algorithm for the surface code". Physical Review A **89**, 022326 (2014).

[79] James R. Wootton. "Hexagonal matching codes with 2-body measurements" (2021). arXiv:2109.13308.

[80] David Poulin. "Optimal and efficient decoding of concatenated quantum block codes". Physical Review A **74**, 052333 (2006).

[81] Ingemar Bengtsson and Basudha Srivastava. "Dimension towers of SICS: II. Some constructions". Journal of Physics A: Mathematical and Theoretical **55**, 215302 (2022).

Part III
Research papers

Paper A

The XYZ^2 hexagonal stabilizer code
Basudha Srivastava, Anton Frisk Kockum, and Mats Granath
Quantum **6**, 698 (2022).

Paper B

Error-rate-agnostic decoding of topological stabilizer codes
Karl Hammar, Alexei Orekhov, Patrik Wallin Hybelius, Anna Katariina Wisakanto, Basudha Srivastava, Anton Frisk Kockum, and Mats Granath
Physical Review A **105**, 042616 (2022).