5-2021

Characterization and Bechmarking of Quantum Computers

Megan L. Dahlhauser
University of Tennessee, Knoxville, mlilly1@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Megan L. Dahlhauser entitled "Characterization and Bechmarking of Quantum Computers." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Energy Science and Engineering.

Travis S. Humble, Major Professor

We have read this dissertation and recommend its acceptance:

Cristian D. Batista, Raphael C. Pooser, Bruce J. MacLennan

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Characterization and Benchmarking of Quantum Computers

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Megan Lilly Dahlhauser

May 2021

Dedicated to Terry Lilly, Jane Lilly, and Thomas Dahlhauser

Acknowledgments

First and foremost, I would like to thank my advisor, Travis Humble. His wisdom, kindness, patience, and encouragement have forever shaped the way I approach science and life. I could not have asked for a better mentor, and I will be forever grateful for my time as his student.

Thanks also to Ian Hincks, Dar Dahlen, Egor Ospadov, Joseph Emerson, and Joel Wallman at Quantum Benchmark for their help in working with the True-Q software and understanding its capabilities, and to Erik Nielsen at Sandia National Laboratories for help in working with the pyGSTi software and getting experiments set up.

To my doctoral committee members, Cristian Batista, Bruce MacLennan, and Raphael Pooser, for teaching some of my absolute favorite classes and providing such helpful advice and feedback throughout my graduate school experience.

To Eugene Dumitrescu for being a wonderful mentor during my internships at Oak Ridge National Laboratory. With his guidance, I learned how to approach scientific research and gained so much experience in such a short time.

To Lori Davis, my calculus teacher for three of my four years at William Monroe High, for being a significant reason I was able to succeed in math and science.

To Michelle Halsted, Erica Grant, Paul Kairys, and Elizabeth Piersall for being on this graduate school journey with me.

"Modeling Noisy Quantum Circuits Using Experimental Characterization" authored by Megan L. Dahlhauser and Travis S. Humble and published by the American Physical Society in Physical Review A [1] (DOI: https://doi.org/10.1103/PhysRevA.103.042603) is

reproduced in Chapter 4 under the Creative Commons Attribution 4.0 International license (CC BY 4.0) [2]. Changes to the formatting have been made but the content is unchanged.

We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team.

Abstract

Quantum computers are a promising technology expected to provide substantial speedups to important computational problems, but modern quantum devices are imperfect and prone to noise. In order to program and debug quantum computers as well as monitor progress towards more advanced devices, we must characterize their dynamics and benchmark their performance. Characterization methods vary in measured quantities and computational requirements, and their accuracy in describing arbitrary quantum devices in an arbitrary context is not guaranteed. The leading techniques for characterization are based on fine-grain physical models that are typically accurate but computationally expensive. This raises the question of how to extend characterization efficiently to larger scales. We present an empirical-based approach to direct characterization of quantum circuits that reconciles accuracy with scalability by using a reduced set of test circuits that target a chosen application and coarse-graining the noise modeling process to reduce the model complexity. We show that this method performs well in tests with Greenberger-Horne-Zeilinger-state preparation circuits and the Bernstein-Vazirani algorithm, though it does not describe all error present in the system. We benchmark this method with the leading methods of gate set tomography, cycle benchmarking, and Pauli channel noise reconstruction to characterize quantum circuits and we compare the accuracy of these methods in predicting quantum device behavior. We find that our method for empirical direct characterization offers competitive accuracy when compared with finer-grained techniques, while significantly reducing the resources required for characterization. By testing on quantum devices, we quantify the quantum and classical resources required for each characterization method and we monitor the decrease in accuracy as a function of circuit size. We find that these characterization methods can provide an accurate estimate of a quantum computer's

performance on a benchmark but the best-performing method varied by test. Our results indicate that these characterization methods perform well in describing the noise of a quantum computer but their performance depends on the size and the context of the application.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Quantum computing has been heralded as a new form of computing capable of expanding our technological capabilities beyond the reach of classical computing. As classical computing reaches foretold limits, quantum computing is a potential alternative and enhancement for computational ability. Operating on the fundamental properties of quantum mechanics, quantum computers can harness unique characteristics such as superposition and entanglement to enable new computational operations. Yet as quantum computers emerge and gain use, the importance of characterization and benchmarking techniques to understand and evaluate these devices becomes ever more apparent.

