A Model for Ideal and Non-Ideal Behavior of Quantum Circuits

Arpan Gupta
arpan.gupta@rutgers.edu
Rutgers University, ECE + CS Department

Michael Schleppy
michael.schleppy@rutgers.edu
Rutgers University, ECE + Math Department

1 PROBLEM AND MOTIVATION

Since the 1980s, researchers have entertained using the bizarre properties of quantum mechanics to develop powerful algorithms for problems in computer science. Ever since the first quantum computer was built, rapid advances in the field have opened up new realms of possibilities for the different applications of quantum computing. Although still in its infancy, the near future Noisy Intermediate-Scale Quantum (NISQ) devices will be able to run algorithms that beat classical computing [6]. The implications of such a feat resound in various fields including cryptography, biotechnology, finance, and many more.

The inherent nature of NISQ devices means that they will be noisy. To advance towards a future of quantum computing, it is imperative that these machines exhibit fault tolerance and error correction, akin to their classical computing counterparts. This means complete isolation from the external environment, as well as well defined interactions between qubits. In order to achieve this state, researchers rely on simulation models to incorporate noisy events into quantum circuits. By studying such interactions, researchers can work on incorporating error correcting codes into quantum systems. Furthermore, with an expanding number of qubits, the likelihood of error events rises, bringing upon a limitation to the capacity of NISQ devices. This limitation makes it ever more important to have excellent simulation models.

Current popular simulation frameworks include IBM Qiskit [8] and Google Cirq [2], both of which have extensive capabilities such as simulation and the ability to run circuits on a real quantum computer. However, these simulation tools focus primarily on noise-free, ideal circuits. These circuits do not accurately model the observations of real quantum computers where noise events are present. Furthermore, there exist varying different languages to run different simulations between ideal circuits and noisy circuits. These varying languages for the diverse layers of quantum representations require different rules on composition and operation. Such tools for representation include state vectors, density matrices, and Kraus operators. This leads to many challenges for researchers looking to simulate both simple and complex noisy circuits in an efficient and unified model.

We propose a unified language model to accurately simulate ideal and noisy circuits in all degrees of complexity, through the use of complex-valued probabilistic graphical models (PGMs). PGMs prove to be a highly effective mechanism for encoding quantum information, facilitating the depiction of correlations between quantum states, and ensuring the precise retention of state information, even amidst probabilistic noisy events. Furthermore, PGMs can unify the language of quantum operations and noise models through a single algorithm - variable elimination. We validate our results through the use of a python PGM library - pgmpy [1], and compare results with proven derivations.

2 BACKGROUND

In this section, we provide comprehensive background information for important concepts in our research. We begin by introducing key quantum computing principles used for representation of quantum circuits including state vectors and density matrices. We then provide background to different quantum noise channels and introduce one method of representing these noise channels using Kraus operators. Finally, we provide information on general probabilistic graphical models (PGMs).

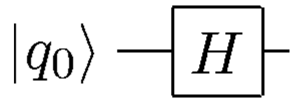


Figure 1: Circuit Example with Hadamard Gate

2.1 Quantum Computing

Classical computing systems use transistors to represent information as bits, which can exist in one of two states, either 0 or 1. Quantum computing systems store information in a very different way using qubits or quantum bits. The properties of quantum mechanics state that qubits can exist in a superposition of 0 and 1. The key notable properties of qubits include: superposition and entanglement. Quantum entanglement is a phenomenon in which two or more quantum systems become correlated in such a way that the state of each system cannot be described independently of the others. This means that the measurement of one part of the entangled system will instantly affect the state of the other system, regardless of the distance between the two systems. These two properties are the key to the power of quantum computers, since it allows for much faster and more efficient computations compared to classical computers.

Quantum information is represented using $|\psi\rangle$ which represents a superposition of the classical states, 0 and 1. When a quantum system is measured, the qubits always collapse to a definite value. In order to exploit the benefits of quantum mechanics, quantum algorithms manipulate the interference between entangled qubits to allow some states to have higher amplitudes, improving the probability of collapsing to that state. An n-qubit system can be represented by $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$ where α_i is a complex number representing the probability of the qubit collapsing to that state. An entangled qubit system can be represented with: $|\psi\rangle = \alpha |00\rangle + \beta |11\rangle$. α and β are complex numbers that represent the probability of the qubit being in that state when measured, meaning $|\alpha|^2 + |\beta|^2 = 1$. The states $|0\rangle$ and $|1\rangle$ are the orthonormal basis vectors $\begin{bmatrix} 1 & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & 1 \end{bmatrix}$ respectively.

