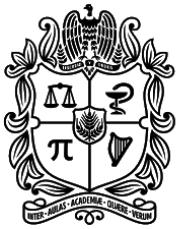

UNIVERSIDAD
NACIONAL
DE COLOMBIA

Quantum Error Correction via Quantum Convolutional Neural Networks

José Luis Falla León

Universidad Nacional de Colombia
Facultad de Ciencias - Física
Bogotá, Colombia
2024

Quantum Error Correction via Quantum Convolutional Neural Networks

José Luis Falla León

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:
Magister en Física

Director:
Dr. Carlos Leonardo Viviescas Ramírez

Línea de Investigación:
Computación Cuántica
Grupo de Investigación:
Caos y Complejidad

Universidad Nacional de Colombia
Facultad de Ciencias - Física
Bogotá, Colombia
2024

Para Carlos E. Murcia,
infinitamente más 1.

Resumen

Título: **Corrección de Error Cuántico Mediante Redes Neuronales Convolucionales Cuánticas**

Como una subclase de algoritmos cuánticos variacionales (ACVs), las redes neuronales convolucionales cuánticas (RNCCs), han surgido como un algoritmo eficiente de corrección de errores cuánticos (QEC) y como un código de corrección de errores cuánticos completo. A través de la optimización híbrida cuántico-clásica de una arquitectura RNCC para un modelo de error en particular, es posible "entrenar" una red neuronal para reducir las tasas de error lógico para modelos de errores específicos. Entrando a la era de la tecnología cuántica ruidosa de escala intermedia (CREI), la corrección de errores cuánticos es necesaria para la computación cuántica precisa con qubits ruidosos, y los ACVs pueden propiciar una computación cuántica confiable a corto plazo y a escala intermedia.
Palabras clave: *corrección de error cuántico; redes neuronales convolucionales cuánticas; computación cuántica; algoritmos cuánticos.*

Abstract

A sub-class of variational quantum algorithms (VQAs), the quantum convolutional neural network (QCNN), has emerged as an efficient quantum error correction (QEC) algorithm and full quantum error-correcting code. Through hybrid quantum-classical optimization of a QCNN architecture for a particular error model, it is possible to "train" a neural network to decrease the logical error rates for specific error models. Going into the noisy intermediate-scale quantum (NISQ) technology era, effective quantum error correction is necessary for accurate quantum computing with noisy qubits, and VQAs can bring about near-term, intermediate-scale, reliable quantum computing.

Keywords: *quantum error correction; quantum convolutional neural network; quantum computing; quantum algorithms.*

Contents

1 Introduction **7**

2 Quantum Noise **11**
2.1 Building the Basis for Quantum Noise 12
2.2 Quantum Operations 14
2.3 Operator-Sum Representation 15
2.4 Examples of Quantum Noise and Quantum Operations 17
2.4.1 Error Channels 17
2.4.2 Trace and Partial Trace 20
2.4.3 Master Equation Approach 21

3 Quantum Error Correction **24**
3.1 Constructing Error-Correcting Codes 26
3.1.1 The 3-Qubit Code 26
3.1.2 The 9-Qubit Code 28
3.2 Error Correction vs. Error Mitigation 29

4 Neural Networks **30**
4.1 The Brain as a Model for Neural Networks 31
4.2 From Biological to Artificial 31
4.3 Mathematical Formulation of Neural Networks 33
4.3.1 Single Neuron 33
4.3.2 Single Layer 34
4.4 Convolutional Neural Networks 35

5 Variational Quantum Algorithms **37**
5.1 Quantum Variational Error Correction Algorithm 38
5.2 Quantum Convolutional Neural Networks 38
5.2.1 Multiscale Entanglement Renormalization Ansatz . . . 39
5.2.2 Implementation of QCNN 39

6 Conclusions **45**

List of Figures

1 Classical Error Scheme 12
2 Effects of Quantum Error in a Simple Circuit 18
3 3-qubit Error-Correction Scheme 27
4 9-qubit Error-Correction Scheme 29
5 Biological Neuron 31
6 Neural Network 32
7 Single Artificial Neuron 34
8 Convolutional Neural Network 35
9 Multiscale Entanglement Renormalization Ansatz Representation 40
10 Quantum Convolutional Neural Network Architecture 41
11 Quantum Error Correction with QCNN/MERA Architecture . 42
12 3-qubit QCNN 43
13 Error Correction for 3-qubit QCNN 44
14 Decrease in Logical Error Rate for 3-qubit QCNN 45

List of Tables

1 Percent Logical Error Decrease in 3-qubit QCNN 45

1 Introduction

As we move into the intermediate noisy-scale quantum (NISQ) technology era [1], it is more pressing than ever to develop efficient quantum error-correction algorithms for near-term, reliable quantum processing. In recent years, the field of quantum error correction has seen promising advances from the use of machine learning for error mitigation. The quest for quantum neural networks is taking center stage as we approach this new technological era–one lying at the interface between artificial intelligence (AI), learning systems, quantum information theory, and emerging quantum technologies. With the revival of machine learning and neural networks as competent learning systems, it is crucial to investigate to what extent quantum information and quantum computation ideas can benefit from machine learning and neural network problems, and vice versa. In particular, it is important to understand whether quantum error correction can improve through the use of neural networks, given a neural network's ability to learn highly complex, non-linear functions.

It is reasonable to postulate that quantum computers may outperform classical computers on machine learning tasks [2], given that there is already sub-classes of problems for which quantum computers outperform classical machines [3]. Furthermore, quantum neural networks can be applied to many areas of research, highlighting their importance and relevance as an emerging technology. Many applications of machine learning and neural networks are geared towards condensed matter problems, as these problems often become intractable due to the exponentially-growing number of parameters needed to describe a quantum system [4–6]. Additionally, there is one particular area where quantum neural networks can provide significant advances: quantum error correction [7]. A quantum neural network can be constructed with quantum unitary gates and trained such that it is able to accurately detect and correct errors associated with particular error models.

Current implementations of medium- and large-scale quantum computing and quantum information processing rely heavily on effective quantum error correction schemes. For more than two decades now, many efforts have been made to determine effective quantum error correction algorithms [8–11]. The field of quantum error correction has been mostly theoretical, though more recent efforts focus on experimental quantum error correction schemes, with approaches ranging from the use of superconducting qubits [12] and trapped ions [13], to materials with long quantum coherence times [14]. One of the main issues to overcome in quantum error correction is the large number

of ancillary qubits required to correct errors without destroying the superposition of the encoded logical qubit. In general, quantum computation requires a large number of qubits in order to be a relevant technology; the "quantum supremacy" limit is set around 60 qubits [3].[1] As the number of qubits increases, so increases the possible errors due to decoherence. Additionally, it is experimentally complex to achieve full connectivity between all qubits. Currently, the largest fully-connected quantum computer is an 11-qubit progammable quantum computer in a trapped ion system composed of 13 $^{171}Yb^{+}$ ions [15]. Even though there are schemes to reduce the number of qubits involved in quantum error correction [16], it is still necessary to determine algorithms that will effectively detect and correct quantum errors.

The best resource to combat quantum error is entanglement (ironically, entanglement is also the reason for quantum error). By entangling logical qubits with ancilliary qubits, it is possible to measure these ancilla qubits and act on logical qubits based on measurement outcomes. The first *full* quantum error correction code developed, the 9-qubit Shor code [9], requires at least eight physical qubits in order to encode a single logical qubit; correcting for both phase-flip and bit-flip errors. Since the Hilbert space dimension scales as 2^N (where N is the number of qubits), it is clear that the use of large numbers of qubits makes this problem intractable, as the number of ancilla qubits would be impossible to achieve in the near future. Efficient QEC schemes are necessary moving forward, even in the NISQ technology era [17], in order to achieve fault-tolerant quantum computing. Therefore, one of the goals in the NISQ era is to extract the maximum quantum computational power from current devices, while developing techniques that might be suited for the long-term goal of fault-tolerant quantum computing [18]. In this regard, it is important to focus on the various types of error that can affect quantum information processing, including correlated errors, for which efficient quantum error-correcting algorithms are yet to be developed.

Even though there have been some efforts in trying to study and reduce correlated noise [19, 20], many schemes assume that decoherence only affects one qubit of the superposition, while the other qubits remain unchanged. The assumption that noise is not correlated stems from classical information theory's independence of noise [21]. At the macroscopic scale of classical systems, the theory of independence of noise is accurate, yet, based on the nature of quantum systems and quantum interactions, it is evident that this assump-

[1] As of the writing of this work, the largest quantum computer is IBM's 'Osprey' processor, containing 433 connected qubits.

tion does not necessarily hold true, as correlated errors can arise from a myriad of interactions. These interactions can range from "small scale", such as two-particle interacting via a dipole-dipole interaction, to "large scale", such as a system interacting with external fields. These types of interactions are common in open quantum systems and are considered sources of quantum noise. Even with current quantum control techniques, there are errors associated with physical realizations of quantum computer frameworks, quantum logic gates, and quantum circuits, to name a few, which arise from these unwanted interactions. Additional to devising novel techniques for keeping quantum systems "closed", it is necessary to develop effective schemes to detect and correct quantum errors, as most quantum system inevitably interact with the environment.

Recently, the idea of a quantum convolutional neural network has emerged as a viable scheme for quantum error correction. It has been shown that a QCNN is able to outperform other quantum error correction schemes when correcting for correlated errors, while still correcting single-qubit errors [7]. A quantum convolutional neural network is part of a more general class of algorithms that have become an active field of quantum computation research: variational quantum algorithms (VQAs). These variational quantum algorithms, in turn, are part of the larger concept of quantum differentiable programming. In essence, these algorithms involve a quantum-classical interface, in which a quantum circuit is constructed with unitary gates depending on parameters to be variationally optimized through a classical optimization procedure. These classes of variational algorithms have a wide range of applications, including finding ground states (variational quantum eigensolver), solving combinatorial optimization problems (quantum approximate optimization algorithm), in mathematical applications such as factoring, and in new frontiers of quantum information and quantum metrology, just to name a few [22].

Based on their wide range of applications, it is paramount to investigate variational quantum algorithms, especially those that bring us closer to large-scale, fault-tolerant quantum computing. In this work, I introduce the basic building blocks toward building an error-correcting scheme based on a variational approach and show that it is possible to achieve a reduction in logical error for a particular QCNN 3-qubit architecture, exemplifying the power of this variational approach. In Chapter 2 I introduce the concept of quantum noise and present the mathematical formalism to describe noise, including the operator-sum representation and the master equation approach. In Chapter 3 I introduce the concept of quantum error correction and present textbook

examples of error-correcting schemes. In Chapter 4 I delve into the machine learning concept of neural networks and bring it all together in Chapter 5, where I present the quantum convolutional neural network as an error correction scheme, first introduced in [7], and show how this model can be trained to correct arbitrary single-qubit errors in a 3-qubit code. Finally, in Chapter 6 I present the conclusions of this work and provide and outlook for the field of quantum error correction via quantum convolutional neural networks.

2 Quantum Noise

Open quantum systems are prone to unwanted interactions with the environment, giving rise to decoherence (noise) in quantum information processing. Therefore, to have reliable quantum information processing, it is necessary to understand the mechanisms behind noise and devise noise control methods. In order to understand quantum noise, it is imperative to find a way to describe the dynamics of open quantum systems. The *quantum operations* mathematical formalism is a helpful tool in describing such dynamics, as it addresses a wide range of physical scenarios. The quantum operations formalism can describe the dynamics of a system that is weakly coupled to the environment, a system that is strongly coupled to the environment, a system that is originally closed and suddenly opened to the environment, as well as systems with discrete state changes.

