On novel techniques for Quantum Error Correction

Mahya Jamshidian

1 Introduction

Quantum error-correcting codes (QECCs) are a family of quantum algorithms that are designed to protect quantum information from the effects of noise and errors that arise in quantum systems. Just as classical error-correcting codes (ECCs) are used to protect classical information in communication and computing systems, QECCs are used to protect quantum information in quantum computing and communication systems (Kaye et al. [2007]).

The main challenge of quantum error correction is that quantum information cannot be measured or copied without disturbing its state. This is due to the fundamental principles of quantum mechanics, which state that the act of measurement changes the state of a quantum system. As a result, traditional error-correcting techniques used in classical computing, such as redundancy and error detection, are not directly applicable in the quantum case.

To address this challenge, QECCs use a combination of quantum entanglement, quantum error detection, and quantum error correction techniques to protect quantum information from noise and errors. The basic idea is to encode the quantum information into a larger quantum system, called a quantum code, in such a way that errors can be detected and corrected without measuring the encoded quantum state. This is achieved by exploiting the properties of entanglement, which allow the encoded quantum state to be protected by detecting and correcting errors on a smaller subsystem of the larger system.

The hardware devices used to implement quantum computing are very susceptible to noise, which can affect the state of qubits during the execution of a quantum circuit. For this reason, the ability to detect and correct these errors is a topic of great relevance in quantum computing. Error correction has been studied extensively for classical computers, so one might think of applying the knowledge we already have for classical error correction to correct quantum errors. Unfortunately, in the quantum case, there are some more issues that complicate error correction:

- We cannot introduce redundancy by copying arbitrary qubits, because of the no-cloning theorem
- We cannot measure a qubit disturbing the superposition
- An error on a qubit is a continuous error while in classical computing we only have discrete bit flip errors

To protect a quantum state $|\psi\rangle$ from errors, the first step is to define an error model that describes the type of error that might occur on the state. After defining the error model, an encoding operator ENC is used to add extra ancillary qubits to the state and produce an encoded state $\left|\psi'\right\rangle$. The space of all possible encoded states forms a subspace of the Hilbert space, which is referred to as the quantum error correcting code or simply code. If the input state $|\psi\rangle$ has k qubits and n qubits are added during encoding, the code space $\mathcal{C}$ will be a subspace of the 2^{k+n}-dimensional Hilbert space. Encoding the input state in a higher-dimensional state allows errors in the encoded state to leave information that can be used to identify their effects, referred to as error syndrome. A recovery operator $\mathcal{R}$ is defined to undo the effect of the error if it occurs on the encoded state $\left|\psi'\right\rangle$. Finally, the decoding operator ENC is used to decode $|\psi'\rangle$ back to $|\psi\rangle$ (Gottesman [1997]).

In the case of correcting bit flip errors, the simple error is where a qubit in the state $\alpha|0\rangle + \beta|1\rangle$ is transformed into $\beta|0\rangle + \alpha|1\rangle$, which is equivalent to applying the X Pauli gate to the qubit. To correct this error, a three-qubit encoding is used for the states $|0\rangle$ and $|1\rangle$, where the physical states are encoded respectively in the logical states $|000\rangle$ and $|111\rangle$. To correct the error, the values of the three qubits are checked, and if they are not the same, the value of the majority qubit is imposed. This is done by storing the information of which qubit of the state has been flipped in some extra qubits, which serves as the error syndrome.

2 Types of Quantum Errors

Quantum systems are inherently susceptible to a variety of errors that arise from noise and decoherence. These errors can be broadly classified into two categories: stochastic errors and systematic errors.

Stochastic errors are random and unpredictable, and can occur due to environmental factors such as thermal fluctuations and electromagnetic interference. These errors are typically modeled as a probability distribution that describes the likelihood of a given error occurring at any point in time. Stochastic errors can be reduced by improving the stability of the quantum system or by using error-correcting codes that can detect and correct errors.

Systematic errors, on the other hand, are deterministic and arise from imperfections in the design or calibration of the quantum system. These errors are typically modeled as a fixed bias or offset that affects the measurement or operation of the quantum system. Systematic errors can be reduced by improving the design and calibration of the quantum system or by using error mitigation techniques that can correct for the effects of systematic errors.