Quantum computing has garnered attention largely because several algorithms have indicated that these devices could be used to find solutions to problems that evade even the most advanced modern technology. One of the earliest and most prominent examples is Shor's algorithm, developed in 1994 [9]. This algorithm demonstrates how a quantum computer may efficiently find the prime factors of any integer. There are no existing efficient classical algorithms for this problem, although it has not been proven that none can exist. But the presumed hardness of this problem has led to its inclusion in cryptographic schemes, such that finding the factors of an integer used in encryption would ultimately provide the means to decrypt encrypted messages. The discovery of an efficient means to factor integers using a quantum computer launched development of quantum computers into a global security concern over the safety of encryption techniques.

Several algorithms for performing chemical simulations on quantum computing hardware demonstrate that quantum computers may be able to efficiently solve many problems of relevance to chemistry. One such example is the electronic structure problem which is the study of the energy levels of a chemical system. Such problems are difficult for classical computers because every additional constituent of a chemical system grows the complexity of the wavefunction description exponentially [10]. Quantum computers, however, achieve polynomial scaling in this problem growth, and this can largely be attributed to the fact that a quantum computer is operated on the same fundamentals as the system in question– namely, quantum mechanics, a compelling idea first made famous by Richard Feynman in 1981 [11]. Chemistry simulations constitute a majority of the use of modern supercomputers, and speedups offered by quantum computers either as a separate device or hybrid solver can have a huge impact on computational capabilities.

Another application of quantum computing that has received great attention is that of machine learning. Machine learning is the process by which a computer attempts to "learn" a pattern or make deductions based on statistical methods applied to input data. Several quantum algorithms exist which indicate achievable speedups to machine learning using quantum computers. A quantum support vector machine, a technique which searches for a hyperplane that bisects data into two classifications with maximum margin of separation and probability of successful classification, could experience exponential speedup compared to classical implementations. Similarly, quantum principal component analysis, a method which reduces the dimensionality of data based on identifying the prominent trends and correlations of the data, could also experience exponential speedup [12].

These are some of the most prominent and promising examples of what we might use quantum computers to do. More examples exist and more are likely to be found as research continues to advance our understanding of the capabilities of quantum computing [13]. Our goal in pursuing quantum computing is nothing short of running these algorithms. The applications for which quantum computing offers us an advantage over classical computing are the reason to chase progress in quantum computing. Advantages gained from quantum computers will likely manifest by either making previously intractable problems accessible or in significantly reducing the resources required. This is particularly important in the realm of

computational energy consumption, for example, which is projected to outpace total global energy production within just a few decades [14].

Early examples of quantum computers have been emerging rapidly and undergoing even more rapid testing and experimentation. Made up of quantum bits (qubits), some prominent examples are those based on superconducting qubits and ion trap qubits, although several other types of qubits have found success as well. In experiments, quantum computers have successfully factored the integer 291311 [15], calculated the ground state of a water molecule [16], and implemented a quantum support vector machine on handwriting recognition [17]. While none of this challenges what classical computers can do, this early stage gives us a glimpse of what quantum computers might do someday.

This stage has come to be known as the "Noisy Intermediate-Scale Quantum" era, or NISQ [18]. Practically, the NISQ era is a hardware-focused stepping-stone on the path to quantum advantage. As the name suggests, the two major hallmarks of the NISQ era are noise and small size. Every example of a NISQ-era device, of which there are many, is susceptible to some level of noise and has a relatively small register of qubits. Some of the most prominent examples are those produced by IBM [19], Rigetti [20], IonQ [16], Google [21], and University of Innsbruck [22]. These devices have on the order of tens of qubits and noise that manifests as error rates that vary by several orders of magnitude per device and operation.

Understanding the dynamics of quantum computers and their ultimate capabilities necessitates gaining in-depth knowledge of the machines themselves. Developing appropriate metrics for quantum devices is therefore vital, especially in the NISQ era. The prevalence of noise, the rapid growth in device size, and the pressing question of what NISQ devices are capable of compel development of methods for understanding quantum devices at every level, from qubits to processors to networks. For example, a quantum computer using quantum error correction codes can operate at an arbitrarily low error rate so long as the error rates in the physical qubits are below a particular threshold defined by the codes in use. This property, referred to as fault tolerance, would of course be highly valuable for a quantum computer, but determining error thresholds and tracking progress requires knowledge of the error behaviors in quantum devices [23]. Characterization and benchmarking are crucial for

understanding the capabilities and challenges of NISQ devices and charting a path forward. Delineating the motivations and use cases for characterization and benchmarking methods provides the necessary foundation for this discussion, and Fig. 1.1 outlines these terms, which include characterization, verification, validation, benchmarking, and meta-metrics.