Quantum circuits provide a means of representing and manipulating quantum information via a sequence of quantum gates, enabling the realization of quantum algorithms with a multitude of promising applications. Quantum gates are analogous to logic gates in classical computing. A quantum gate is nothing but a unitary operator that acts on one or more qubits, thus transforming the state of the qubits according to the laws of quantum mechanics. These transformations include rotation, phase shift, or entanglement. Some common gates are the Hadamard gate which creates a superposition of $|0\rangle$ and $|1\rangle$, and the CNOT gate which entangles two qubits. By combining different quantum gates in specific sequences, quantum algorithms can be designed to solve problems faster and more efficiently than classical algorithms.

An example of a quantum circuit representing a one-qubit system undergoing a Hadamard transform is shown in Figure 1. The mathematical equivalence can be represented as:

$$H = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

We can apply this matrix to the quantum state $|0\rangle$ giving us:

$$H|0\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

2.2 State Vector

A state vector is used to represent pure states in quantum computing. A quantum system can be in a superposition of different states, and the state vector specifies the amplitude and phase of each possible state of the system. The state vector is represented in the Hilbert space, and by the postulates of quantum mechanics has unit norm. When you make a measurement on a qubit, the state vector collapses to a single state. An example of a state vector can be shown where a single qubit undergoes a Hadamard transform. We got the final state as: $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$.

2.3 Density Matrix

In quantum computing, a density matrix is a mathematical representation for states that are not in a pure state. Pure states are described using state vectors in Hilbert space, whereas mixed states are represented by density matrices, which are Hermitian matrices that represent the different probabilities of different outcomes in quantum measurement. The density matrix can be represented by taking the outer product of the state vector of a qubit - $|\psi\rangle\langle\psi|$, where ψ is the state vector of the qubit. The density matrix $\rho = \sum_i p_i(\psi_i)\,|\psi_i\rangle\langle\psi_i|$ where $p_i(\psi_i)$ represents the probability of of the system being in the quantum state $|\psi_i\rangle$. An example of a density matrix for the Bell state is:

$$|\Phi+\rangle = \frac{(|00\rangle + |11\rangle)}{\sqrt{2}}$$
$$\rho = |\Phi+\rangle\langle\Phi+|$$
$$\rho = \frac{(|00\rangle + |11\rangle)}{\sqrt{2}} * \frac{(\langle 00| + \langle 11|)}{\sqrt{2}}$$
$$\rho = \frac{(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|}{2}$$
$$\rho = \frac{1}{2}\begin{bmatrix}1 & 0 & 0 & 1\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\1 & 0 & 0 & 1\end{bmatrix}$$

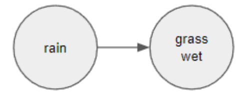
The diagonal terms of a density matrix are the elements of the form $\langle i|\rho|i\rangle$ where i represents a basis state of the system. These terms give the probability of finding the system in the state $|i\rangle$ when measured. For example, suppose we have a qubit in the state: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. The density matrix for this system would be: $\rho = |\psi\rangle\langle\psi|$. The diagonal terms of this density matrix are $\langle 0|\rho|0\rangle = \alpha^2$ and $\langle 1|\rho|1\rangle = \beta^2$. These terms give the probabilities of finding the qubit in the $|0\rangle$ or $|1\rangle$ state, respectively, when measured.

2.4 Quantum Noise

In ideal conditions, we assume qubits to be in a completely isolated system. This means there is no interaction with the environment outside the system, mitigating error events caused by the entanglement of the system and environment. The qubits are all also assumed to be coherent. In reality, qubit systems are not completely isolated from the environment and many different interactions can lead to different noise events. There are many different noise channels for quantum computing, all of which fall under the categories of coherent and decoherent noise. Coherent noise events arise from the quantum properties of the system itself meaning fluctuations affect all parts of the system in the same way. This type of noise is easy to correct and can often be exploited for quantum information processing tasks. Decoherent noise arises from interactions between the quantum system and the environment. This type of noise can limit the ability of a quantum system to perform useful tasks. Examples of this noise include: depolarizing, flip, and dampening. Each of these noise channels can occur from a different interaction within qubits, as well as with the environment, yielding potentially varying outcomes.