Another important distinction between types of quantum errors is between coherent errors and incoherent errors. Coherent errors are errors that preserve the quantum coherence of the system, such as phase errors, while incoherent errors are errors that destroy the quantum coherence of the system, such as amplitude damping and depolarizing errors.

Some common types of quantum errors include:

1. **Depolarizing errors:** These errors occur when the quantum state is subjected to random rotations around any axis in the Bloch sphere, leading to a loss of information.
2. **Amplitude damping errors:** These errors occur when the quantum state loses energy to the environment, leading to a decay in the amplitude of the state.
3. **Phase errors:** These errors occur when the phase of the quantum state is rotated by an angle that is different from the intended value.
4. **Measurement errors:** These errors occur when the measurement process introduces noise or disturbance into the quantum state being measured.
5. **Control errors**: These errors occur when the control parameters used to manipulate the quantum state, such as gate durations or pulse amplitudes, are not precisely tuned.
6. **Crosstalk errors:** These errors occur when the interactions between different components of the quantum system lead to unintended changes in the state of the system.

There are three types of quantum errors when on execution that are generated as a result of the environmental effects mentioned above:

1. **Bit flip error (X error):** A bit flip error occurs when the value of a qubit is flipped from $|0\rangle$ to $|1\rangle$ or vice versa.
2. **Phase flip error (Z error):** A phase flip error occurs when the phase of a qubit is flipped from $|0\rangle$ to $|1\rangle$r vice versa.
3. **Bit-phase flip error (Y error):** A bit-phase flip error occurs when both the bit and phase of a qubit are flipped.

3 Quantum Error-Correcting Codes

Quantum error-correcting codes are a family of techniques used to protect quantum information from the effects of noise and errors. These codes encode the original quantum information into a larger quantum system, allowing for the detection and correction of errors that may arise during the processing of the information. There are several types of quantum error-correcting codes, each with its own advantages and disadvantages (Nielsen and Chuang [2002]).

- **The three-qubit bit-flip code:** This is the simplest type of quantum error-correcting code, consisting of three qubits. The code is designed to correct errors caused by a bit-flip, where a qubit changes from the state $|0\rangle$ to the state $|1\rangle$, or vice versa. The code works by encoding a single qubit in a three-qubit state, such that any bit-flip error can be detected and corrected.
- **The five-qubit code:** This code is designed to correct errors caused by both bit-flip and phase-flip errors. The code encodes a single qubit in a five-qubit state, allowing for the detection and correction of both types of errors.
- **The surface code:** This is a two-dimensional code that is particularly effective at correcting errors caused by noise and decoherence. The code works by encoding a qubit in a two-dimensional array of qubits, such that any errors can be detected and corrected by measuring the syndrome of the code.
- **The topological code:** This is a class of codes that are particularly robust to noise and errors, and are inspired by the principles of topology. These codes rely on the use of non-local properties of the quantum system, such as the braiding of anyons, to protect the information from errors (Théveniaut and van Nieuwenburg [2021]).
- **The Calderbank-Shor-Steane (CSS) code:** This code is a generalization of the code that is particularly effective at correcting errors caused by coherent errors. The code works by encoding the quantum information into two subsystems, and using the difference between the subsystems to detect and correct errors (Ostrev et al. [2022]).

In addition to these specific codes, there are also a variety of hybrid codes that combine elements of different codes to provide improved performance in specific contexts. Overall, the choice of quantum error-correcting code depends on the specific requirements of the quantum system, as well as the types of errors that are expected to occur.

3.1 Three-qubit bit-flip code

The 3-qubit code is a quantum error correction code that encodes a single qubit in a three-qubit system. The code is capable of detecting and correcting a single bit flip error, where a single qubit flips from 0 to 1 or from 1 to 0.