Characterization methods determine the effects of noise and control on a quantum system. This provides the information needed to model quantum circuits and includes data such as error rates and associated noise models. A primary use case of characterization methods is estimating noise models to describe behavior of quantum devices in experiment, but other use cases include error mitigation or low-level routing [24]. Characterization of quantum devices represents the lowest-level information about the device and is therefore a foundation upon which other metrics may be based.

Verification evaluates how well the controls perform a desired operation, typically up to a certain threshold of precision. Verification methods are used to confirm that an application circuit is designed correctly. An example of this is process fidelity, which measures how close an operation implemented in experiment is to the expected ideal operation. In practice, verification processes have a strong overlap with characterization techniques, since defining how well an operation is performed is clearly related to the error rates of the operation. However, the distinction between these two terms relates to the importance of defining metrics such as fidelity to evaluate the performance of low-level operations. While error rates inherently connect to a type of noise model, the measured process fidelity is more general as it indicates the overlap between the noisy and noiseless operations independent of a selected noise model.

Validation methods are used to confirm that an application implementation is executed as designed. Validation demonstrates that a quantum computer can be applied to a particular problem. The quantum supremacy experiment, an example problem that pits quantum and classical computers against each other, is one such example [21]. This experiment confirmed that a quantum computer could successfully execute a clearly defined application circuit.

Benchmarking methods evaluate performance metrics that are defined for specified conditions. In particular, benchmarks measure the performance of a quantum computer on a selected application in such a way that the resultant metric may be used to compare