For more information on these noise channels, please refer to the textbook Quantum Computation and Quantum Information [7].

Rain	True	False
	0.2	0.8



rain/grass wet	True	False
True	1	0
False	0	1

Figure 2: Bayesian Network Example

2.5 Kraus Operators

Kraus operators are mathematical representations of quantum operations, which describe how a quantum system is affected by a quantum channel. The Kraus operators for a quantum channel are typically represented as a set of linear operators $\{K_1, K_2, ..., K_i\}$ that satisfy the condition $\sum K_i^\dagger K_i = I$ where $K_i^\dagger$ represents the Hermitian conjugate of K_i. We can use Kraus operators to describe the evolution of a density matrix ρ under the influence of a quantum channel, like a noise channel, as: $\rho' = \sum K_i \rho K_i^\dagger$ where ρ' is the density matrix after applying the quantum channel to density matrix ρ.

2.6 Pauli and Clifford Set

[3] The Pauli gates are a set of 4 quantum gates - I, X, Y, Z. The significance of them lies in the fact that you can represent any single-qubit quantum state as a linear combination of Pauli gates. The Pauli gate matrices look like:

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

The Clifford set is a set of quantum gates that include the Pauli gates, Hadamard gate, phase gate, and CNOT gate. Each one of these gates can be represented with a combination of the Pauli gates, making the Pauli gates a very powerful set. Thus, we use the Pauli gates to represent possible states in our model.

2.7 Probabilistic Graphical Models

PGMs are very powerful models that are frequently used in machine learning and artificial intelligence applications. The idea behind PGMs is to use a graphical model to represent the dependencies between variables, and to combine this with probability distributions to specify the likelihood of each variable taking a particular value based on its neighbors.

A key advantage of PGMs is the ability to run exact inference algorithms like variable elimination to efficiently eliminate variables not of interest and obtain the desired distribution. Variable elimination is equivalent to tensor contraction in tensor networks, another type of graphical framework used to represent complex mathematical objects in a compact and efficient way. This single operation can effectively simulate all types of complex quantum circuits, include noisy and noise-free circuits.

Two different useful PGMs include Markov networks and Bayesian networks. Markov networks are undirected graphs that model conditional independence relationships between variables. On the other hand, Bayesian networks are directed, acyclic graphs that model causal relationships between variables using conditional probability. Bayesian networks must obey classical probability rules including normalization.

Previous work on using PGMs to encode quantum circuits was done, and demonstrates the effectiveness and efficiency of such a logical abstraction for quantum information [4].

The normalization of quantum information presents a challenge when implementing Bayesian networks, as it may not always be feasible to normalize quantum probability distributions due to the nature of quantum mechanics.

Figure 2 shows a Bayesian network with two events - rain and grass wet. Each event is represented by a node, along with a conditional probability table representing the dependence of each event on other events. The arrows represent the direction of the dependence. We can

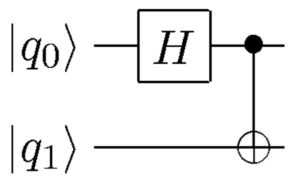


Figure 3: Bell State Circuit

look at an example conclusion drawn from the network, P(grass wet | rain) = 1.

The process of conversion from a Bayesian network to Markov network involves one main step: all parents of a node get an edge. This leads to added edges between nodes, and updates to tables associated with those nodes. Furthermore, there is no direction in Markov networks.

For further reading on the definitions and properties of Markov and Bayesian networks, as well as information on inference algorithms, please refer to the textbook Probabilistic Graphical Models: Principles and Techniques [5].

3 APPROACH AND UNIQUENESS

Current simulation models greatly lack the ability to simulate complex noise events, and rely on elaborate simulation tools to model simple noise events. This introduces a large need for a single, unified model to represent all aspects of noise events. Quantum researchers rely heavily on simulations to work on noise mitigation and correction algorithms, a key component to advance the future of quantum computing. Furthermore, with the limitations of the number of qubits in the NISQ-era, the need for accurate and reliable quantum systems remains extremely high.