In this chapter, I review some classical noise concepts before moving on to the quantum operations formalism. The quantum operation formalism can be explained with three different approaches; each of these approaches turn out to be equivalent, albeit their differences. After building the basis to study quantum noise through the quantum operations formalism, I look at examples of quantum noise, including the bit flip, phase flip, depolarization, amplitude damping, and phase damping channels and how errors arise from each of these channels. In the end, I present an alternative approach to study open quantum system dynamics under noise, the master equation approach, and show an example of a two-level system under noise. This chapter follows Nielsen and Chuang's *Quantum Computation and Quantum Information* textbook closely [23], as it provides a very clear picture of both classical and quantum noise in a way that can be applied to various problems of interest, including quantum error correction, which will follow in a subsequent chapter.

2.1 Building the Basis for Quantum Noise

In order to understand quantum noise, it is important to understand noise in classical systems. A simple model to understand classical noise is a classical hardware used to store bits of information. In classical computing, a bit is either stored in binary as 0 or 1. If these bits are stored in a hard disk drive, as is most common, they may experience external magnetic fields that can cause a flip to occur, that is, a bit flipping from the state 0 to the state 1, or vice versa. This probability of flipping can be modeled as having a probability p that the bit will flip and a probability $1-p$ that the bit remains the same.

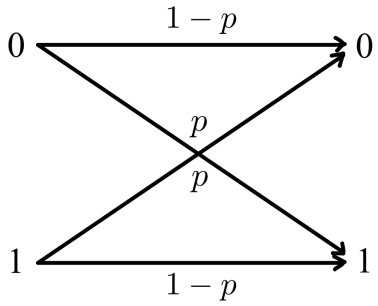

Figure 1: Schematic of classical bit flip. With probability p the bit flips, and it remains the same with probability $1 - p$.

In principle, the probability of a bit flip occurring is calculated by sampling the external magnetic fields in the environment in which the hard drive is operating (through Maxwell's equations) and model how the fields will affect the stored bits over time.

This prescription can be generalized to find the probability of an error occurring in various systems and environments. The issue here becomes finding an effective model for the environment, as well as the environment-system interactions. A high level of accuracy in this model, which describes a physical system, can be attained if the model is constructed conservatively and the system's observables are studied closely to determine whether they follow the model or not.

In the hard drive example above, the behavior of a bit can be described by:

$$p(Y = y) = \sum_x p(Y = y|X = x)p(X = x), \tag{1}$$

where X is the initial state of the bit, Y is the final state of the bit, and the conditional probabilities $p(Y = y|X = x)$ are the transition probabilities. This equation is known as the law of total probability. If the initial probabilities that the bit is in the state 0 or 1 are defined as p_0 and p_1, respectively, and q_0 and q_1 as the corresponding probabilities after the noise, equation 1 can be written explicitly as:

$$\begin{pmatrix} q_0 \\ p_0 \end{pmatrix} = \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix} \begin{pmatrix} p_0 \\ q_0 \end{pmatrix}. \tag{2}$$

Therefore, for a single stage process, the output probabilities $\vec{q}$ are related to the input probabilities $\vec{p}$ by:

$$\vec{q} = E\vec{p}, \tag{3}$$

where E is the transition probabilities matrix, also referred to as the evolution matrix. The evolution matrix E must be positive and complete. It is clear that a main feature of a classical noise process is that there is a linear relationship between input and output probabilities. This linearity will translate well when looking at quantum noise, since the dynamics of quantum systems behave in a linear manner.

Expanding to instances where noise happens at multiple stages in the process, the noise acts independently in each stage. Namely, this process becomes a stochastic Markovian process. Physically, it is safe to assume that noise acts independently: an environmental source of noise acting on one component of the system is different than a noise source acting on a different part of the system, therefore independent. If the noise acting on a system is not caused by independent environments, then the Markovicity rule does not hold.

In general, there are two categories of quantum noise: coherent noise and incoherent noise. For the case of coherent noise, it can be described by unitary operations that maintain the purity of the output quantum state. In this sense, coherent noise is less problematic than incoherent noise. Common sources of coherent noise are systematic errors originating from devices that are not perfectly calibrated; for example, a logical gate performing a rotation of $\phi + \epsilon$ rather than a rotation of ϕ. On the other hand, incoherent noise is more problematic, as it involves entanglement of qubits with the environment, resulting in mixed output states. A key difference between these types of noise is that incoherent noise results in random outputs, regardless

of the measurement basis (in the Examples of Quantum Noise and Quantum Operations subsection, I show how this is not the case for coherent noise).

2.2 Quantum Operations

Classical noise can be described as a stochastic Markov process involving classical states, which are described by vectors of probabilities. In a similar fashion, the dynamics of a quantum state can be described by employing the density operator ρ, and studying how this operator transforms under different mappings. In short, quantum states transform as:

$$\rho' = \mathcal{E}(\rho), \tag{4}$$

where $\mathcal{E}$ is a quantum operation mapping ρ to ρ'. A common example of a quantum operation is the unitary evolution of a quantum system. In the quantum operations formalism, this unitary evolution can be described as:

$$\mathcal{E}(\rho) = U\rho U^\dagger. \tag{5}$$

Since pure states evolve under unitary transforms as $|\psi\rangle \to U|\psi\rangle$, it is clear that, equivalently, $\rho \to \mathcal{E}(\rho) \equiv U\rho U^\dagger$ for $\rho = |\psi\rangle\langle\psi|$. Thus, the dynamics of closed quantum systems can be studied through the quantum operations formalism, as it is described by unitary transforms.

Unfortunately, closed quantum systems are practically impossible to achieve (except possibly if the whole universe is taken as a closed-off system). Nevertheless, the quantum operations formalism can still be applied to open quantum systems, if a few assumptions are made. Namely, taking both the system of interest and the environment as a closed system being subjected to a unitary transformation U. In this scenario, the final state of the system, $\mathcal{E}(\rho)$ might not be related to the initial state of the system ρ through a unitary transform. A further assumption is that the total input state, meaning the system-environment state, is a product state $\rho \otimes \rho_{\text{env}}$. After the transformation U the reduced density matrix of the system is obtained by performing a partial trace over the environment:

$$\mathcal{E}(\rho) = \text{tr}_{\text{env}}[U(\rho \otimes \rho_{\text{env}})U^\dagger]. \tag{6}$$

In general, the system and the environment do not start in a product state, although under certain experimental considerations, this assumption is accurate. For example, a quantum state can be prepared so that there are no correlations with the environment. Clearly, as the system evolves, the

system's and environment's degrees of freedom become correlated, and the system-environment density matrix is no longer a product state.

In reality, an environment has infinite degrees of freedom, yet, for this formalism, it is only necessary to model the environment up to a Hilbert space dimension of d^2 if the Hilbert space of the system is d.

This definition of quantum operations can be generalized to include different input and output spaces. For example, a qubit prepared in some unknown state ρ, labeled A, and a qutrit prepared in the standard state $|0\rangle$, labeled B, evolve through a unitary transformation U, and the joint system evolves to the state $U(\rho \otimes |0\rangle \langle 0|)U^\dagger$. If system A is traced out to leave system B in a final state ρ', then the quantum operation describing this process is:

$$\mathcal{E}(\rho) = \rho' = \mathrm{tr_A}(U(\rho \otimes |0\rangle \langle 0|)U^\dagger). \tag{7}$$

In this case, the quantum operation $\mathcal{E}$ maps density operators of the input system, A, to density operators of the output system, B. It is also possible to trace over subsystem B to obtain a quantum operation on subsystem A, that is, the quantum operation maps density operators of system B to density operators of system A.

2.3 Operator-Sum Representation

The operator-sum representation consists of a different approach to quantum operations, with the same results. In the previous section, the approach involved studying the interactions between systems and environments. This approach is helpful because it provides a clear physical picture, yet it lacks utility. On the other hand, the operator-sum representation provides a mathematical framework for quantum operations that is useful for calculations and theoretical work, yet it lacks the concreteness of the system-environment approach.

The operator-sum representation recasts equation 6 in terms of operators on the principal system's Hilbert space alone. If an orthonormal basis for a finite-dimensional state space of the environment is defined as $|e_k\rangle$, then the initial state of the environment can be described by $\rho_{\mathrm{env}} = |e_0\rangle \langle e_0|$. Plugging this expression into equation 6:

$$\begin{aligned}\mathcal{E}(\rho) &= \sum_k \langle e_k| U \big[\rho \otimes |e_0\rangle \langle e_0| \big] U^\dagger |e_k\rangle \\ &= \sum_k E_k \rho E_k^\dagger.\end{aligned} \tag{8}$$

This equation corresponds to the operator-sum representation of $\mathcal{E}$, where $E_k \equiv \langle e_k| U |e_0\rangle$ are operators, commonly referred to as Kraus operators, that act on the state space of the principal system. As was the case for the evolution matrix, the operator elements $\{E_k\}$ must satisfy the completeness relation. For the case of the classical evolution matrix, this required that the probability distributions be normalized to one. For the operator-sum elements, the completeness relation means that the trace of $\mathcal{E}(\rho)$ be equal to one. Namely,

$$\sum_k E_k^\dagger E_k = I. \tag{9}$$

The equation above holds for operations that are trace-preserving. This definition is equivalent to equation 6. There are non-trace-preserving quantum operations for which $\sum_k E_k^\dagger E_k \leq I$, in which extra information about what occurred in the process is obtained via measurement.

From the definition above, the operator-sum representation offers an intrinsic way of studying system dynamics, since properties of the environment don't have to be considered explicitly: everything is contained within the operator elements E_k. This representation of quantum operations greatly simplifies calculations and allows for theoretical insight. Lastly, it is worth noting that if there is only interest in the dynamics of the principal system, then a representation that ignores unimportant information about other systems can be picked, as many different environmental interactions may cause similar dynamics.

In the next section, I present a few examples of quantum noise and quantum operations. Specifically, I show the trace and partial trace within the quantum operations formalism, as well as the error channels used to construct the error-correction model to follow.

2.4 Examples of Quantum Noise and Quantum Operations

In order to illustrate the power of the quantum operations formalism, I begin this section by presenting concrete examples of quantum operations and quantum noise. It is important to understand, under this formalism, the practical effect of quantum noise on systems, as it will lead to a better understanding of the techniques used for error correction.

First, I introduce the most common sources of error (and error channels) in quantum systems, including the bit flip, phase flip, amplitude damping, generalized amplitude damping, phase damping, and depolarizing channels. With a simple example of a Bell state construction, I show how different sources of error affect our quantum computation.

I follow by introducing the trace and partial trace as quantum operations, and finish this section by providing a concrete application of quantum operations. Namely, the master equation approach to quantum dynamics, as it is a powerful tool to describe continuous-time dynamics of open quantum systems.

2.4.1 Error Channels

Within the operator-sum representation (Kraus operators), common sources of error can be described, along with their associated probabilities. A common error in quantum computing is the bit flip error. This type of error can be modeled by a bit flip channel: a transformation that flips the state of a qubit (ie. by applying a Pauli X gate) with probability p, and leaves it unchanged (ie. by applying the Identity gate) with probability $1-p$ (For a list of common quantum gates, see [23]). In the Kraus operator formalism:

$$E_0 = \sqrt{1-p}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$E_1 = \sqrt{p}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Similarly, the Kraus operator representation of a phase flip channel—a transformation that changes the phase of a qubit (by applying a Pauli Z gate) with

probability p and leaves it unchanged otherwise—is given by:

$$E_0 = \sqrt{1-p}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$E_1 = \sqrt{p}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

In both of these cases, the only "error" parameter is the probability of error p, which is not always the case.