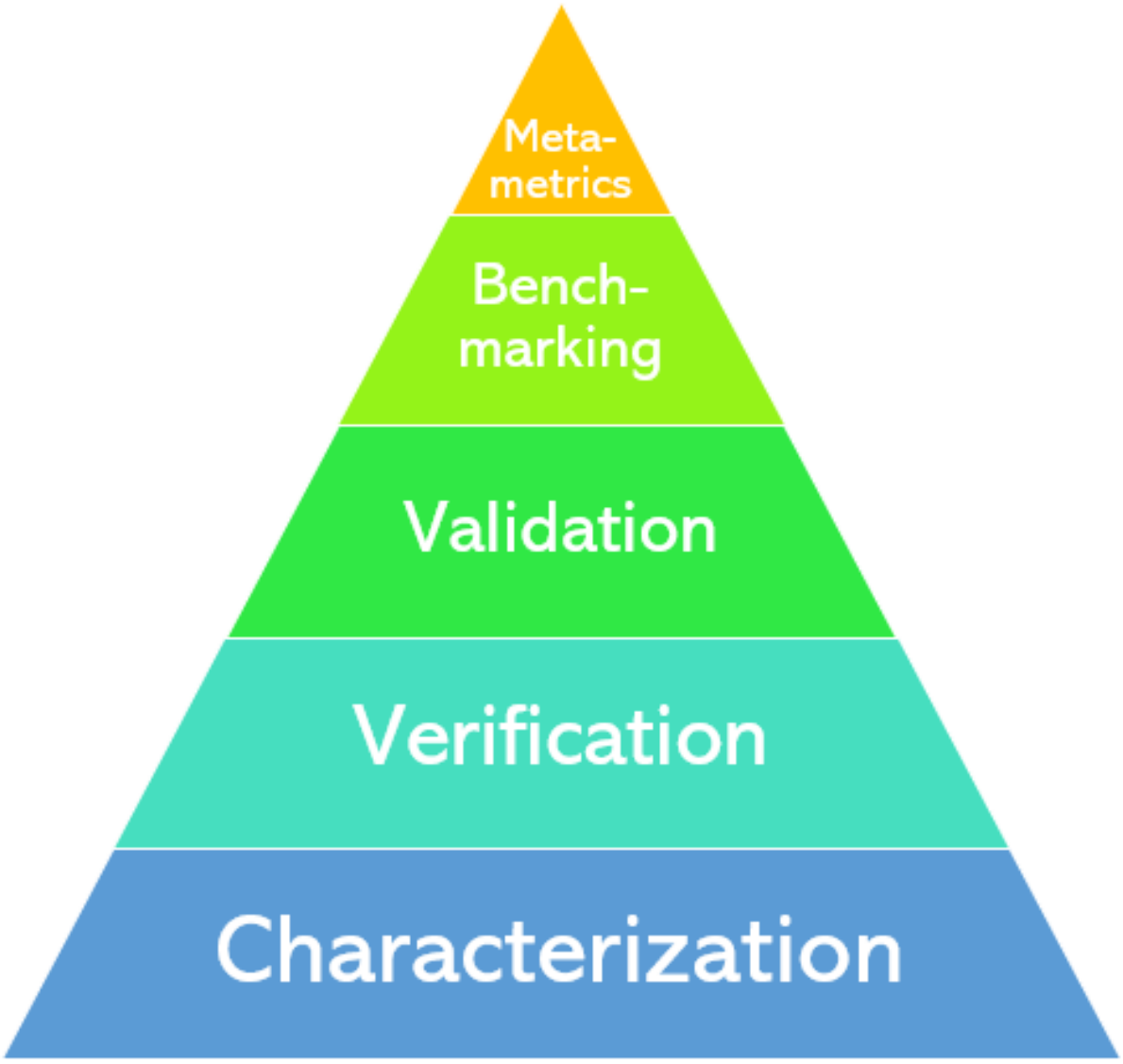

Figure 1.1: Different methods of evaluating the efficacy and performance of quantum devices, from the lowest to highest levels of information. For example, characterization methods provide information about the error rates of gates acting on individual qubits in a quantum register, while benchmarks represent high-level performance tests.

different devices' performance in a meaningful way. Analyzing the solution quality of an algorithm executed on a quantum computer can be considered a performance benchmark. Benchmarking methods may seem to overlap with validation methods. However, while benchmarks may be used to evaluate performance across quantum computing schemes, validation techniques confirm that a quantum computing instance executed a performance test as designed.

Lastly, we also consider "meta-metrics" to evaluate the performance of different examples of all of these metrics. The complexity and nascence of quantum computing at this stage warrant consideration of the efficacy of the characterization, verification, validation, and benchmarking techniques that we develop and use. For example, measurements of the efficiency or accuracy of different methods can be considered meta-metrics.

We focus on the evaluation of characterization techniques for quantum computers. We assess the effectiveness and accuracy of various methods of characterization by applying these methods in experiment on quantum computers. We analyze the results by comparing against simulated and expected results and report metrics based on a well-defined use case. We select as our use case the estimation of noise models which describe the behavior of the quantum device in experiment. In addition, we establish a direct connection between the characterization results and performance on a chosen benchmark and evaluate metrics of these techniques, such as scalability and accuracy. From these metrics, we compare characterization methods, evaluate their performance, and draw conclusions about their use in the NISQ era and beyond.

Chapter 2

Fundamentals

Quantum computers are made of quantum bits called qubits, which may be in any state described by

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{2.1}$$

where the basis states $|0\rangle$ and $|1\rangle$ represent two levels of a quantum system and the coefficients α and β represent the amplitudes, or probability distribution ($|\alpha|^2$ and $|\beta|^2$), of those two states. While in classical computing it is possible and often necessary to inspect the value of a particular bit, it is a principle of quantum mechanics that it is not possible to exactly determine the values of α and β by measurement. In general, measurement of quantum states is described by a set of measurement operators $\{M_m\}$ where the probability of result m is

$$p_m = \langle\psi|\, M_m^\dagger M_m\,|\psi\rangle \tag{2.2}$$

and the state after measurement is

$$\frac{1}{\sqrt{p_m}} M_m\,|\psi\rangle \tag{2.3}$$

In the quantum computing context, measurements are often projection operators in the computational basis, i.e. $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$. When a qubit in state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ is measured in this basis the result will be either 0 with a probability of $|\alpha|^2$ or 1 with probability $|\beta|^2$. Additionally, this measurement changes the quantum state–if the

result is 0, the qubit's new collapsed state is effectively $|0\rangle$, meaning it can be modeled as a classical bit.

Measurements described by measurement operators M_m can be defined as a set of operators E_m such that

$$E_m = M_m^\dagger M_m \tag{2.4}$$

and these operators form a set $\{E_m\}$ which is called a POVM for Positive Operator-Valued Measure. POVMs are a special case of the general definition of quantum measurement given by Eqs. 2.2 and 2.3. This formalism is often used in analysis of the probabilities of measurement outcomes in instances where the state after measurement is not needed or known. This is because POVMs provide the mathematical means to measure an ensemble of quantum states and correctly distinguish among those states some of the time.