The inherent nature of PGMs, specifically Bayesian networks, makes them a great abstraction for representing causal relationships between states. A quantum circuit entails a sequence of quantum gates, which perform unitary transformations on quantum states, generating dependencies between prior and subsequent states. Because of this, we can represent their amplitudes using conditional probability tables (CPTs). We use the Pauli gates to represent our possible states, since we know they can be used to represent any quantum state. An example of what this looks like for the Bell state circuit is shown in Figure 4. The network depicts a two-qubit system, with the upper qubit undergoing a Hadamard transform, and the lower qubit undergoing a CNOT transformation. This network is equivalent to the circuit shown in Figure 3. We can use variable elimination to simulate this circuit using a single operation. This greatly simplifies the normally complex operations required to simulate quantum circuits.

Classical probability rules enforce normalization on Bayesian network conditional probability tables. This is excellent for traditional Bayesian networks, since it can be used to verify the validity of the probabilistic outcomes. For representation of quantum information, we unfortunately can not abide by these rules since we are representing quantum amplitudes, and not pure probabilities. Because of this, we must convert our networks to Markov networks. This allows us to use the majority of the same tools as a Bayesian network, without the constriction of normalization.

Converting a network from a Bayesian network to a Markov network can sometimes be a complicated and tedious process. In order to mitigate this, we use pgmpy to create our networks as Bayesian networks, and then convert them to Markov networks before running our inference algorithms on them. After conducting thorough research and analysis, we have developed a unified language model that integrates Markov networks and variable elimination techniques. This innovative approach enables us to effectively and precisely simulate all quantum circuits with high efficiency and accuracy.

We can show an example of this process with an ideal, noiseless quantum circuit. Figure 3 shows the quantum circuit involving a two

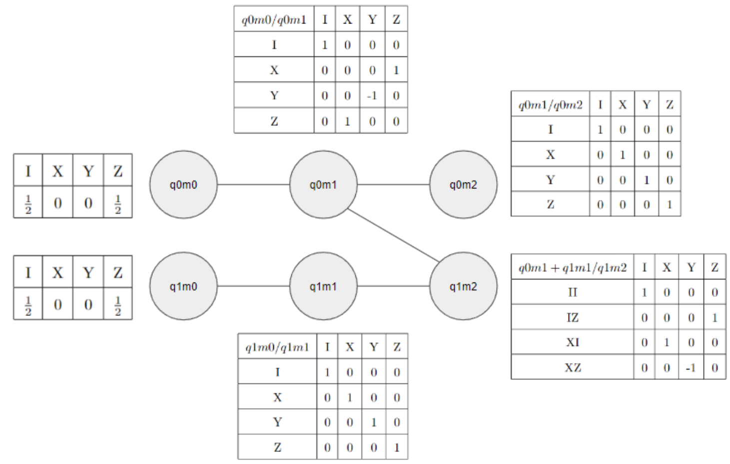


Figure 4: Network Model for Bell State Circuit with initial state $|0\rangle$ and $|0\rangle$

qubit system, with the upper qubit undergoing a Hadamard gate, and then a CNOT gate with the upper qubit being the control bit and lower qubit being the target bit. This circuit is frequently used since it puts two qubits into the maximally entangled state, known as the Bell state. Figure 4 shows our equivalent representation using a network. We can use this example and compare results of using the standard density matrix approach, and our proposed Pauli matrix and PGM approach.

3.1 Standard Density Matrix Approach

We begin our approach by setting qubit 0 to state $|0\rangle$:

$$\rho = |0\rangle\langle 0| = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

Next, a Hadamard gate is applied to qubit 0. Earlier, we derived what the Hadamard gate does

to a qubit with state $|0\rangle$: $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

We can continue working with density matrices by converting this state vector to a density matrix:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}: \rho = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Next, we see a CNOT gate applied to qubits 0 and 1. The CNOT gate matrix representation is:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

We can apply the CNOT gate to the previous state by examining the state of both qubits.

$$q0 : \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} q1 : \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

q1 is still in the $|0\rangle$ state since no gates have been applied to it. The CNOT gate depends on the value of the control bit. If the control bit is in state $|1\rangle$, the target bit flips. If the control bit is in state $|0\rangle$, the target qubit remains unchanged. In order to combine the qubit states, we use the Kronecker product where each element of matrix A is multiplied by the entire matrix B element-wise.