The encoding process for the 3-qubit code is as follows. Suppose we have a single qubit that we wish to encode, with state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. We can encode this state into a three-qubit system using the following circuit (Fig 1):

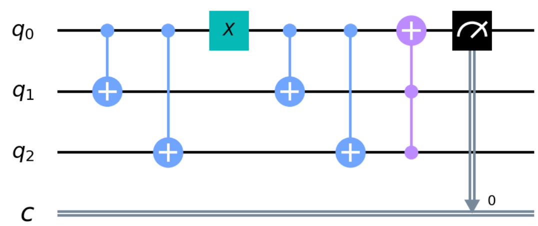

Figure 1: three-qubit encoding circuit

Here, the qubit to be encoded is initially in the state $|\psi\rangle$, and the three qubits are initialized to the state $|000\rangle$. The first two qubits are entangled using a CNOT gate, with the control qubit being the first qubit and the target qubit being the second qubit. The first qubit is then further entangled with the third qubit using another CNOT gate, with the control qubit being the first qubit and the target qubit being the third qubit. Finally, a Hadamard gate is applied to the first qubit.

The resulting state of the three-qubit system after the encoding process is:

$$|\psi_E\rangle = \frac{1}{2}\left[(1+\alpha)|000\rangle + (1-\alpha)|011\rangle + (1-\beta)|101\rangle + (1+\beta)|110\rangle\right] \tag{1}$$

This encoded state is now more robust to errors, as any bit flip error affecting one of the qubits will result in a change of sign in one of the terms above, allowing us to detect and correct the error.

Suppose now that we have an error in the three-qubit system, where one of the qubits flips from 0 to 1 or from 1 to 0. We can detect this error by measuring the three qubits in the computational basis. If the measurement outcome is not $|000\rangle$, then we know that an error has occurred. However, we cannot determine which qubit is affected by the error using this method alone.

To correct the error, we can use the following procedure:

1. Measure the three qubits in the computational basis.
2. If the measurement outcome is not $|000\rangle$, then we know that an error has occurred. Determine which qubit is affected by the error by comparing the measurement outcome to the encoded state $|\psi_E\rangle$.
3. Flip the affected qubit to correct the error.

3.1.1 Examples

To illustrate the error correction process, let's consider a specific example of the 3-qubit error correcting code. Suppose that we have encoded the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ using the 3-qubit code.

The encoding process for the 3-qubit code can be described by the following equations:

$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$
$$|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$$

Suppose that the encoded state is $|\psi_E\rangle = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \beta|111\rangle)$. Now, let's assume that an error occurs on one of the qubits. For the sake of this example, let's suppose that the error affects the second qubit, causing it to flip from $|0\rangle$ to $|1\rangle$. The resulting state is:

$$|\psi'_E\rangle = \frac{1}{\sqrt{2}}(\alpha|010\rangle + \beta|101\rangle)$$

To correct this error, we first apply a set of Hadamard gates to the first and second qubits:

$$|\psi'_E\rangle \to |\psi_1\rangle = \frac{1}{2}[(\alpha|0\rangle + \beta|1\rangle)|00\rangle + (\alpha|0\rangle - \beta|1\rangle)|01\rangle + (\alpha|1\rangle + \beta|0\rangle)|10\rangle - (\alpha|1\rangle - \beta|0\rangle)|11\rangle]$$

We can then measure the first and second qubits and obtain a two-bit string that corresponds to the error syndrome. There are four possible syndromes:

- **Syndrome 00:** Both qubits are measured in the state $|0\rangle$, so we do nothing.
- **Syndrome 01:** The first qubit is measured in the state $|0\rangle$, and the second qubit is measured in the state $|1\rangle$, then we should apply an X gate to the third qubit.
- **Syndrome 10:** The first qubit is measured in the state $|1\rangle$, and the second qubit is measured in the state $|0\rangle$, then we must apply Z to the third qubit.
- **Syndrome 11:** Both qubits are measured in the state $|1\rangle$ and we apply a Y gate to the third qubit.

In our example, the syndrome is 01, so we apply an X gate to the third qubit:

$$|\psi_1\rangle \to |\psi_2\rangle = \frac{1}{2}[(\alpha|0\rangle + \beta|1\rangle)|00\rangle + (\alpha|1\rangle - \beta|0\rangle)|01\rangle + (\alpha|1\rangle + \beta|0\rangle)|10\rangle - (\alpha|0\rangle - \beta|1\rangle)|11\rangle]$$