To express quantum states of multiple qubits, the computational basis must increase. A two-qubit state may be described as

$$|\psi_{ab}\rangle = \alpha_{00}\,|0_a 0_b\rangle + \alpha_{01}\,|0_a 1_b\rangle + \alpha_{10}\,|1_a 0_b\rangle + \alpha_{11}\,|1_a 1_b\rangle \tag{2.5}$$

for qubits labeled a and b where the square of each α coefficient gives the probability of obtaining the associated state result from measurement. The number of amplitudes needed to describe the quantum state scales as 2^n, exponential in the number of qubits n. The number of amplitudes needed to express the state of 270 qubits would be more than the number of particles in the observable universe, yet these states happen in nature all the time. These enormous amounts of data are constantly calculated and evolved through time with incredible speed and perfect precision. Harnessing this natural computational power is part of the tantalizing promise of quantum computing [23].

These $|0\rangle$ and $|1\rangle$ basis vectors may be written as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \; |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.6}$$

and we may write the quantum state $|\psi\rangle$ as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{2.7}$$

Just as the qubit is the quantum analogue to the classical bit, quantum gates are analogues to classical logic gates. It is a postulate of quantum mechanics that the evolution of a quantum state is described by unitary operators such as the Pauli operators X, Y, and Z,

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},\ Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.8}$$

which we use to express quantum gates in quantum computing. For example, the X gate would act on a single qubit quantum state as

$$X\,|\psi\rangle = X\big(\alpha\,|0\rangle + \beta\,|1\rangle\big) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} = \alpha\,|1\rangle + \beta\,|0\rangle \tag{2.9}$$

which is analogous to the classical NOT gate, flipping 0 to 1 and 1 to 0. Because quantum states are represented by linear combinations of basis states, quantum gates can also affect the relative phase, such as the Z gate which performs the following operation on a single qubit quantum state:

$$Z\,|\psi\rangle = Z\big(\alpha\,|0\rangle + \beta\,|1\rangle\big) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix} = \alpha\,|0\rangle - \beta\,|1\rangle \tag{2.10}$$

This operation does not have a classical analogue. Also unlike classical computation, quantum computing is inherently reversible. Any unitary operator has an inverse that is also a unitary operator, and therefore every quantum gate can be 'undone' by another quantum gate.

Multi-qubit states and gates can be expressed by vectors and matrices of larger dimensions. For example, we may write the two-qubit state

$$|\psi_{ab}\rangle = \alpha_{00}\,|0_a 0_b\rangle + \alpha_{01}\,|0_a 1_b\rangle + \alpha_{10}\,|1_a 0_b\rangle + \alpha_{11}\,|1_a 1_b\rangle = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} \tag{2.11}$$

where we have taken

$$|0_a 0_b\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},\ |0_a 1_b\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},\ |1_a 0_b\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix},\ |1_a 1_b\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{2.12}$$

Multi-qubit operations are then matrices acting on these state expressions. For example, a controlled-X gate, which is a common two-qubit gate also called a controlled-NOT (CNOT) gate, operates as

$$\textsc{cnot}\big(\,|\psi_{ab}\rangle\,\big) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{11} \\ \alpha_{10} \end{pmatrix} \tag{2.13}$$

The CNOT gate has the effect of flipping the state of qubit b if the state of qubit a is $|1\rangle$, and has no effect if the state of qubit a is $|0\rangle$. Qubit a is labeled the control qubit and qubit b is labeled the target qubit.

In come cases, multi-qubit states are inseparable, i.e. $|\psi_{ab}\rangle \neq |\psi_a\rangle\,|\psi_b\rangle$. One example is the state $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. States which cannot be expressed as a product of their component states are called entangled, and the entanglement property is unique to quantum mechanical systems.

Qubits and gates interact in quantum circuits, which we may pictorially represent in diagrams similar to the one in Figure 2.1. Qubits are typically labelled as q_n for $n = 0$ to size n of the quantum register, which is generally the size of the physical quantum processor. Qubits are operated on by quantum operators represented as icons on a quantum "wire". Each wire represents the evolution of the qubit state through the circuit until final measurement produces a classical bitstring.

The language of quantum states, gates, and circuits implies that quantum computing is constrained to perfectly understood mathematical formalisms, but typically this is not the case due to the presence of noise. Noise in quantum computers can originate from sources which generally vary with the type of hardware but can include improper control of the device, thermal activity, or qubit instability [23]. Regardless of the source, noise generates errors which can be calamitous for computation, and remains the biggest overall threat to quantum computing development.