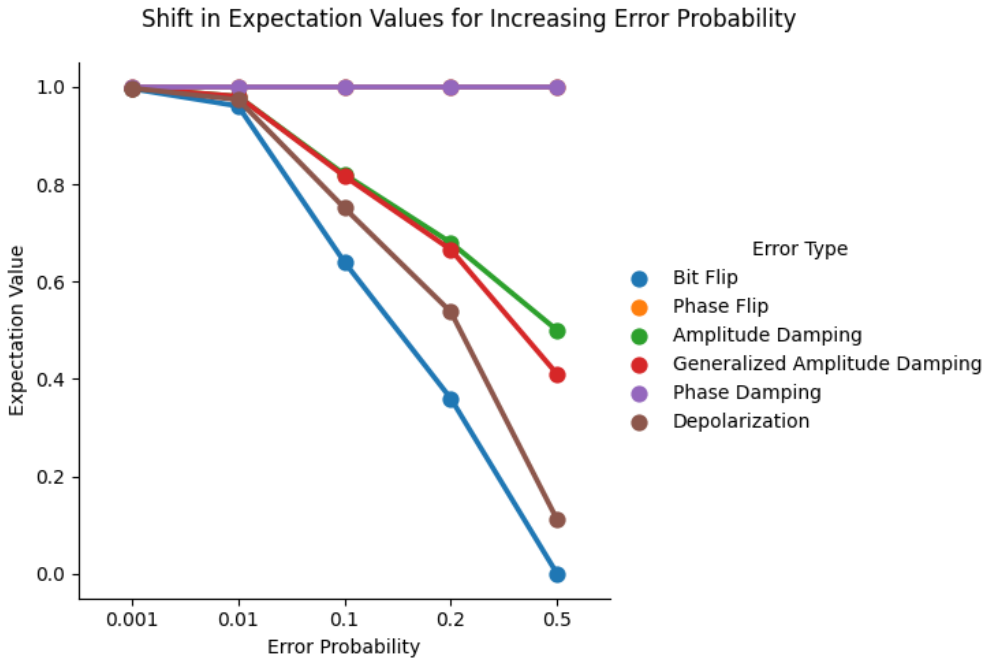

Figure 2: Variation of expectation value (measured in the computational basis) in the construction of a Bell state for increasing probability of error, shown for various error channels. The change in expectation value is measurement-basis dependent, which is why there is no change in expectation value for the phase flip and phase damping channels (overlapped at 1.0). This figure was produced using PennyLane's noisy circuit simulator, with each of the corresponding noise probabilities injected into the circuit.

Figure 2 presents a schematic of the effect of an increasing probability of error when preparing a Bell state. A Bell state is defined as:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{12}$$

which corresponds to a maximally entangled state. The Bell state is ubiquitous in quantum computation, since it is the basis for many quantum algorithms, such as the quantum teleportation algorithm. In an ideal, error-free implementation of the Bell state construction, a measurement of the expectation value of both qubits in the computational basis should unequivocally yield 1.

Figure 2 shows the expectation value of making a measurement in the computational basis (Z-basis) on both qubits. That is, the expectation value represents the operation $Z_0 \otimes Z_1$ on the Bell state. In other words,

$$\langle\psi| Z_0 Z_1 |\psi\rangle = \frac{1}{2}(\langle 00| + \langle 11|)Z_0 Z_1(|00\rangle + |11\rangle). \tag{13}$$

For this particular case, the expectation value is 1, as already mentioned, since no error has a occurred; the Bell state is as eigenstate of the $Z_0 Z_1$ operator.

The fact that the measurement is performed in the computational basis is reflected in the fact that for the case of the phase flip and phase damping channels there is no reduction in the expectation value. Since the Bell state is an eigenstate of the Pauli Z matrix, the expectation value does not change. If, instead, the measurement were performed in the X-basis, there would be a decrease in the expectation value for these error channels and the expectation value for the bit flip channel would remain unchanged.

As mentioned previously, most of these channels only involve the parameter p, representing the probability of an error occurring. This is not the case for the generalized amplitude damping channel. The Kraus operators for this error type are:

$$E_0 = \sqrt{p} \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix},$$

$$E_1 = \sqrt{p} \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix},$$

$$E_2 = \sqrt{1-p} \begin{pmatrix} \sqrt{1-\gamma} & 0 \\ 0 & 1 \end{pmatrix},$$

$$E_3 = \sqrt{1-p} \begin{pmatrix} 0 & 0 \\ \sqrt{\gamma} & 0 \end{pmatrix}.$$

Here, the parameter $\gamma \in [0,1]$ is the probability of damping, and $p \in [0,1]$ is the probability of the system being excited by the environment. [2] This error channel is useful when modeling the exchange of energy between a qubit and its environment at finite temperature. In Figure 2, $p = 0.1$; not to be confused with the probability of a bit flip or a phase flip occurring, as is the case for the bit flip and phase flip error channels.

2.4.2 Trace and Partial Trace

It is well known that measurements performed on quantum systems lead to a permanent change in the system. The quantum operations formalism can be employed to describe the effects of a measurement on a quantum system. On the one hand, the outcome of a measurement on a system can be determined, with its respective probability, and on the other hand the change that took place in the system as an effect of that measurement can be determined.

A simple quantity related to a measurement is the trace map $\rho \to \mathrm{tr}(\rho)$. This mapping can be cast through the quantum operations formalism as follows: Let H_Q be any input Hilbert space spanned by an orthonormal basis $|1\rangle ... |d\rangle$, and H'_Q be a one-dimensional output space spanned by the state $|0\rangle$. This quantum operation is defined as:

$$\mathcal{E}(\rho) = \sum_{i=1}^{d} |0\rangle \langle i| \rho |i\rangle \langle 0| . \tag{15}$$

Since $|0\rangle \langle 0|$ is an unimportant multiplier, $\mathcal{E}(\rho) = \mathrm{tr}(\rho) |0\rangle \langle 0|$ is the trace function.

Now, to trace out a system R from a joint system QR, a basis $|j\rangle$ for system R and a linear operator $E_i : H_{QR} \to H_Q$ can be defined as:

[2] Notice that in previous error channels, the probability of an error occurring is denoted by p. In the case for the amplitude damping channel, this p takes a different meaning, as explained in the text

$$E_i\left(\sum_j \lambda_j \left|q_j\right\rangle \left|j\right\rangle\right) \equiv \lambda_i \left|q_i\right\rangle, \tag{16}$$

where λ_j are complex coefficients, and $|q_j\rangle$ are arbitrary states of the system Q. Here E_i is an operator that forms a basis of the form (8). Therefore, the quantum operation $\mathcal{E}$ becomes:

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger. \tag{17}$$

This operation takes the system from QR to Q, by tracing out subsystem R. If the operation $\mathcal{E}$ is applied:

$$\mathcal{E}(\rho \otimes \left|j\right\rangle \left\langle j'\right|) = \rho\delta_{j,j'} = \mathrm{tr}_R(\rho \otimes \left|j\right\rangle \left\langle j'\right|), \tag{18}$$

it is clear that, by linearity, $\mathcal{E} = \mathrm{tr}_R$. Thus, the trace and partial trace can be defined within the operator-sum formalism.

2.4.3 Master Equation Approach

Research of open quantum systems is currently an active area of research, and one that gives great insight into quantum error and quantum error correction. As such, it important to have a wide range of tools to study open quantum systems, such as the master equation approach.

In the master equation approach, the dynamics, or time evolution, of an open quantum system is described with a differential equation that properly describes non-unitary behavior. The most general form of a master equation comes in the form of the Lindblad equation:

$$\frac{\partial \rho}{\partial t} = -\frac{i}{\hbar}[H, \rho] + \sum_j [2L_j \rho L_j^\dagger - \{L_j^\dagger L_j, \rho\}], \tag{19}$$

where $L_j, L_j^\dagger$ are the Lindblad operators describing the coupling of the system to the environment, and $[\cdot,\cdot]$ and $\{\cdot,\cdot\}$ represent the usual commutator and anti-commutator, respectively. In Equation 19, the Hamiltonian is a Hermitian operator that described the coherent dynamics of the system, and at all times $\mathrm{tr}[\rho(t)] = 1$. For this approach to work, the system and environment are assumed to start out as a product state. Additionally, in order to determine the Lindblad operators L_j, the system + environment Hamiltonian is simplified by first taking the Born approximation (weak coupling between the system and the environment) and then the Markov approximation (short

correlation times).

As an example, a two-level atom coupled to the vacuum that undergoes spontaneous emission can be described with the master equation approach. For this system, the Hamiltonian describing the coherent evolution is given by $H = -\omega S_z = -\frac{\hbar\omega}{2}\sigma_z$, where S_z is the spin operator, and σ_z is the Pauli Z operator. The atom undergoes spontaneous emission by going from state $|1\rangle$ to state $|0\rangle$ and emitting a photon in the process. The Lindblad operator for this process is given by $\sqrt{\gamma}\sigma_-$, where γ is the rate of spontaneous emission and $\sigma_- \equiv |0\rangle\langle 1|$, in the second quantization picture, correspond to the atomic lowering operator. Similarly, σ_+ is the raising operator. Substituting this Lindblad operator into the master equation:

$$\frac{\partial\rho}{\partial t} = -\frac{i}{\hbar}[H,\rho] + \gamma[2\sigma_-\rho\sigma_+ - \sigma_+\sigma_-\rho - \rho\sigma_+\sigma_-]. \tag{20}$$

Moving into the interaction picture [24] with the change of variables:

$$\tilde{\rho}(t) \equiv e^{iHt}\rho(t)e^{-iHt}, \tag{21}$$

Equation 20 becomes:

$$\frac{\partial\tilde{\rho}}{\partial t} = \gamma[2\tilde{\sigma}_-\tilde{\rho}\tilde{\sigma}_+ - \tilde{\sigma}_+\tilde{\sigma}_-\tilde{\rho} - \tilde{\sigma}\tilde{\sigma}_+\tilde{\sigma}_-],^{3} \tag{22}$$

where

$$\tilde{\sigma}_- \equiv e^{iHt}\sigma_- e^{-iHt} = e^{-i\omega t}\sigma_-$$
$$\tilde{\sigma}_+ \equiv e^{iHt}\sigma_+ e^{-iHt} = e^{i\omega t}\sigma_+.$$

Substituting these into Equation 22, the final equation of motion becomes:

$$\frac{\partial\tilde{\rho}}{\partial t} = \gamma[2\sigma_-\tilde{\rho}\sigma_+ - \sigma_+\sigma_-\tilde{\rho} - \tilde{\rho}\sigma_+\sigma_-]. \tag{24}$$

This equation of motion can be solved by using the Bloch vector representation for $\tilde{\rho}$ [23], which yield the solution:

$$\lambda_x = \lambda_x(0)e^{-\gamma t}$$
$$\lambda_y = \lambda_y(0)e^{-\gamma t}$$
$$\lambda_z = \lambda_z(0)e^{-2\gamma t} + 1 - e^{-2\gamma t}.$$

[3] In the interaction picture, the commutator of the density matrix and the Hamiltonian is zero since we jump into a "rotating frame".

With a further substitution of the variable $\gamma^{'} = 1 - e^{-2\gamma t}$ and going back to the density matrix representation:

$$\tilde{\rho}(t) = \mathcal{E}(\tilde{\rho}(0)) \equiv E_0 \tilde{\rho}(0) E_0^{\dagger} + E_1 \tilde{\rho}(0) E_1^{\dagger}, \tag{26}$$

where

$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma^{'}} \end{pmatrix},$$

$$E_1 = \begin{pmatrix} 0 & \sqrt{\gamma^{'}} \\ 0 & 0 \end{pmatrix}.$$

These Kraus operators are equivalent to the amplitude damping operators. In this example, the quantum system was small and the bath was modeled as a collection of simple harmonic oscillators so, while this example shows the power of the master equation approach, it also reveals some of its drawbacks. The master equation approach is less general than the quantum operations formalism: a quantum process that is described in the operator-sum representation cannot necessarily be described by a master equation, such as in the case for non-Markovian dynamics.