The resulting state $|\psi_2\rangle$ is the corrected state, and we can measure it to extract the original state $|\psi\rangle$. To do so, we can use the following measurement:

$$\begin{aligned}
M_{000} &= |000\rangle\langle 000| + |111\rangle\langle 111| \\
M_{001} &= |001\rangle\langle 001| + |110\rangle\langle 110| \\
M_{010} &= |010\rangle\langle 010| + |101\rangle\langle 101| \\
M_{011} &= |011\rangle\langle 011| + |100\rangle\langle 100| \\
M_{100} &= |100\rangle\langle 100| + |011\rangle\langle 011| \\
M_{101} &= |101\rangle\langle 101| + |010\rangle\langle 010| \\
M_{110} &= |110\rangle\langle 110| + |001\rangle\langle 001| \\
M_{111} &= |111\rangle\langle 111| + |000\rangle\langle 000|
\end{aligned}$$

where each M_{abc} measures the probability of obtaining the state $|abc\rangle$. After performing the measurement, we can use the majority rule to correct the state $|\psi_E\rangle$. If the measurement gives us, for example, $|010\rangle$, we know that the error has affected the second qubit, so we flip its value to obtain the corrected state $|\psi\rangle = |000\rangle$.

3.2 The five-qubit code

The five-qubit code is a quantum error-correcting code that can correct for single-qubit errors. It encodes a single logical qubit into five physical qubits, providing redundancy that can detect and correct errors.

The code is constructed using the following five-qubit state, which is an equal superposition of all the possible states with a single error:

$$|\psi_5\rangle = \frac{1}{\sqrt{5}}(|00000\rangle + |00111\rangle + |01010\rangle + |01101\rangle + |10011\rangle + |10100\rangle + |11001\rangle + |11110\rangle)$$

The five-qubit code can detect and correct single-qubit errors by applying the following syndrome measurement circuit (Fig 2).

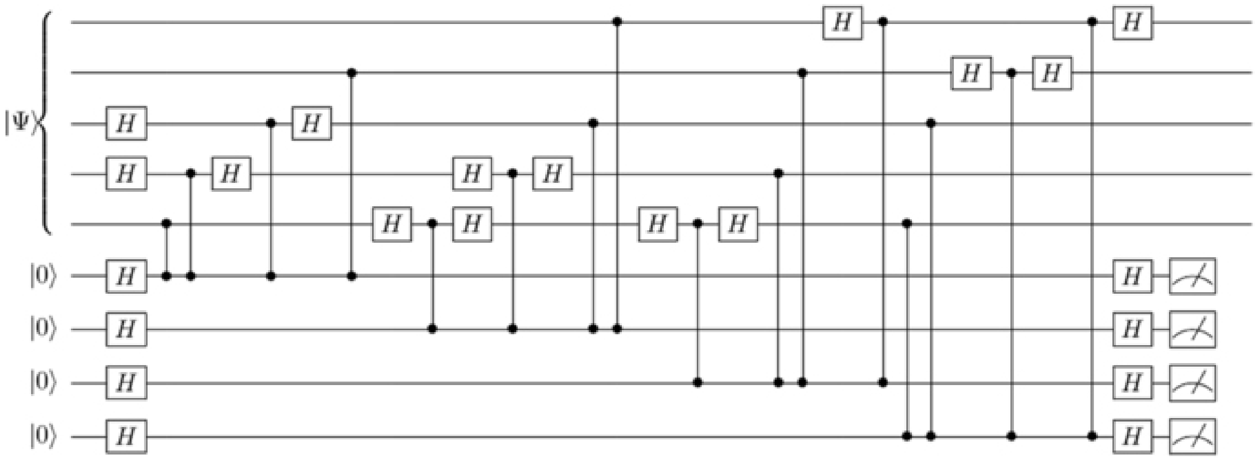

Figure 2: 5-qubit encoding circuit

In this circuit, each physical qubit is connected to two ancilla qubits. The X and Z gates are used to perform measurements in the X and Z bases, respectively, to obtain the syndrome information. The syndrome measurements are then used to determine the type of error that occurred and to correct it.

The five-qubit code algorithm consists of the following steps:

1. Prepare the five-qubit state $|\psi_5\rangle$.
2. Apply the unitary gate that represents the error that occurred. This will result in a new state $|\psi_5'\rangle$.
3. Apply the syndrome measurement circuit to obtain the syndrome information.
4. Determine the type of error that occurred based on the syndrome information and apply the correction. This can be done using the following table:
5. Verify that the error has been corrected by measuring the state in the computational basis.