Errors are not a concept unique to quantum computing. In the early days of classical computing, scientists and engineers were convinced that error correction would be absolutely necessary when devices reached a number of bits on the order of billions, trillions, or far more, as we have now. After all, even an incredibly low error rate of 1 in a trillion can become catastrophic for computers made of many times that number of bits. But surprisingly, this fear never came to fruition as manufacture of classical hardware proved to be far more reliable and controllable than originally expected [23].

In quantum computing, we may yet reach such a stage of highly reliable and controllable hardware such that error correction and noise mitigation are not needed for effective computing. But based on current knowledge and experiment we are a long way from that, even if it is achievable. For one, qubits are naturally more delicate and prone to error by design than their classical counterparts. In order to take advantage of quantum entanglement, a key aspect for quantum computation, qubits must be easily entangled. Yet that characteristic in turn makes it likely for qubits to become entangled in undesirable ways such as with other qubits or with the qubits' environment, which creates uncontrolled and unanticipated behaviors in the computer. Additionally, it is more difficult to detect quantum errors than classical errors because probing the quantum state may cause it to

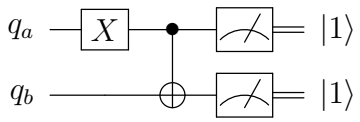

Figure 2.1: An example of a quantum circuit diagram. Qubits a and b, labeled q_a and q_b, respectively, are initialized to state $|0\rangle$ and operated on by X_a and $\textsc{cnot}_{a\to b}$. The result of measurement (without the presence of noise) is shown with a double line, indicating "classical wires" in contrast to the single line "quantum wires" connecting the qubit operations in the circuit.

collapse. Furthermore, because classical computing consists of encoding into states 0 and 1 the only observable errors are bit flips, where a 0 is incorrectly reported as 1 and vice versa. Errors in quantum computers are more complicated, since they may impact not just the states $|0\rangle$ and $|1\rangle$ but also their coefficients α and β, leading to a virtually infinite number of ways an error may affect the state of a qubit. Moreover, noise in quantum systems may be Markovian, which is uncorrelated and stochastic by nature, or non-Markovian, which might be correlated, temporally dependent, or influenced by the circuit context [25].

The language used to describe the impact of noise in a quantum system is often density matrices. This formulation is similar mathematically to the state vector formulation used in Eq. 2.1, but allows for more flexibility in certain situations. For example, the density matrix formulation is a natural expression for both pure and mixed states. As with Eq. 2.1, pure states are known exactly, and under the density matrix formalism Eq. 2.1 can be written as

$$\rho = |\Psi\rangle\langle\Psi| = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \begin{pmatrix} \alpha^* & \beta^* \end{pmatrix} = \begin{pmatrix} \alpha\alpha^* & \alpha\beta^* \\ \beta\alpha^* & \beta\beta^* \end{pmatrix} \tag{2.14}$$

Mixed states describe a combination of pure states in an ensemble, which can be defined as the density matrix

$$\rho \equiv \sum_i p_i |\psi_i\rangle\langle\psi_i| \tag{2.15}$$

for each state $|\psi_i\rangle$ which the quantum system is in with respective probability p_i. With the density matrix formalism quantum operations are defined as

$$\rho' = \varepsilon(\rho) \tag{2.16}$$

where the operation ε can be operations such as unitary transformations or measurements. For example, applying an X operator to the state ρ is expressed as

$$\varepsilon_X(\rho) = X\rho X^\dagger \tag{2.17}$$

Density matrices are particularly useful in the discussion of quantum systems and their environments and therefore typically convenient for investigating the effects of noise. Closed

systems, i.e. quantum systems which are isolated from their surrounding environment, may be described with unitary transforms while open systems, i.e. quantum systems comprised of a principle system interacting with its environment, generally cannot. Specifically, for a state ρ which is coupled to its environment while operated on by some transformation U, it cannot be assumed that the final state $\varepsilon(\rho)$ relates to the original state ρ by a unitary transformation. Rather, the reduced state of the system without the environment may be expressed by

$$\varepsilon(\rho) = \mathrm{Tr}_{env}[U(\rho \otimes \rho_{env})U^{\dagger}] \tag{2.18}$$

This expression assumes that the state ρ and its environment are separable but generally this is not the case. However, this picture is still useful in many scenarios practical to quantum computing.