Understanding quantum noise and the processes behind it is vital in order to determine novel and effective ways to correct for these errors. In the following chapter, I delve into the field of quantum error correction, now having the tools to describe quantum noise and quantum system's behaviors under external interactions.

3 Quantum Error Correction

The theory of quantum error-correction codes was established more than two decades ago as the primary tool for fighting decoherence in quantum computers and quantum communication systems. Quantum systems, such as qubits, are susceptible to a variety of errors that can perturb or even destroy their quantum state (quantum information content) [25]. This loss of information is caused by entanglement of the system to either the environment's degrees of freedom or other subsystems' degrees of freedom—also known as decoherence. Fortunately, there are ways to counteract errors caused by decoherence, through the clever use of quantum features. That is, quantum errors caused by entanglement can be countered with entanglement: entanglement is both a source of error and a powerful tool in the quantum error correction toolbox [26]. Entanglement is an important tool since it enables teleportation of quantum states without physically sending quantum systems [27], it doubles the capacity of quantum channels for sending classical information, and it helps counteract quantum errors [23].

Many efforts are currently being undertaken to reduce quantum noise levels in physical implementations of quantum computers; and while QEC has been well understood theoretically, it has just begun its experimental trajectory. The first experimental demonstration of quantum error correction was done using three beryllium atomic-ion qubits confined to a linear, multizone trap [28]. Unfortunately, experimental implementations of quantum error correction require scalability, especially moving towards the large-scale quantum computing era. Experimental efforts continue in order to achieve highly accurate quantum computation and quantum information processing.

One of the most striking developments in the field of quantum error correction was the introduction of the stabilizer formalism, whereby quantum codes are code spaces in Hilbert space and are specified by giving the generators of an Abelian subgroup of the Pauli group, called the stabilizer of the code space [23,29,30]. To date, most quantum error-correcting codes are stabilizer codes. In simple terms, a stabilizer code protects quantum information by encoding logical qubits via entanglement with ancilla qubits. This idea stems directly from classical error-correction repetition codes, where redundancy is created in order to protect information. By appending ancillae, a stabilizer code can restore a noisy, decohered quantum state to a pure quantum state. By successfully encoding quantum information, local sources of error can be detected and corrected via syndrome measurements of ancilla qubits. Many quantum error-correcting schemes are based on the idea of reverting a deco-

hered state to a pure state, for example, via a reversal operation, which takes into account both noise and initial density operator to achieve near-optimal preservation of the initial density operator's quantum entanglement or classical correlation with a reference system [31].

3.1 Constructing Error-Correcting Codes

Many current quantum error-correction schemes are able to determine single-qubit errors and correct for them. These errors include bit-flip, phase-flip, damping errors, or combinations of multiple errors [23]. Quantum error is realized with a noise channel as two independent events: an error is introduced to the qubit with a certain probability p, and with probability $(1-p)$ the qubit remains the same (Fig. 1).

3.1.1 The 3-Qubit Code

The simplest implementation of a quantum error correcting scheme, the three-qubit code, stems from the classical error-correction concept of a repetition code, where redundancy is used as a tool to protect information against noise [9]. By creating redundancy, the total size of the Hilbert space is expanded, so that errors on individual qubits are mapped to a large set of mutually orthogonal subspaces. The three-qubit code is able to detect and correct an arbitrary bit-flip error. In the three-qubit code, the target logical qubit is protected as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \rightarrow \alpha\,|000\rangle + \beta\,|111\rangle\,. \tag{28}$$

To detect and correct a bit flip, a series of projective measurements are performed, so separate auxiliary qubits (ancillae) are necessary for syndrome measurements, storage and correction. The three-qubit code involves a series of controlled-NOT (CNOT) gates between encoded qubits and ancilla qubits, followed by measurements on ancilla qubits, as shown in Figure 3. $|\psi\rangle$ represents the logical qubit, and the $|0\rangle$ qubits represent the ancillae. The CNOT gate acts by flipping the target qubit (black dot) if the control qubit (crossed circle) is in the $|1\rangle$ states. If the control qubit is in the $|0\rangle$ state, the target qubit remains unchanged. As opposed to a classical circuit, the horizontal lines stemming from the qubits do not necessarily represent an actual circuit, but can represent, for example, the passing of time.
If two unitary operators are defined as $\hat{U}_1 = \mathrm{CNOT}_{q1,a1}\mathrm{CNOT}_{q2,a1}$ and $\hat{U}_2 = \mathrm{CNOT}_{q2,a2}\mathrm{CNOT}_{q3,a2}$ (here, q refer to encoded qubits and a refer to ancilla qubits), it is possible to calculate $\hat{U}_2\hat{U}_1(\alpha\,|000\rangle + \beta\,|111\rangle)\,|00\rangle$ and characterize the effect of bit flips via the ancilla syndrome measurement. There are four possible outcomes:

$$\hat{U}_2\hat{U}_1(\alpha\,|000\rangle + \beta\,|111\rangle)\,|00\rangle = (\alpha\,|000\rangle + \beta\,|111\rangle)\,|00\rangle\,, \tag{29a}$$

$$\hat{U}_2\hat{U}_1(\alpha\,|100\rangle + \beta\,|011\rangle)\,|00\rangle = (\alpha\,|100\rangle + \beta\,|011\rangle)\,|10\rangle\,, \tag{29b}$$

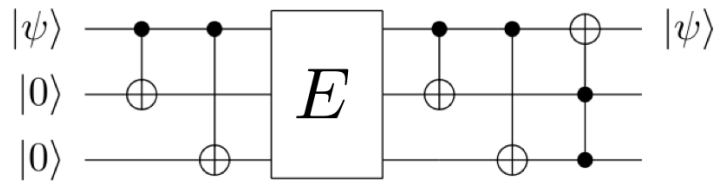

Figure 3: Representation of 3-qubit error-correcting scheme. Ancilla qubits are (usually) prepared in the $|0\rangle$ state. Through a series of quantum control gates, bit flip errors in $|\psi\rangle$ can be detected and corrected. The bit-flip channel E_{bit} is modelled as a Pauli-X gate.

$$\hat{U}_2\hat{U}_1(\alpha\,|010\rangle + \beta\,|101\rangle)\,|00\rangle = (\alpha\,|010\rangle + \beta\,|101\rangle)\,|11\rangle\,, \tag{29c}$$

$$\hat{U}_2\hat{U}_1(\alpha\,|001\rangle + \beta\,|110\rangle)\,|00\rangle = (\alpha\,|001\rangle + \beta\,|110\rangle)\,|01\rangle\,. \tag{29d}$$

Measurement of the ancilla qubits determines if a bit flip error occurred, and if it did, in which qubit. For example, if the syndrome measurement is 11, a bit flip occurred on the second qubit. Similarly, if the syndrome measurement is 00, no error occurred, and so on. After the error is detected, there are a few ways to correct it. One way involves feed-forward application of Pauli-X gates via fast electronics to flip the faulty qubit. A second way of correcting bit flip errors is applying Toffoli gates with suitable truth tables, so that error correction becomes "automatic". The Toffoli gate, also referred to as a CCNOT (controlled-controlled-not), is a 3-qubit gate that takes the input of two qubits to determine whether a third qubit is flipped or not. Using Toffoli gates with suitable truth tables for error correction involves choosing which two of the three qubits act as controls, and which qubit acts as the target. Finally, errors can be stored in classical memory and corrected at the end. Implementation of this coding scheme has been done experimentally in ion traps [13, 28, 32] and transmon circuits [12, 33].

Similarly, a three-qubit phase flip code is able to detect and correct for arbitrary phase flips on qubits. The effect of a phase flip on the computational basis is $\alpha\,|0\rangle + \beta\,|1\rangle \rightarrow \alpha\,|0\rangle - \beta\,|1\rangle$. The phase flip becomes analogous to the bit flip when looking at the effect on the $|\pm\rangle$ basis:

$$|+\rangle = \frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \longrightarrow |-\rangle\,, \tag{30a}$$

$$|-\rangle = \frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \longrightarrow |+\rangle\,. \tag{30b}$$

Therefore, the logical qubit is encoded as $|0_L\rangle = |+++\rangle$ and $|1_L\rangle = |---\rangle$. As for the bit flip code, the phase flip code performs the same projective measures, only conjugated by Hadamard gates, that is, $H^{\otimes 3}\hat{U}_1\hat{U}_2H^{\otimes 3}$. The phase flip error is fixed by Hadamard conjugated recovery [23].

3.1.2 The 9-Qubit Code

Building on the three-qubit bit flip and phase flip codes, the 9-qubit Shor code extends to scope of the error correcting scheme to account for arbitrary phase flip *and* bit flip errors [9]. The Shor code for error correction starts by encoding the qubit using the phase flip code (i.e. $|0\rangle = |+++\rangle$). Next, each of these qubits is encoded using the bit flip code, for example, $|+\rangle$ is encoded as $(|000\rangle + |111\rangle)/\sqrt{2}$. This results in a 9-qubit code, with codewords[4] given by:

$$|0\rangle \longrightarrow |0_L\rangle \equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}, \tag{31a}$$

$$|1\rangle \longrightarrow |1_L\rangle \equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}. \tag{31b}$$

A general scheme for the 9-qubit Shor code is presented in Figure 4. Again, by a series of projective measurements, this code is able to correct for arbitrary errors on any of the qubits, as long as the error happens on a single qubit. Since the mapping between correctable errors and unique states is not unique, the Shor code is a degenerate code. Additionally, since the 9-qubit code can correct for a single X-error in any one block of three and single Z-error on any of the nine qubits, it is a *full* quantum error correcting code. However, the 9-qubit code is only a single error correcting code, and cannot fix multiple errors occurring in different locations.

Full quantum error correcting codes, such as the 9-qubit Shor code, perform well under the assumption that noise acts independently on each qubit, as is assumed in classical error correction, and that noise is not too intense. In the quantum regime, this assumption no longer holds true, as correlated errors are prone to occur and can vary in intensity. In the realm of correlated quantum errors, the scope can vary from two-qubit correlations [19] to fully-correlated noise channels [20]. Treatment of such errors is crucial towards achieving large-scale quantum computing. Since quantum error correction

[4] A codeword is defined as the string of bits—or in this case, qubits–that is used to define a symbol—in this case, the logical qubit.

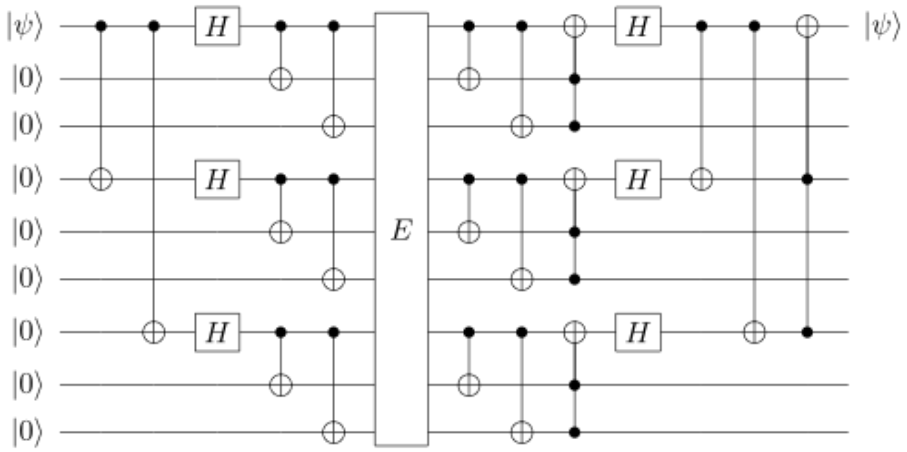

Figure 4: Representation of 9-qubit Shor scheme. The logical qubit $|\psi\rangle$ is encoded in 8 other ancilla qubits through a series of CNOT and Hadamard gates. The initial CNOT gates encode for bit-flip errors while the Hadamards gates with the subsequent CNOT gates encode for phase-flip errors.

can become a computationally expensive and experimentally difficult task to undertake (for the 9-qubit Shor code, 9 physical qubits are necessary to encode a single qubit), it is necessary to develop novel techniques to address what it would otherwise be an intractable problem.