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Applying the CNOT gate to the prior state, we get:

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\begin{bmatrix}1\\0\\1\\0\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\\0\\1\end{bmatrix}=\frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$$

Our expected result is $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$. To convert this to density matrix representation, we can start by taking the Kronecker product of each qubit state.

$$|00\rangle:\begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix}=\begin{bmatrix}1\\0\\0\\0\end{bmatrix}\ |11\rangle:\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}=\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

$$\frac{1}{\sqrt{2}}\left(\begin{bmatrix}1\\0\\0\\0\end{bmatrix}+\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right)=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\\0\\1\end{bmatrix}$$

We now convert this state vector to a density matrix:

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\\0\\1\end{bmatrix}*\frac{1}{\sqrt{2}}\begin{bmatrix}1&0&0&1\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&0&0&1\\0&0&0&0\\0&0&0&0\\1&0&0&1\end{bmatrix}$$

Our final result by using the density matrix approach is shown above. This result is correct because of the properties of quantum mechanics, and will thus serve as the basis for comparison for our approach.

3.2 Pauli Matrix and PGM Approach

We can now analyze the PGM shown in Figure 4. If we examine the CPT for q0 in moment 0, we can see how they represent the state:

$q0m0=$

I	X	Y	Z
$\frac{1}{2}$	0	0	$\frac{1}{2}$

$$=\frac{1}{2}I+\frac{1}{2}Z=\frac{1}{2}\begin{bmatrix}1&0\\0&1\end{bmatrix}+\frac{1}{2}\begin{bmatrix}1&0\\0&-1\end{bmatrix}=\begin{bmatrix}1&0\\0&0\end{bmatrix}$$

Next, in moment 1, we see qubit 0 undergo a Hadamard transformation. Our model represents a Hadamard transformation as follows:

$q0m0/q0m1$	I	X	Y	Z
I	1	0	0	0
X	0	0	0	1
Y	0	0	-1	0
Z	0	1	0	0

After running variable elimination and querying q0m1, our model returns the follow table for the events of the Hadamard transform on q0:

q0m1 =

I	X	Y	Z
$\frac{1}{2}$	$\frac{1}{2}$	0	0

$$=\frac{1}{2}I+\frac{1}{2}X=\frac{1}{2}\begin{bmatrix}1&0\\0&1\end{bmatrix}+\frac{1}{2}\begin{bmatrix}0&1\\1&0\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&1\\1&1\end{bmatrix}$$

Since the results agree, we can confidently say our model is valid for the Hadamard gate. We follow the same procedure for the final gate, the CNOT gate.

For our model, we applied the CNOT gate, ran variable elimination, and queried the output to get:

II	XX	-YY	ZZ
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

We can breakdown the above result by computing the states using the Pauli set. Above, we saw that the Hadamard gate puts qubit 0 into density matrix:

$$q0=\frac{1}{2}\begin{bmatrix}1&1\\1&1\end{bmatrix}=\frac{I+X}{2}$$

We also know the state of qubit 1 is $\begin{bmatrix}1\\0\end{bmatrix}$ which can be represented as a density matrix:

$$q1=\frac{1}{2}\begin{bmatrix}1&0\\0&0\end{bmatrix}=\frac{I+Z}{2}$$

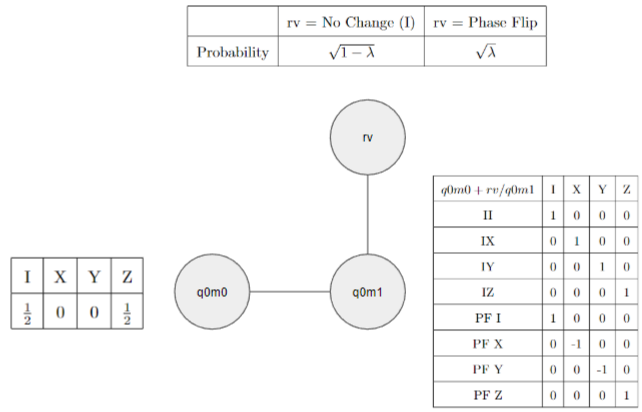


Figure 5: Network Model for Phase Flip Circuit with initial state $|0\rangle$

As we did previously, we can take the Kronecker product to combine these two qubits:

$$\frac{I+X}{2}\otimes\frac{I+Z}{2}=\frac{1}{4}[II+IZ+XI+XZ]$$

Now, applying the CNOT gate gives:

$$\frac{1}{4}[II+XX-YY+ZZ]$$

Converting this back to density matrix form gives us:

$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}+\begin{bmatrix}0&0&0&1\\0&0&1&0\\0&1&0&0\\1&0&0&0\end{bmatrix}-\begin{bmatrix}0&0&0&-1\\0&0&1&0\\0&1&0&0\\-1&0&0&0\end{bmatrix}+$$

$$\begin{bmatrix}1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&1\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&0&0&1\\0&0&0&0\\0&0&0&0\\1&0&0&1\end{bmatrix}$$

Comparing our final results, as well as the intermediate step results, we see that they are equivalent in their density matrix representations. This proves the validity of our model for the Bell state circuit. By following this procedure for many other complex circuits, we were able to come up with our results, and prove the efficiency and validity of our model.