The operator-sum representation is a mathematically convenient way to describe quantum operations on open systems. This representation rewrites Eq. 2.18 as

$$\varepsilon(\rho) = \sum_k \langle e_k | U \left[\rho \otimes |e_0\rangle \langle e_0| \right] U^{\dagger} |e_k\rangle \tag{2.19}$$

$$= \sum_k E_k \rho E_k^{\dagger} \tag{2.20}$$

where the operator elements $E_k = \langle e_k | U | e_0 \rangle$ operate on the state space of the principle system. These elements must satisfy

$$\sum_k E_k^{\dagger} E_k = I \tag{2.21}$$

which is true for all trace-preserving quantum operations [23].

Chapter 3

Characterization Methods

Several different characterization techniques for quantum computers have been developed to address varying needs. The primary driving needs in the characterization space are the information supplied by the protocol and the efficiency of the protocol in the quantum computational characterization problem.

Currently, quantum computing characterization protocols can be divided into three tiers delineated by characterization goal and acquired information [26]. The first tier provides the most information about the quantum processor and can be used to develop a highly detailed model, with a tradeoff in resource consumption and scalability. This includes tomography processes (quantum state [27], quantum process [28], gate set [29], randomized benchmarking [30]), robust phase estimation [31], and Hamiltonian estimation [32]. The second tier represents protocols which focus on developing descriptions of certain error rates to reduce resource consumption and improve scalability while sacrificing some amount of information gained about the system. This includes coarse-graining techniques such as direct fidelity estimation [33], randomized benchmarking [34], and cycle benchmarking [6]. The final tier is that which focuses solely on quantum processor performance, and does not attempt to make statements about the low-level operations. This includes more holistic characterizations such as quantum volume [35], volumetric benchmarks [36], and the use of specific quantum circuits or applications as estimations of performance [37].

The suite of characterization methods for quantum devices provides a spectrum of tradeoffs in information gain versus scalability of resource consumption, an example of

which is shown in Fig. 3.1. Scalability is becoming a determining factor for the choice of characterization techniques. Because tomographic techniques can provide a full description of a quantum operation, these methods would render any other method unnecessary if not for their inefficiency in characterizing more than a handful of qubits at a time. As quantum hardware rapidly increases in size well beyond handfuls of qubits, we are reaching regimes which require scalable methods for characterization.

We select three protocols for characterization: gate set tomography, cycle benchmarking, and empirical direct characterization. These three methods have commonalities that we take advantage of in evaluating performance. Gate set tomography (GST) and cycle benchmarking (CB) have similar goals, i.e. they are both focused on characterizing a quantum process or operation. Empirical direct characterization (EDC) also intersects with this goal, in estimating error rates for selected quantum gates. Moreover, all three methods share a common language. From GST, we calculate the Pauli transfer matrix of a process, and from this we extract information about the effective noise channels in the process as well as its fidelity compared to the expected operation. Cycle benchmarking yields the process fidelity, and we pair this with noise reconstruction (NR) protocols to produce effective Pauli noise channels of the process [38]. Finally, using EDC we calculate estimates for selected noise models tuned to experiment, which we select to match the noise models generated by CB and GST to create a lateral comparison. To estimate the fidelity from EDC results, we use the noise models in simulation and calculate an approximate fidelity for an operation or subcircuit. Tables 3.1 and 3.2 summarize the primary characteristics of each selected method.

Process fidelity and Pauli noise models are a common language among GST, CB, and EDC and a basis for a benchmark of these benchmarks. We use protocols to estimate these metrics and evaluate their performance with designed tests. To evaluate how accurate each method is in characterizing the selected processes, we use estimated noise models in simulation of the same process and compare simulation results to experiment results via total variation distance.

To evaluate if these characterizations are capturing the fundamental behavior of the device, we also test their accuracy in predicting results on new applications. We first

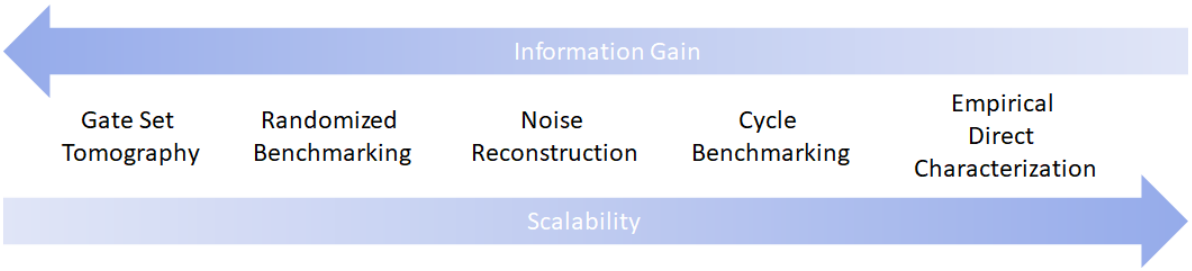

Figure 3.1: Relationship among different characterization methods. In general, the more information gained, the less scalable the method. This tradeoff is typically balanced based on the experiment parameters. However, as quantum processors increase in size, scalability will dominate as the primary motivation for characterization methodology selection and development. This compels probing the accuracy and effectiveness of the most scalable methods and continuing to develop highly efficient protocols for characterization.