3.2 Error Correction vs. Error Mitigation

As seen in the previous two sections, quantum error-correcting codes fully correct arbitrary errors on a single qubit. In practice, and within the context of the current NISQ quantum devices, full error correction is not experimentally realizable. Instead, current implementations try to *minimize* quantum error. This is referred to as error mitigation [34, 35]. With error mitigation protocols, it is possible to achieve a good level of fidelity for some classes of circuits; especially for short-depth circuits [36–38]. Until large scale, fault-tolerant quantum computation with error correction becomes achievable, error mitigation can help harness the power of quantum computers in the NISQ era.

4 Neural Networks

In recent years, machine learning has resurfaced as a practical and effective way to solve a variety of problems, with applications ranging from behavioral psychology to economics and finance, and even medical disciplines. Machine learning algorithms allow computers to perform specific tasks effectively without being explicitly programmed to do so. In a sense, a computer "learns" to perform a task, such as classifying input from a training data set. This class of algorithms reduce the problem to an optimization problem. That is, these algorithms try to minimize the difference between the "calculated" output, to the actual (i.e. known) value of a training set; the cost function.

Within the realm of machine learning algorithms, there are classes of algorithms, which are tailored to different types of problems. Some of these algorithms include nearest neighbor, naive Bayes, decision trees, linear regression, support vector machines (SVMs), and neural networks [39]. Furthermore, each architecture can contain sub-classes of algorithms. For example, within neural networks, there are recurrent, feed-forward, and convolutional neural networks. In this chapter, I focus on the latter types of machine learning frameworks and the types of problems that can be solved using these.

I begin this chapter with an introduction to classical neural networks and the inspiration behind their architecture. The mathematical construction of neural networks follows, as well as the types of problems one can study with neural networks. In particular, I will focus on a particular type of neural network: the convolutional neural network, and highlight some of its most important features. Finally, I will bridge classical and quantum by expanding the idea of a convolutional neural network in terms of a quantum architecture, which will serve as a basis for the following chapter.

4.1 The Brain as a Model for Neural Networks

The human brain, one of the most complex biological system known to science, is composed of a convoluted network of interconnected biological neurons. These biological neurons receive information, in the form of an electrical impulse, through their dendrites and, based on this input, they produce an output to another neuron or neurons via their axon. A schematic of a biological neuron is shown below.

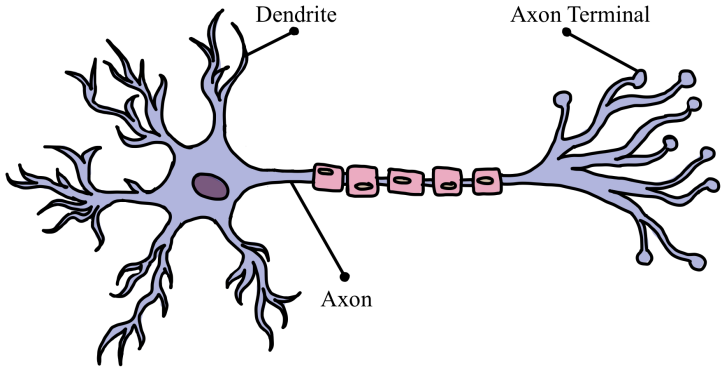

Figure 5: Schematic of a biological neuron. The dendrites receive a signal, which is processed and output to other neurons through the axon terminal.

Every signal in the nervous system is transferred through a network of these neurons, producing a particular output, which can include muscle movement, memory retrieval, pattern recognition, etc. The term "neural network" was derived from the work by neuroscientist Warren S. McCulloch and logician Walter Pitts, who were the first to develop the first conceptual model of an "artificial neural network" [40]. This became the basis for the neural network architecture ubiquitous in many areas of science and technology.

4.2 From Biological to Artificial

Based on the machine learning paradigm, and with the aid of studies in neuroscience and the brain, the concept of a neural network emerged as a powerful learning tool [41]. As previously mentioned, neural networks are models of interconnected units based on biological neurons feeding signals into one another, as depicted in Figure 6. The transmission of signal between neurons can be modeled as nodes having two states: "active" when signal is being fed into the neuron, and "resting" when there is no signal

going through the neuron.

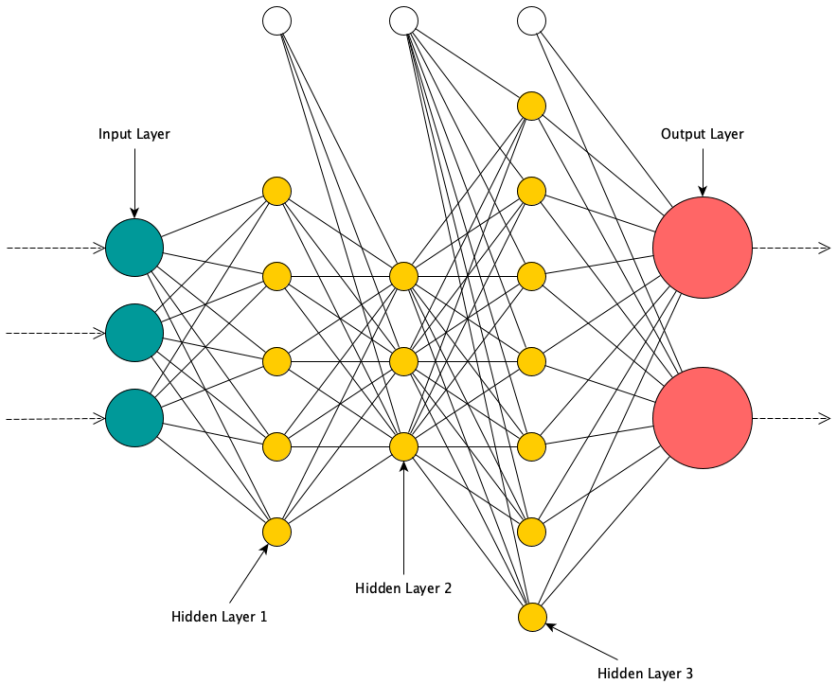

Figure 6: Representation of a neural network. The network consists of three main sections: the input layer (green nodes), the hidden layers (yellow nodes), and the output layer (red nodes). The number of nodes at each layer can vary, as well as the number of layers in each hidden layer. The white nodes represent bias nodes.

A neural network sends signals between connected neurons (nodes), each of which represent a real number. A neuron receives a signal, which it then processes, and computes an output with some non-linear function—usually referred to as an activation function—given the sum of its inputs. The learning process is carried out by adjusting the weights of each connection, therefore increasing or decreasing the strength of a particular signal, for a particular input/output. The number of neurons in each layer, as well the the number of hidden layers and the learning rate are collectively referred as the hyperparameters, which vary depending on the learning task at hand.

An active area of research involves the optimization of hyperparameters, since this represents a major challenge in designing neural networks. Typically, hyperparameters are chosen heuristically and manually fine-tuned, which can

be very time consuming [42]. Not only is choosing the optimal hyperparameters time consuming; increasing the number of nodes and layers in a neural network often comes with longer run times and the need for more computational resources. Some cases arise in which choosing a large number of nodes and/or layers causes the neural network to over-fit the data, rendering it useless in classification of new data, ie. not training data properly.

4.3 Mathematical Formulation of Neural Networks

In this section I introduce the mathematical formalism used to describe neural networks. Starting from a single neuron, a collection of neurons (layer), the weights connecting each layer, and the activation function, a fully-connected neural network for a given input/output can be formally described.

4.3.1 Single Neuron

The smallest element of a neural network is the neuron, or node. Each neuron receives as input a set of x-values, $x = \{1, 2, ..., n\}$, and predicts an output y. The data set $\boldsymbol{x}$ contains the features of one of m training set examples. As seen in Figure 6, each unit contains an associated weight vector $\boldsymbol{w}$ and a bias node b (white), which changes during the learning process. In each iteration, the neuron calculates a weighted average of the values of vector $\boldsymbol{x}$,

$$z = w_1 x_1 + w_2 x_2 + ... + w_n x_n + b = \boldsymbol{w}^T \cdot \boldsymbol{x} + b. \tag{32}$$

Finally, the result of this calculation is passed through a non-linear activation function $g(z)$. Figure 7 shows the process a single neuron undergoes in a neural network.
The activation function is one of the key elements of a neural network. Without the non-linear activation function, the neural network would just consist of a combination of linear functions, so it would just be a linear function itself. If this were the case, the neural network would be limited to solve only linear regression type problems. With the addition of a non-linear activation function, the neural network is able to explore more complex optimization landscapes. Some of the most common activation functions in use today are the sigmoid, tanh, ReLU, and leaky ReLU functions [43].

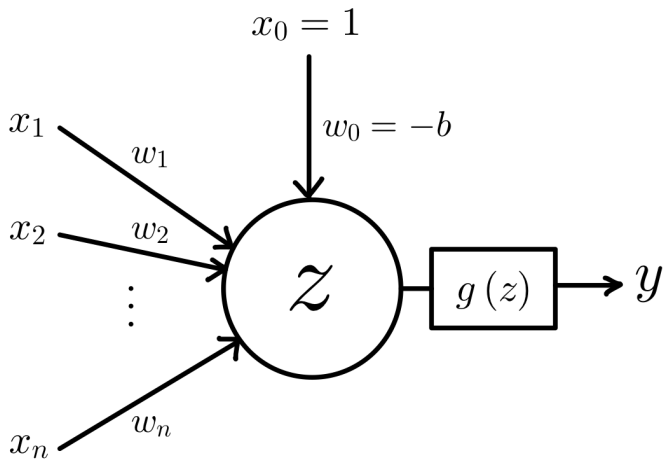

Figure 7: Schematic of an artificial neuron. Each input x has a weight w associated with it. The result of all of these inputs is passed through a non-linear activation function to obtain the output y. There is a bias input x_0, which introduces a threshold for the activation function.

4.3.2 Single Layer

Now that the mathematical basis for a single neuron has been established, it is possible to expand this idea to a layer containing multiple neurons. This is done by vectorizing across a full layer l to obtain the same equations for a single neuron with index i:

$$z_i^l = \boldsymbol{w}_i^T \cdot \boldsymbol{a}^{(l-1)} + b_i, \tag{33}$$

where

$$a_i^l = g^l(z_i^l). \tag{34}$$

Here, $\boldsymbol{x}$, which is the activation vector for the input layer, is replaced with $\boldsymbol{a}$, which denotes a general activation vector for a given layer.

It is clear that for each layer a number of very similar operations is performed. Therefore, a matrix $\boldsymbol{W}$ of dimension (n^l, n^{l-1}) can be built from the transpose of the vectors $\boldsymbol{w}$, and a bias vector $\boldsymbol{b}$ of dimension $(n^l, 1)$.

By applying multiple of these layers the neural network is constructed, as in Figure 6. The learning task becomes finding the appropriate weights $\boldsymbol{w}$ that

minimize the cost function for a particular training set.