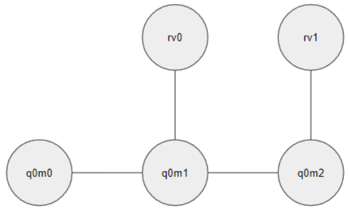
4 RESULTS

In order to verify the accuracy of our model, we subjected it to a range of quantum circuits, spanning from straightforward ideal circuits to intricate noisy circuits. By proving the accuracy of our model for each of these circuits, we can confidently say our model is accurate for quantum circuit simulation. Our experimental setup involved creating the quantum circuit, converting it to a network and generating density matrix outputs by hand, and finally verifying our results using pgmpy. Above, we proved the model for an ideal, noiseless circuit. Modern day simulations can handle these circuits very well, and the majority of our focus was on noisy circuits that currently can not be simulated very well.

4.1 Phase Flip Case Study

Our most significant results were in the area of noise simulation. We began by verifying the model for bit flip and phase flip events. Bit flip events cause random changes in state from $|0\rangle$ to $|1\rangle$ and vice versa. Phase flip events do not change the $|0\rangle$ state, but when applied to the $|1\rangle$ state, they apply a rotation of π radians around the Z-axis, flipping the sign of the state and changing its phase by π leaving us with $|1\rangle$. We can represent these events using Kraus operators. The Kraus operators for phase flip are:

$$E_0=\sqrt{p}I \qquad E_1=\sqrt{1-p}Z$$



	rv = No Change (I)	rv = Amplitude Damping
Probability	$\frac{1+\sqrt{1-\lambda}}{2}$	$\frac{1-\sqrt{1-\lambda}}{2}$

I	X	Y	Z
$\frac{1}{2}$	0	0	$\frac{1}{2}$

$q0m0 + rv/q0m1$	I	X	Y	Z
II	1	0	0	-1
IX	0	1	0	0
IY	0	0	1	0
IZ	0	0	0	1
AD I	1	0	0	1
AD X	0	-1	0	0
AD Y	0	0	-1	0
AD Z	0	0	0	-1

$q0m1 + rv/q0m2$	I	X	Y	Z
II	1	0	0	1
IX	0	1	0	0
IY	0	0	1	0
IZ	0	0	0	1
AD I	1	0	0	1
AD X	0	1	0	0
AD Y	0	0	1	0
AD Z	0	0	0	-1

Figure 6: Network Model for Amplitude Damping Circuit with initial state $|0\rangle$

We implement these Kraus operators using our CPTs to represent the impact on each Pauli state.

Both phase flip and bit flip can be very damaging to the accuracy of quantum information, and thus are extremely important to model in order for researchers to work on correcting them. Figure 5 shows the network model for a phase flip circuit. The rv node represents the probability of a flip event occurring, which is represented using λ in the table. In the table for q0m1, we combine the events of q0m0 and rv.

4.2 Amplitude Damping Case Study

After flip events, we proved the validity for damping noise, including phase damping and amplitude damping. Phase damping is one of the most common types of decoherence noise in quantum systems, and is caused by the loss of energy from a qubit to its environment. Similarly, amplitude damping occurs when the amplitude of a quantum state decreases overtime due to interaction with the environment. Both of these noise channels pose a significant challenge for quantum information processing, since it limits the ability to store and manipulate information in qubits. This exemplifies the need for an accurate model to simulate these noises.