The five-qubit code algorithm is a stabilizer code, which means that it encodes the state of a qubit into a subspace of the larger Hilbert space that is invariant under certain stabilizer operators. These operators form a group, known as the stabilizer group, that commutes with the code's logical operators. This allows the code to detect and correct errors by measuring the stabilizer group and applying appropriate corrections.

Syndrome	Error	Correction
00000	No error	$X_2X_3X_4$
00011	X_1 error	$X_1X_3X_4$
00101	X_2 error	$X_1X_2X_4$
00110	X_3 error	$X_1X_2X_3$
01001	X_4 error	$Z_2Z_3Z_4$
01010	Z_1 error	$Z_2Z_3Z_4$
01100	Z_2 error	$Z_1Z_3Z_4$
10001	Z_3 error	$Z_1Z_2Z_4$
10010	Z_4 error	$Z_1Z_2Z_3$
11111	$X_1X_2X_3X_4$ error	$Z_1Z_2Z_3Z_4$

Table 1: Part of the syndrome table for the five-qubit code algorithm.

3.2.1 Example

Suppose we want to encode the state $|\psi\rangle = |1\rangle$ using the five-qubit code. The encoded state is:

$$\psi_5 = \frac{1}{\sqrt{8}}(|00000\rangle + |00111\rangle + |01010\rangle + |01101\rangle + |10011\rangle + |10100\rangle + |11001\rangle + |11110\rangle)$$

Now, suppose there is an error in the first qubit of the encoded state, which can be represented by the Pauli-X operator. To measure the syndrome in the X basis, we apply a Hadamard gate to each qubit of the encoded state, and measure each qubit in the X basis:

$$\begin{aligned} H^{\otimes 5}|\psi_5\rangle = \frac{1}{\sqrt{8}}(&H^{\otimes 5}|00000\rangle + H^{\otimes 5}|00111\rangle + H^{\otimes 5}|01010\rangle + H^{\otimes 5}|01101\rangle \\ &+ H^{\otimes 5}|10011\rangle + H^{\otimes 5}|10100\rangle + H^{\otimes 5}|11001\rangle + H^{\otimes 5}|11110\rangle) \end{aligned}$$

then we have

$$\begin{aligned} H^{\otimes 5}|\psi_5\rangle =& \frac{1}{\sqrt{8}}(|+++++\rangle + |++---\rangle + |+-+++\rangle + |+-+-+\rangle \\ &+ |-+++-\rangle + |+--++\rangle + |-+-+-\rangle + |--+++\rangle) \end{aligned}$$

Suppose the measurement outcome is $|+++++\rangle$. This means that there are no X errors on any of the qubits. If the measurement outcome is any of the other seven possible outcomes, then there is a single X error on one of the qubits. From the table in the previous message, we can see that the syndrome $|++---\rangle$ corresponds to an X_1 error, so we know that the error occurred on the first qubit. To correct the error, we apply a Pauli-X gate to the first qubit of the encoded state. We can then apply the inverse of the Hadamard gate on all five qubits to return to the original basis.

3.3 Surface Codes

The surface code is a type of quantum error-correcting code that is designed to correct errors arising from the interaction of qubits with their environment. It was introduced in 1997 by Alexei Kitaev, and is one of the most promising quantum error-correcting codes due to its ability to protect qubits against a wide range of errors.

The surface code can be thought of as a two-dimensional lattice of qubits, with each qubit having four neighboring qubits. These qubits are arranged in a checkerboard pattern, with alternating rows and columns of qubits. The surface code is a type of topological code, which means that it is designed to protect against errors that change the topology of the qubit lattice.

To understand how the surface code works, it is helpful to consider a simple example. Let's say we have a 3 x 3 grid of qubits, as shown in figure 3.

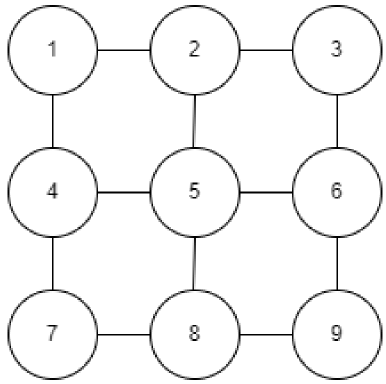

Figure 3: Graph state of a simple surface code

Each qubit in the grid has four neighboring qubits, which are connected by horizontal and vertical lines. We can think of these lines as edges in a graph, where each qubit is a vertex and the edges connect neighboring vertices.