4.4 Convolutional Neural Networks

Convolutional neural networks (CNNs) have become ubiquitous in classification tasks, such as image recognition [44]. A CNN consists of a sequence of convolution and pooling layers, followed by a fully connected layer. The convolution and pooling layers are image processing layers that, upon each application of a convolution-pooling layer, abstract and reduce the dimensionality (i.e. pixels) of the image. In each layer of image processing a new feature map (a 2D array of pixels) is generated from the previous layer.

The convolutional layer computes new pixel values $x_{i,j}^{(l)}$ from a linear combination of nearby ones in the preceding map, such that $x_{i,j}^{(l)} = \sum_{a,b=1}^{\omega} \omega_{a,b} x_{i+a,j+b}^{(l-1)}$. The weights $\omega_{a,b}$ form a $\omega \times \omega$ kernel. The pooling layer reduces the size of the feature map by taking the maximum value from a small neighboring subset of pixels, frequently followed by application of an activation function. Finally, a fully connected layer is applied to all the remaining pixels once the feature map has become sufficiently small. A schematic of a typical convolutional neural network is shown in Figure 8.

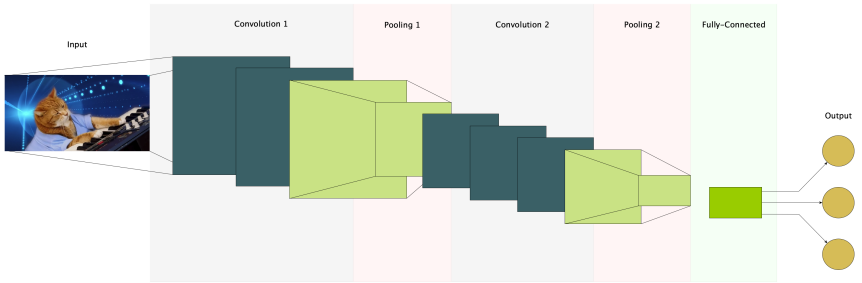

Figure 8: Representation of a CNN architecture. The input (image) goes trough convolution and pooling layers, followed by a fully-connected layer. The CNN then classifies the input into one of the previously trained classifiers.

The power of convolutional neural networks comes from the reduction in the dimensionality of the problem through the extraction of key features.

The range of applications of convolutional neural networks is wide: from image recognition to natural language processing, and even in drug discovery

and classic board games, such as Checkers and Go. Application of neural network algorithms in the study of quantum many-body systems and open quantum systems has been recently demonstrated [6,45–47]. These problems reduce to training a neural network to perform some classification task on a quantum system. For example, one can train a neural network to generate the ground state of a Bose-Einstein condensate [4].

Even though this implementation of neural networks has provided good results in condensed matter physics, there is even more potential in the implementation of a fully *quantum* neural network. That is, a neural network that is constructed from quantum principles, and is able to perform quantum tasks on quantum systems. The intuition behind the construction of a quantum neural network to study quantum systems stems from the same intuition for using quantum computation to study quantum systems, or, as Richard Feynman conjectured: a quantum computer which uses quantum mechanics intrinsically might be more powerful than a computer mired in the classical world [48].

Not only does a quantum neural network provide a speed-up over a classical neural network, but it does so while inherently possessing a quantum nature. In the following section I show the application of convolutional neural networks to quantum error correction and provide concrete examples of error mitigation instances.

5 Variational Quantum Algorithms

One of the main challenges of many-body physics is the exponential growth of Hilbert space dimension with the system's size. Even though modern computing affords incredibly powerful machines, the analytical study of quantum systems is still intractable. At present, many numerical techniques and simplifications are made in order to study quantum many-body systems. Additionally, physical states frequently have some internal structures, simplifying the general problem further [49]. Even with these simplifications, studies of many-body quantum systems are still a computationally expensive task.

A particular area that benefits from machine learning algorithms is quantum error correction. Through the clever use of classical machine learning algorithms, such as neural networks, in conjunction with a quantum information processing device, i.e. a quantum computer, hybrid quantum-classical algorithms can be constructed. These algorithms are able to "learn" error correction for specific noise models.

In this chapter, I starts by introducing variational quantum algorithms. In the VQA section, I review two particular algorithms that have proven effective in quantum error correction: the quantum variational error corrector (QVECTOR) and the quantum convolutional neural network (QCNN) [7,50]. Following this, I show instances of quantum error correction simulations using the QCNN architecture.

5.1 Quantum Variational Error Correction Algorithm

As mentioned previously, quantum error-correcting codes often require a large overhead in physical qubits. This is not the case only for Shor's error code; it is also the case for color code and surface code correction schemes. Estimates for the overhead of surface codes is even more egregious than for the Shor code, with estimates ranging in the 10^3–10^4 physical qubits required to protect a single logical qubit [51].

Reference [50] is one of the first proposals of a variational approach to quantum error correction. The main objective of the quantum variational error correction algorithm is to generate an encoding and a recovery circuit that is able to sufficiently mitigate error in a quantum memory.

This approach to quantum error correction is based on the general mathematical formalism of subspace code quantum error correction. In this approach, k logical qubits with Hilbert space $\mathcal{H}_L \simeq \mathcal{Q}^{\otimes k}$ are encoded via an encoding process $\mathcal{E}$ into n physical qubits. These physical qubits are then subjected to a noise channel $\mathcal{N}$ before being subjected to a decoding process $\mathcal{D}$. The quantum error-correcting performance of this scheme is given by how close the encoding-noise-decoding mechanism resembles unity: $\mathcal{D} \circ \mathcal{N} \circ \mathcal{E} \approx \mathcal{I}$.[5] This performance is quantifiable either via the average fidelity or worst-case fidelity of the quantum process.

The process of variationally correcting quantum error begins with the parametrization of the encoding/decoding mechanism via parameterized unitary gates. These unitary gates often involve rotations along an axis (parametrized Pauli-X or Pauli-Z gates). At the heart of QVECTOR is obtaining the optimal parameters to perform these rotations in order to mitigate the error caused by the channel $\mathcal{N}$.

5.2 Quantum Convolutional Neural Networks

With the idea of a variational quantum error correction scheme present, I begin this section by introducing a quantum convolutional neural network: a scheme that combines both a variational approach and a convolution of the state. I show the construction of the QCNN, and show how this scheme

[5] Recall that in quantum operator formalism, the operations are applying beginning with the right-most operator.

performs in correcting error for a particular error channel.

5.2.1 Multiscale Entanglement Renormalization Ansatz

The particular construction of this QCNN is motivated by the multiscale entanglement renormalization ansatz (MERA) [5]. The multiscale entanglement renormalization ansatz is a structure that efficiently encodes quantum many-body states of D-dimensional lattice systems and from which local expectation values can be computed exactly. A MERA consists of a network of isometric tensors in D+1 dimensions. The extra dimension can be seen as parametrizing different length scales in the system, according to successive applications of a lattice coarse-graining procedure known as entanglement renormalization [52]. Each isometry layer introduces a set of new qubits in a predetermined state, such as $|0\rangle$, before applying unitary gates on nearby qubits. A schematic of the MERA architecture is shown in Figure 9.

In relation to QCNNs, MERA shares the same circuit structure, but runs in reverse direction. Hence, for any given state ψ with a MERA representation, there is always a QCNN that recognizes ψ with deterministic measurement outcomes. If a state ψ that is prepared with a MERA representation is disturbed, the QCNN will no longer produce deterministic measurement outcomes, and the measurement outcomes become syndrome measures for quantum error correction. These syndrome measures determine specific rotation unitaries to apply to the remaining qubits. A QEC scheme based on a MERA and QCNN architecture is able to detect and correct local quantum errors without collapsing the wavefunction, even if those errors are correlated [7, 52].

5.2.2 Implementation of QCNN

A QCNN takes as input an unknown quantum state ρ_{in}. Convolution, pooling, and fully-connected layers are realized by quantum unitary gates, as seen in Figure 10. For the convolution layer, a quasi-local unitary is applied in a translationally-invariant manner. For the pooling layer, a fraction of the qubits are measured, and the results of these measurements determine rotations on the fraction of unmeasured qubits. After a series of convolution-pooling layers the system size is small enough and a fully-connected layer is applied. Finally, the outcome of the QCNN is obtained by measuring a fixed number of the output qubits. For a QCNN the unitaries corresponding to the convolution, pooling, and fully-connected layers are learned.

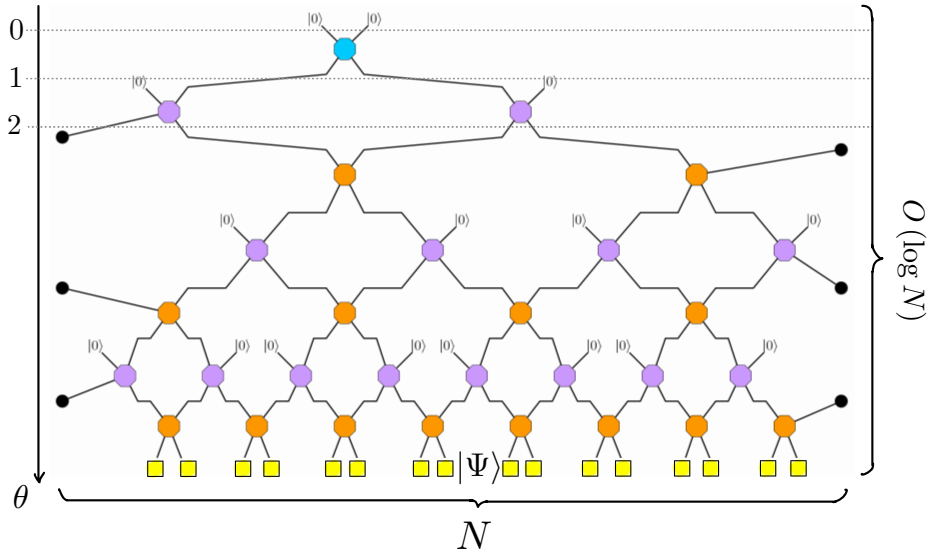

Figure 9: Representation of a MERA architecture. Reading the schematic from top to bottom: the blue octagon represents the logical qubit to be protected. At each branch (which can be interpreted as discrete times that give the depth Θ of the circuit) of the MERA, the qubit is entangled with an ancilliary qubit (in this case, with the state $|0\rangle$). This quantum circuit transforms the state $|0\rangle^{\otimes N}$ into the N-site state $|\Psi\rangle$ of a 1D lattice. The circuit contains $2N-1$ gates organized in $O(\log N)$ layers labeled by a discrete time Θ.