The network model for phase damping is almost equivalent to phase flip, with the only difference being the probabilistic events that occur. This proves the versatility of our model, with only minor modifications needed to present varying, complex quantum circuits. A more complicated network was developed for amplitude damping. Unlike phase flip and phase damping where the application of the random variable representing the noise event was simple, in amplitude damping, the affects of the noise channel force us to split the application of noise over two moments. The network and associated CPTs are shown in Figure 6. In theory, the network should have one random variable with an edge to both q0m1 and q0m2, but for our application, we found that this model was not valid. Instead, we split up the random variable into rv0 and rv1, and apply it individually to each moment. The random variable probabilities are identical, and are thus represented using only one table. The network represents the Kraus operators for amplitude damping which are:

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{bmatrix} \qquad E_1 = \begin{bmatrix} 0 & \sqrt{\lambda} \\ 0 & 0 \end{bmatrix}$$

5 CONCLUSION AND FUTURE WORK

In this research work, we investigated the idea of using PGMs to encode quantum circuit information without loss of generality. We proved that our model is capable of accurately and efficiently simulating all quantum circuits, including complex, noisy circuits. The severe limitations of modern day quantum simulations tools lie in the fact that there is no unified language to explain all types of noise events, a lack of support for all types of noise, and the complexity of computation for each noisy circuit. These key lackings can all be addressed by our proposed model. Through the use of a visual representation and simple tabular amplitudes and probabilities, we simplify the previously complex language of representing noisy circuits. Furthermore, we support all types of noise events, as well as ideal circuits with no loss of coherence or information. We use inference algorithms like variable elimination to greatly simplify the complex calculations of noise events, and allow for a single operation on a mostly sparse matrix to get our result.

We validated our model using pgmpy and IBM Qiskit. For ideal circuits, Qiskit provided a valuable checking tool to compare our model results. pgmpy was used extensively as the PGM library to create and run our model, and ensure validity of our computations.

The implications of such a model are huge for researchers, as well as for simple quantum enthusiasts. By introducing a single, unified language, researchers can expedite the necessary research for error correction and quantum system isolation, leading to a brighter future for quantum computing.

There are still many unexplored avenues in this project that can expand the use of our single, unified language to broader applications. One key challenge that researchers face when working with noisy events is learning the noise parameters. The current process involves extensive simulation, leading to large overhead and complex calculations. Efficiently learning quantum noise is currently a topic of great research interest. Our model of using PGMs allows us access to algorithms like parameter learning which can greatly simplify the arduous process of learning noise parameters. Through parameter learning, our model will have the capability to receive quantum measurement data and accurately predict the likelihood of a noise event occurring. With the use of a single operation, our model can efficiently identify the probabilities of noise occurrences. This serves as a significant improvement in noise learning research, and is currently being worked on as part of this project. The long term scope of this visual language representation for quantum circuits will involve full development of key algorithms like parameter learning, as well further noise channel improvements like correlated noise events that no modern day quantum simulators are capable of processing. As Qiskit is a widely adopted tool for creating quantum circuits, we aim to enhance the usability of our model by streamlining the process of converting a Qiskit circuit to our framework. This enables researchers to work with a familiar tool while benefiting from the advanced features of our model. Furthermore, a key proposed advantage of this model is the improved efficiency of calculations. The use of a single operation over sparse matrices should allow for significant speed and overhead improvements, which will be key in the next generation of quantum computing, with the ever growing number of qubits in recent devices.

REFERENCES

[1] Ankur Ankan and Abinash Panda. 2015. pgmpy: Probabilistic graphical models using python. In *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*. Citeseer.

[2] Cirq Developers. 2022. *Cirq*. https://doi.org/10.5281/zenodo.7465577 See full list of authors on Github: https://github .com/quantumlib/Cirq/graphs/contributors.

[3] Daniel Gottesman. 1998. The Heisenberg Representation of Quantum Computers. arXiv:quant-ph/9807006 [quant-ph]

[4] Yipeng Huang, Steven Holtzen, Todd Millstein, Guy Van den Broeck, and Margaret Martonosi. 2021. Logical Abstractions for Noisy Variational Quantum Algorithm Simulation. arXiv:2103.17226 [quant-ph]

[5] Daphne Koller and Nir Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press.

[6] Jonathan Wei Zhong Lau, Kian Hwee Lim, Harshank Shrotriya, and Leong Chuan Kwek. 2022. NISQ computing: where are we and where do we go? *AAPPS Bulletin* 32, 1 (September 2022), 27. https://doi.org/10.1007/s43673-022-00058-z

[7] Michael A. Nielsen and Isaac L. Chuang. 2011. *Quantum Computation and Quantum Information: 10th Anniversary Edition* (10th ed.). Cambridge University Press, USA.

[8] Qiskit contributors. 2023. Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2573505