To encode a logical qubit using the surface code, we first prepare the qubits in a certain way. We apply a Hadamard gate to each qubit, which puts it into an equal superposition of the —0⟩ and —1⟩ states. We then apply a CNOT gate to each pair of neighboring qubits, which entangles them. This creates a state that is a superposition of all possible states of the qubits, with each state occurring with equal probability.

To detect and correct errors in the encoded qubit, we measure the state of each qubit in the lattice. If a qubit is measured to be in the $|0\rangle$ state, we say that it has a Z error. If it is measured to be in the $|1\rangle$ state, we say that it has a Z error. We also measure the state of each edge in the lattice, which can be thought of as measuring the parity of the qubits on either side of the edge. If an edge is measured to have an odd parity, we say that it has an X error.

3.4 Topological error correcting codes

Topological error-correcting codes are based on the concept of a topological order, which describes how the quantum states of a system are organized. In a topologically ordered system, local perturbations have only a limited effect on the overall state of the system. This is because the system is characterized by long-range correlations that are protected by the topology of the system.

The basic idea of topological error-correcting codes is to encode a logical qubit into a set of physical qubits that are arranged in a topologically ordered pattern. The encoded qubit is then protected by the topology of the pattern, which ensures that local perturbations do not affect the logical qubit. To perform error correction, measurements are made on a subset of the physical qubits, and the results of these measurements are used to correct any errors that may have occurred.

One of the simplest examples of a topological error-correcting code is the toric code. The toric code is a 2D lattice of physical qubits arranged in a torus shape. Each qubit has four neighbors,

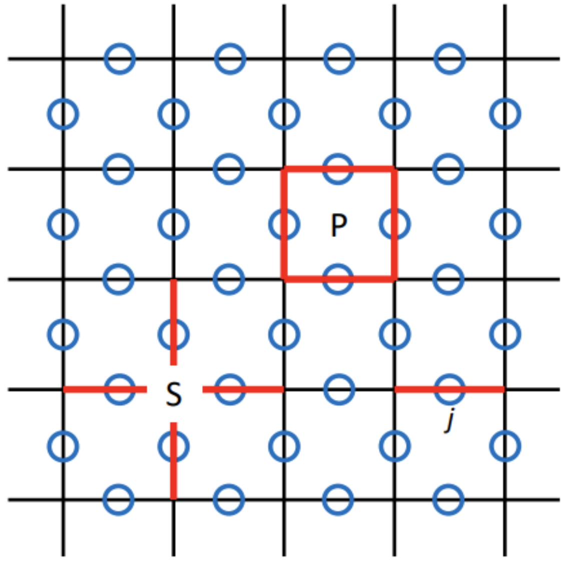

Figure 4: : Illustration of the toric code model on a square lattice. Circles indicate the positions of spin-1/2 particles on the bonds of the lattice; j labels the spin located at the center of the jth lattice bond; P labels a plaquette containing a total of four spins; S labels a 'star' also containing a total of four spins.

and each pair of neighboring qubits is connected by a virtual link. The logical qubit is encoded in the pattern of virtual links, rather than in the physical qubits themselves (Figure 4).

To perform error correction on the toric code, measurements are made on a set of plaquettes, which are square regions of four physical qubits. Each plaquette is measured in the X and Z basis, and the results of these measurements are used to determine whether any errors have occurred. If an error is detected, it can be corrected by flipping the state of the affected qubits.

3.4.1 NEAT Code

NEAT (NeuroEvolution of Augmenting Topologies) is a type of artificial intelligence algorithm used to evolve artificial neural networks. In NEAT, a population of neural networks is evolved through a process of selection, reproduction, and mutation. Each network is assigned a fitness score based on how well it performs a given task. The networks with the highest fitness scores are selected to reproduce and pass on their genetic material to the next generation.

NEAT uses a genetic algorithm to create and modify the structure of neural networks. This is done by starting with a population of small, simple networks and adding complexity over time. This allows the network to gradually learn more complex patterns and behaviors.