The unitaries are parametrized as exponentials of generalized $a \times a$ Gell-Mann matrices $\{\Lambda_i\}$, where $a = 2^w$ and w is the number of qubits involved in the unitary, such that $U = \exp\left(-i \sum_j c_j \Lambda_j\right)$. The QCNN learning procedure involves optimizing the coefficients c_μ of the Gell-Mann matrices [7]. This procedure is performed by computing the derivative of the mean-squared error function,

$$MSE = \frac{1}{2M} \sum_{\alpha=1}^{M} (y_\alpha - f_{\{U_i\}}(|\psi_\alpha\rangle))^2 \tag{35}$$

as:

$$\frac{\partial MSE}{\partial c_\mu} = \frac{1}{2\epsilon}(MSE(c_\mu + \epsilon) - MSE(c_\mu - \epsilon)) + \mathcal{O}(\epsilon^2). \tag{36}$$

In the expression for mean-error squared, $f_{\{U_i\}}(|\psi_\alpha\rangle)$ represents the expected QCNN output value for input $|\psi_\alpha\rangle$. For this instance, the expected value is

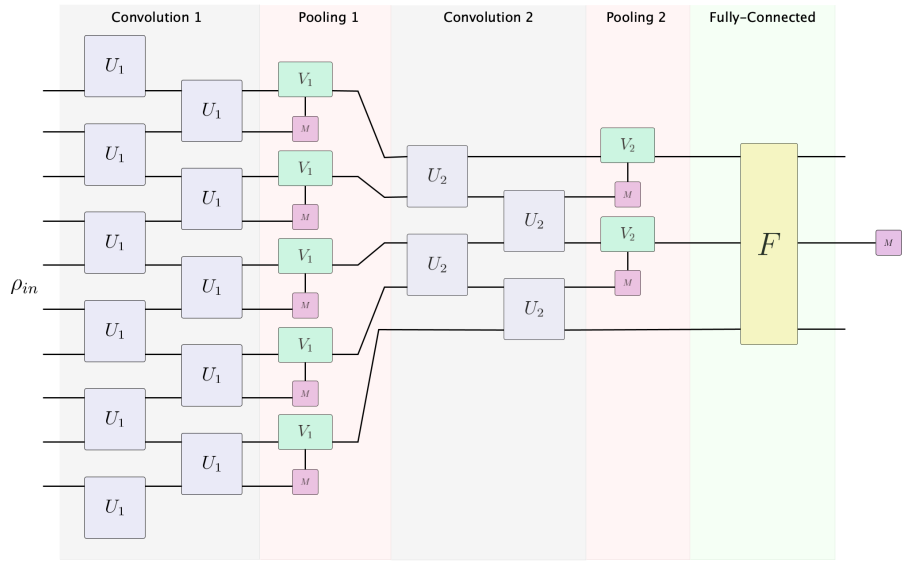

Figure 10: Representation of a QCNN architecture. The state ρ_{in} is an unknown quantum state. The state goes through a series of convolution and pooling layers, marked by unitaries U_i and V_i, respectively. In the pooling layer, measurements are performed on a fraction of qubits to determine rotations on the remaining qubits. Finally, the qubits pass through a fully-connected layer F, and a measurement is performed on a fraction of qubits.

the fidelity:

$$f_q = \sum_{|\psi_l\rangle \in \{|\pm x, \pm y, \pm z\rangle\}} \langle\psi_l| \, \mathcal{M}_q^{-1}(\mathcal{N}(\mathcal{M}_q(|\psi_l\rangle \langle\psi_l|))) \, |\psi_l\rangle \,, \tag{37}$$

where $\mathcal{M}_q$ and $\mathcal{M}_q^{-1}$ are the encoding and decoding schemes generated by a QCNN circuit, and $|\pm x, \pm y, \pm z\rangle$ are the ± 1 eigenstates of the Pauli matrices. $\mathcal{N}$ is the error channel, which is constructed from X-, Y-, and Z-gates, as shown in Figure 11.

The derivation of the MSE is performed to first order with respect to the coefficients c_μ by using the finite-difference method, so that the coefficients are updated as $c_\mu \to c_\mu - \eta \frac{\partial MSE}{\partial c_\mu}$, where η is the learning rate for that iteration. Furthermore, the learning rate is computed using the bold driver technique: η is increased by 5% if the error has decreased from the previous iteration, and decreased by 50% otherwise [53].

This encoding mechanism, based on the MERA representation, is reminis-

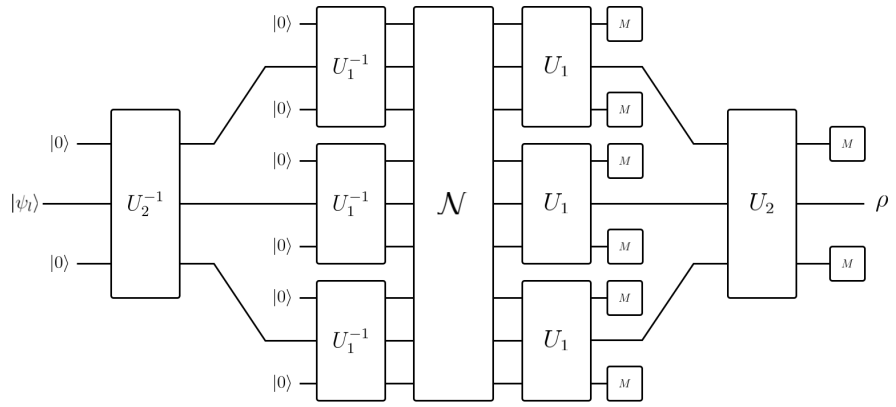

Figure 11: Representation of a QCNN/MERA performing QEC. The state $|\psi_l\rangle$ is passed through two layers of unitary matrices. This entangles the ancilla qubits to the logical qubit. After the state passes through the second layer U_1^{-1}, noise is introduced by an error channel $\mathcal{N}$. This error channel can act independently on any of the qubits, as well as in multiple qubits. Finally, through a series of measurements (denoted by M), the final state is recovered. Comparison of the initial input state to the final output state is done via fidelity measurements.

cent of the stabilizer formalism. A series of ancilla qubits are appended to the logical qubit, and thus the logical qubit can be protected. Through optimization of the Gell-Mann matrix coefficients for a particular error channel, the logical qubit can be better protected and therefore have higher fidelity after the decoding process. In other words, this quantum convolutional neural network stabilizes a logical qubit prior to the error channel, and rebuilds the pure state after the error channel. For experimental considerations, the error channel $\mathcal{N}$ is generally unknown since the fidelity f_q can be measured directly.

The error channel is realized by applying a (generally anisotropic) depolarization quantum channel to each of the physical qubits. This means that, with a certain probability, either a bit-flip (Pauli-X matrix) or phase-flip (Pauli-Z matrix) or both (Pauli-Y matrix) is applied to each qubit. Analytically, this means:

$$\mathcal{N}_{1,i} : \rho \to (1 - \sum_{\mu} p_\mu)\rho + \sum_{\mu} p_\mu \sigma_i^\mu \rho \sigma_i^\mu, \tag{38}$$

where the σ_i^μ are the Pauli matrices for $i \in \{1, 2, ..., N\}$. The expression above does not consider correlated errors. For correlated errors, a second quantum error channel is applied (for example, for two-qubit correlated errors $Z_i Z_{i+1}$

with probability p_{zz}):

$$\mathcal{N}_{2,i} = \rho \to (1 - p_{zz})\rho + p_{zz} Z_i Z_{i+1} \rho Z_i Z_{i+1}, \tag{39}$$

for pair of nearby qubits $i \in \{1, 2, 4, 5, ...\}$. The output state, after the system has gone through the error channel, is compared to the input state via a fidelity measure. The coefficients of the Gell-Mann matrices are optimized so that the fidelity between input and output state is maximized. In general, for an anisotropic logical error model with probabilities p_μ for σ_μ logical errors, the overlap f_q is:

$$f_q = (1 - 2\sum_\mu \frac{p_\mu}{3}), \tag{40}$$

since $\langle \pm\nu | \, \sigma_\mu \, | \pm\mu \rangle = (-1)^{\delta_{\mu,\nu}+1}$. Therefore, the total logical error probability from f_q is computed as $1.5(1 - f_q)$. These expressions are derived from the analytical form of the error channel(s). Computationally, the optimization procedure is performed by a *stochastic* error channel. Using a stochastic error channel has more relevance to physical implementations of this model, since a logical qubit that undergoes the encoding process and is subjected to a quantum error channel can never go back to its original state.

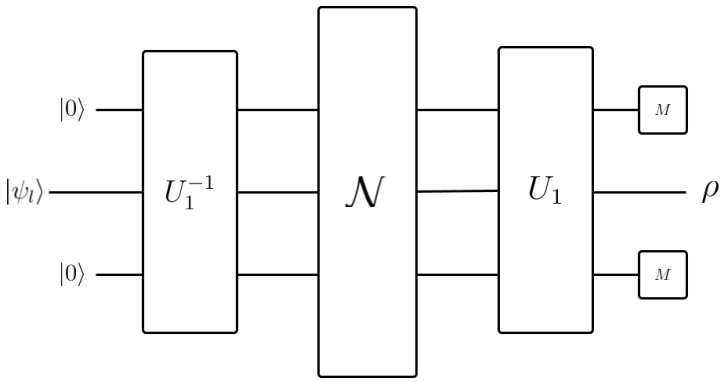

Figure 12: Simple implementation of a 3-qubit QCNN architecture.

Figure 12 shows an implementation of a 3-qubit QCNN architecture. Based on this architecture, error can be mitigated, as shown in Figure 13.

Here, the error channel $\mathcal{N}$ is given by,

$$\mathcal{N}_i : \rho \longrightarrow (1 - p_x)\rho + p_x \sigma_i^x \rho \sigma_i^x, \tag{41}$$

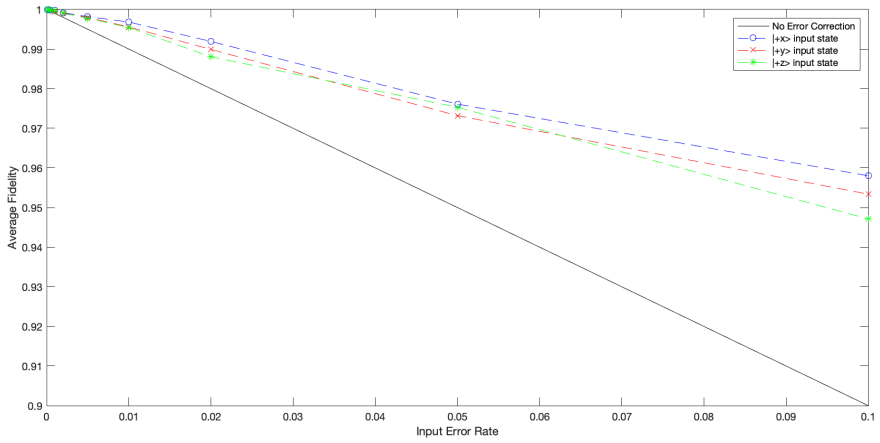

Figure 13: Optimized QCNN: average fidelity measurement for various input error rates. Solid black line represents no error correction. Colored dashed lines represent different input states.

where $i \in \{1, 2, 3\}$. This implementation of a QCNN was done in MATLAB, simulating the time evolution of the density matrix in a tensor network representation approach. The results were obtained by performing the encoding and decoding mechanism over ∼10,000 shots and averaging individual fidelities by following the prescription of repeating the gradient descent procedure until the error function changes on the order of 10^{-5} between successive iterations. It can be seen that there is a clear increase in the average quantum fidelity for all three input states. This can also be seen in Figure 14.

In general for all input states, the 3-qubit QCNN is able to reduce logical error rate around ∼50-60%. The percent decrease in logical error rate for different input error rates is summarized in Table 1.
For all three input states, there is a decrease in logical error as compared to no error. This method demonstrates the power of a variational approach to "learn" errors and mitigate them. Even with this reduction in logical error rates, a much greater fidelity is needed in order to achieve large-scale, fault-tolerant quantum computing. In the example above, there are very few unitary gates, making the circuit very shallow. As mentioned previously, these variational approaches work well for shallow-depth circuits. The real crux becomes employing these quantum correction algorithms for more complex circuits, as these circuits are the ones that will have an advantage over classical computing.

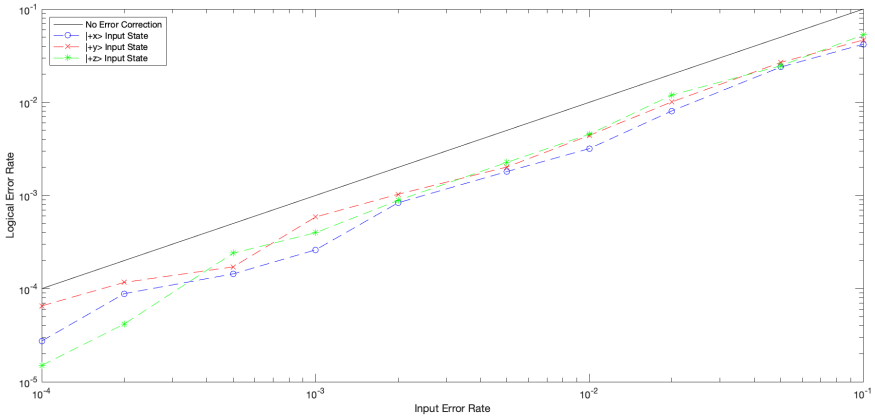

Figure 14: Log-log plot for input error rate vs. logical error rate. Solid black line represents no error correction. Colored dashed lines represent logical error rates for different input states.