NEAT also uses a technique called speciation to maintain diversity in the population. Speciation involves grouping similar neural networks into species and giving them a separate fitness score. This encourages the evolution of a diverse set of neural networks, preventing the population from converging on a single solution.

One of the advantages of NEAT is that it is a highly adaptive algorithm that can evolve networks for a wide range of tasks. It is also able to learn quickly and efficiently, often outperforming other machine learning algorithms on certain tasks.

3.5 CSS Code

The CSS code is based on the idea of combining two classical codes to create a quantum code. The first code is a binary linear code that corrects errors in the X basis (i.e., errors that flip the value of a qubit). The second code is a binary linear code that corrects errors in the Z basis (i.e., errors that change the phase of a qubit).

To construct a CSS code, we take the dual code of the X code and the original Z code, and concatenate them to form a new code. The dual code is simply the code where the roles of 0's and 1's are swapped, and the addition operation is replaced with the exclusive-or (XOR) operation.

For example, let's consider a CSS code that uses the [7,4] Hamming code for the X code and the [7,3] repetition code for the Z code. The dual of the [7,4] Hamming code is the [7,3] code, and the dual of the [7,3] repetition code is itself.

To encode a qubit using the CSS code, we first apply the X code to the qubit, and then apply the Z code to the resulting state. To correct errors, we first apply the Z code to the state, and then apply the X code.

The CSS code has the advantage that it can correct both X and Z errors, and can even correct errors in both bases simultaneously. It also has a higher code rate than the individual codes, meaning that it can protect more qubits with the same number of physical qubits.

4 Experiments

In this project, we explore various error-correcting codes for quantum information. We conduct five experiments on different codes: 3-qubit code, 5-qubit code, surface code, NEAT code, and [7,1,3] CSS code. Each experiment involves implementing the code in Qiskit and testing its error-correction capability.

In the first experiment, we implement a simple 3-qubit error-correcting code that can detect and correct one bit-flip error. We apply a Deutch-Josza oracle error to the encoded state to test the code's error correction capability.

In the second experiment, we implement a 5-qubit code, which is an extension of the 3-qubit code that can detect and correct one phase-flip error as well as one bit-flip error. We also test the code's error-correction capability under different error models.

In the third experiment, we implement a surface code, which is a topological error-correcting code that can detect and correct multiple errors. We apply different types of errors to the code and observe its ability to detect and correct them.

In the fourth experiment, we implement the NEAT code, which is a neural network-based error-correcting code that can learn to correct errors without explicitly encoding the qubits. We train the network to correct a specific set of errors and test its generalization ability.

In the fifth experiment, we implement a [7,1,3] CSS code, which is a stabilizer code that can detect and correct phase and bit-flip errors using the Calderbank-Shor-Steane (CSS) construction. We test the code's ability to correct errors and compare it with other codes.

5 Conclusion

In conclusion, this project explored various topics in quantum error correction and implemented some of them using Qiskit. We started by introducing the need for error correction in quantum computers and the basic concepts of quantum error correction codes. We then discussed some important quantum error correction codes, including the surface code, topological codes, and the Calderbank-Shor-Steane (CSS) code.

We also explored the use of machine learning techniques, such as the NEAT method, for error correction in quantum computers. These methods can potentially offer improved performance over traditional quantum error correction codes.

Finally, we implemented some of these concepts using Qiskit and ran several experiments, including a simple 3-qubit code for the Deutsch-Jozsa algorithm with error correction, a 3-qubit bit-flip code, a 5-qubit error correcting code, and a [7,1,3] CSS code. We also implemented a surface code and a NEAT code.

References

Daniel Gottesman. Stabilizer codes and quantum error correction. *arXiv: Quantum Physics*, 1997.

Phillip Kaye, Raymond Laflamme, and Michele Mosca. *An Introduction to Quantum Computing*. Oxford University Press, Inc., USA, 2007. ISBN 0198570007.

Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

Dimiter Ostrev, Davide Orsucci, Francisco Lázaro, and Balazs Matuz. Classical product code constructions for quantum calderbank-shor-steane codes, 2022.

Hugo Théveniaut and Everard van Nieuwenburg. A NEAT quantum error decoder. *SciPost Physics*, 11(1), jul 2021. doi: 10.21468/scipostphys.11.1.005.