Table 1: Percent Logical Error Decrease in 3-qubit QCNN

p	0.1	0.05	0.02	0.01	0.005	0.002	0.001	0.0005	0.0002	0.0001
$\|+x\rangle$	58%	52%	60%	68%	63%	58%	74%	71%	56%	73%
$\|+y\rangle$	53%	46%	49%	56%	59%	48%	41%	66%	42%	35%
$\|+z\rangle$	47%	51%	40%	54%	54%	56%	60%	52%	79%	85%

6 Conclusions

In this work, I have presented an implementation of a quantum convolutional neural network as an effective tool for quantum error correction. The power behind this approach comes from the fact that, being variational, the algorithm finds an error correction scheme for arbitrary noise. That is, by finding the vector of optimal parameters (the coefficients to the Gell-Mann matrices) for a particular error channel, a decoding procedure can be applied to recover the originally-encoded logical qubit such that the fidelity increases, as opposed to having no error correction.

Additionally, the fact that this model is constructed in the context of the multi-scale entanglement renormalization ansatz, facilities the simulation of the circuits, since a tensor network approach can be used. For circuits of shallow depth, this tensor network approach is much less computationally expensive as, for example, carrying out the time evolution of the entire density matrix.

Furthermore, given the fact that neural networks can be trained on highly non-linear models, a quantum convolutional neural network approach to error correction might lead to mitigation of errors past arbitrary single-qubit errors. In realistic scenarios, quantum hardware is exposed to quantum noise that might affect one or more qubits simultaneously, and that might be correlated in one way or another. Therefore, it is of great importance to find effective tools to combat these types of noise moving forward.

Of the current error-correcting techniques, variational quantum approaches surface as good candidates, given the decrease in physical qubits, as opposed to other techniques like the surface and color codes. Furthermore, these variational approaches have a wide range of applications, making the study of these important for all areas of science and technology.

Nevertheless, improvements on these algorithms are still necessary, as many of the proposed techniques fail for large-depth circuits. Additionally, it is important to pay close attention to the development of these algorithms in regard to their physical implementation and current quantum devices.

In broad strokes, the path to large-scale, fault-tolerant quantum technologies is still long and it will require a great effort. The issues of scalability, quantum control, physical implementations, and algorithms, to name a few, must be tackled with a close interdisciplinary collaboration. In particular, the field of quantum error correction must provide efficient, hardware-implementable techniques to achieve this goal.

With this work, I hope to bring attention to the current state of quantum error correction, as well as the hurdles still to overcome in the quest for reliable quantum technologies.

References

[1] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018. arXiv: 1801.00862.

[2] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, September 2017.

[3] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, October 2019.

[4] Xiao Liang, Sheng Liu, Yan Li, and Yong-Sheng Zhang. Generation of Bose-Einstein Condensates' Ground State Through Machine Learning. *arXiv:1712.10093 [quant-ph]*, December 2017.

[5] G. Vidal. Class of Quantum Many-Body States That Can Be Efficiently Simulated. *Physical Review Letters*, 101(11):110501, September 2008.

[6] Nobuyuki Yoshioka and Ryusuke Hamazaki. Constructing neural stationary states for open quantum many-body systems. *Physical Review B*, 99(21):214306, June 2019.

[7] Iris Cong, Soonwon Choi, and Mikhail D. Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, December 2019.

[8] Raymond Laflamme, Cesar Miquel, Juan Pablo Paz, and Wojciech Hubert Zurek. Perfect Quantum Error Correcting Code. *Physical Review Letters*, 77(1):198–201, July 1996.

[9] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Physical Review A*, 52(4):R2493–R2496, October 1995.

[10] A. M. Steane. Error Correcting Codes in Quantum Theory. *Physical Review Letters*, 77(5):793–797, July 1996.

[11] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098–1105, August 1996.

[12] M. D. Reed, L. DiCarlo, S. E. Nigg, L. Sun, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf. Realization of three-qubit quantum error correction with superconducting circuits. *Nature*, 482(7385):382–385, February 2012.

[13] P. Schindler, J. T. Barreiro, T. Monz, V. Nebendahl, D. Nigg, M. Chwalla, M. Hennrich, and R. Blatt. Experimental Repetitive Quantum Error Correction. *Science*, 332(6033):1059–1061, May 2011.

[14] Charles D. Hill, Eldad Peretz, Samuel J. Hile, Matthew G. House, Martin Fuechsle, Sven Rogge, Michelle Y. Simmons, and Lloyd C. L. Hollenberg. A surface code quantum computer in silicon. *Science Advances*, 1(9):e1500707, October 2015.

[15] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins, K. M. Hudek, J. Mizrahi, J. D. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. Ducore, A. Blinov, S. M. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim. Benchmarking an 11-qubit quantum computer. *Nature Communications*, 10(1):5464, November 2019.

[16] Kosuke Fukui, Akihisa Tomita, and Atsushi Okamoto. Tracking quantum error correction. *Physical Review A*, 98(2):022326, August 2018.

[17] Morten Kjaergaard, Mollie E. Schwartz, Jochen Braumüller, Philip Krantz, Joel I.-Jan Wang, Simon Gustavsson, and William D. Oliver. Superconducting Qubits: Current State of Play. *arXiv:1905.13641 [cond-mat, physics:physics, physics:quant-ph]*, May 2019.

[18] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. Noisy intermediate-scale quantum (NISQ) algorithms. *Reviews of Modern Physics*, 94(1):015004, February 2022. arXiv:2101.08448 [cond-mat, physics:quant-ph].

[19] F. Vatan, V. P. Roychowdhury, and M. P. Anantram. Spatially Correlated Qubit Errors and Burst-Correcting Quantum Codes. *arXiv:quant-ph/9704019*, April 1997.

[20] Chi-Kwong Li, Mikio Nakahara, Yiu-Tung Poon, Nung-Sing Sze, and Hiroyuki Tomita. Efficient Quantum Error Correction for Fully Correlated Noise. *Physics Letters A*, 375(37):3255–3258, August 2011.

[21] Emanuel Knill, Raymond Laflamme, and Lorenza Viola. Theory of Quantum Error Correction for General Noise. *arXiv:quant-ph/9908066*, August 1999.

[22] 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(9):625–644, August 2021.

[23] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[24] Ramamurti Shankar. *Principles of quantum mechanics*. Plenum, New York, NY, 1980.

[25] Simon J. Devitt, Kae Nemoto, and William J. Munro. Quantum Error Correction for Beginners. *Reports on Progress in Physics*, 76(7):076001, July 2013.

[26] T. Brun, I. Devetak, and M.-H. Hsieh. Correcting Quantum Errors with Entanglement. *Science*, 314(5798):436–439, October 2006.

[27] P.G. Kwiat and D.F.V. James. Quantum optics — entanglement and quantum information. In Robert D. Guenther, editor, *Encyclopedia of Modern Optics*, pages 256–264. Elsevier, Oxford, 2005.

[28] J. Chiaverini, D. Leibfried, T. Schaetz, M. D. Barrett, R. B. Blakestad, J. Britton, W. M. Itano, J. D. Jost, E. Knill, C. Langer, R. Ozeri, and D. J. Wineland. Realization of quantum error correction. *Nature*, 432(7017):602–605, December 2004.

[29] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane. Quantum Error Correction and Orthogonal Geometry. *Physical Review Letters*, 78(3):405–408, January 1997.

[30] Daniel Gottesman. Stabilizer Codes and Quantum Error Correction. *arXiv:quant-ph/9705052*, May 1997.

[31] H. Barnum and E. Knill. Reversing quantum dynamics with near-optimal quantum and classical fidelity. *Journal of Mathematical Physics*, 43(5):2097, 2002.

[32] Philipp Schindler, Thomas Monz, Daniel Nigg, Julio T. Barreiro, Esteban A. Martinez, Matthias F. Brandl, Michael Chwalla, Markus Hennrich, and Rainer Blatt. Undoing a Quantum Measurement. *Physical Review Letters*, 110(7):070403, February 2013.

[33] D. Ristè, S. Poletto, M.-Z. Huang, A. Bruno, V. Vesterinen, O.-P. Saira, and L. DiCarlo. Detecting bit-flip errors in a logical qubit using stabilizer measurements. *Nature Communications*, 6(1):6983, November 2015.

[34] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. Error Mitigation for Short-Depth Quantum Circuits. *Physical Review Letters*, 119(18):180509, November 2017.

[35] Ying Li and Simon C. Benjamin. Efficient Variational Quantum Simulator Incorporating Active Error Minimization. *Physical Review X*, 7(2):021050, June 2017.

[36] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1):4213, July 2014.

[37] Dave Wecker, Matthew B. Hastings, and Matthias Troyer. Progress towards practical quantum variational algorithms. *Physical Review A*, 92(4):042303, October 2015.

[38] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, February 2016.

[39] David Fumo. Types of machine learning algorithms you should know [blog]. `https://www.scientificstyleandformat.org/Tools/SSF-Citation-Quick-Guide.html`. Accesed: 2021-11-22.

[40] Ashish Sukhadeve. Understanding neural networks: A beginner's guide [blog]. https://www.datasciencecentral.com/profiles/blogs/understanding-neural-network-a-beginner-s-guide. Accesed: 2021-11-22.

[41] Denny Novikov. *Machine Learning: The Ultimate Beginners Guide to Efficiently Learn and Understand Machine Learning, Artificial Neural Network and Data Mining*. Independently Published, 2019.

[42] G. I. Diaz, A. Fokoue-Nkoutche, G. Nannicini, and H. Samulowitz. An effective algorithm for hyperparameter optimization of neural networks. *IBM Journal of Research and Development*, 61(4/5):9:1–9:11, 2017.

[43] Kevin Gurney. *Introduction to Neural Networks*. Taylor & Francis, Oxford, 1997. OCLC: 892785047.

[44] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[45] Alexandra Nagy and Vincenzo Savona. Variational Quantum Monte Carlo Method with a Neural-Network Ansatz for Open Quantum Systems. *Physical Review Letters*, 122(25):250501, June 2019.

[46] Michael J. Hartmann and Giuseppe Carleo. Neural-Network Approach to Dissipative Quantum Many-Body Dynamics. *Physical Review Letters*, 122(25):250502, June 2019.

[47] Filippo Vicentini, Alberto Biella, Nicolas Regnault, and Cristiano Ciuti. Variational Neural-Network Ansatz for Steady States in Open Quantum Systems. *Physical Review Letters*, 122(25):250503, June 2019.

[48] Richard P Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, page 22, 1982.

[49] Roman Orus. A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States. *Annals of Physics*, 349:117–158, October 2014.

[50] Peter D. Johnson, Jonathan Romero, Jonathan Olson, Yudong Cao, and Alán Aspuru-Guzik. QVECTOR: an algorithm for device-tailored quantum error correction, November 2017. arXiv:1711.02249 [quant-ph].

[51] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, September 2012.

[52] G. Vidal. Entanglement Renormalization. *Physical Review Letters*, 99(22):220405, November 2007.

[53] Geoffrey Hinton. Lecture Notes for CSC2515: Lecture 6, 2007.