Table 3.1: Summary of the pros and cons of select characterization methods.

Method	**Strengths**	**Weaknesses**
GST	Less sensitive to state preparation and measurement errors Yields process matrix	Scales exponentially with register size
CB/NR	Robust to state preparation and measurement errors Scales approximately independently of register size	Yields single metric (process fidelity)
EDC	Efficient in quantum experiments	Yields single metric (error rate)

Table 3.2: Summary of the primary properties of each selected method for characterizing quantum devices. Each method demonstrably characterizes at least as many qubits listed in the "Characterization" column.

Method	**Characterization**	**Quantum Resources**	**Output**
GST	1-2 qubits	Scales exponentially in qubit count	Complete process matrix
CB/NR	1-20 qubits	Scales with randomization parameters	Process fidelity Pauli noise channels
EDC	1-27 qubits	Variable scalability depending on application	Noise model of selected noise channels

select a performance test, e.g. a quantum algorithm that is not part of the characterization methodology. To evaluate noise model estimates, we input the composite noise model of each protocol into simulation and produce simulated results for our selected test. The closeness of the simulated results to experimental results as measured by total variation distance (TVD) defined in Chapter 4 by Eq. 4.1 demonstrates the protocol's success in predicting device behavior. We do a similar test to evaluate the process fidelity metric. By running our performance test in experiment, we estimate the observed fidelity in experiment and compare to the composite process fidelity estimated by our characterization protocols.

3.1 Gate Set Tomography

Gate set tomography (GST) is a method for extracting quantitative and qualitative information about quantum gates implemented in a quantum computer [39, 29]. It arose as an extension of quantum process tomography (QPT) [40, 41].

Quantum process tomography characterizes a quantum gate by generating an estimate of the process matrix or the Pauli transfer matrix by measuring the components of a quantum gate operating on a prepared quantum state. The QPT protocol assumes that the quantum state preparation and the measurement are either known or error-free. However, this is generally not the case in experiment, because state preparation and measurement (SPAM) errors are prevalent in many, if not all quantum processing units (QPUs) to date. Furthermore, in practice SPAM errors can often be the result of QPU components that QPT would be used to characterize. Because of this, QPT can be inaccurate in realistic quantum computing experiments. In particular, QPT can actually become less accurate as the gates improve [40].

Gate set tomography rectifies this self-consistency problem by defining and characterizing a set of gates that represents both the quantum gates of interest and the imperfect state preparation and measurement operations. By characterizing the full set of gates at once, GST is able to more accurately estimate the true quantum gates because SPAM operations are characterized explicitly.

Despite requiring more quantum experiments to gather the necessary information to perform GST than quantum process tomography, the lessened sensitivity to SPAM errors is expected to be vital for understanding how to utilize quantum error correction on near-term devices. The degradation of QPT gate characterization results due to the influence of SPAM can be highly problematic. This is particularly true for determining fault-tolerance thresholds, which have stricter conditions on gate error than on SPAM error. Quantum process tomography is unlikely to give accurate threshold estimates when SPAM error is highly prevalent compared to gate error [29].

Gate set tomography completely characterizes

$$\mathcal{G} = \{|\rho\rangle\rangle, \langle\langle E|, G_0, ..., G_k\} \tag{3.1}$$

where $|\rho\rangle\rangle$ represents the initial state, $\langle\langle E|$ is a POVM, and each G_k is a quantum gate. The set $\mathcal{F} = \{F_1, ..., F_N\}$ is defined as the SPAM gates which operate as $|\rho_j\rangle\rangle = F_j|\rho\rangle\rangle$ and $\langle\langle E_i| = \langle\langle E|F_i$. Every F_n must be composed of gates from gate set $\mathcal{G}$; therefore the set $\mathcal{G}$ must include gates sufficient to compose the full set of states and measurements. One example of such a gate set is $\mathcal{G} = \{\{\}, X_{\pi/2}, Y_{\pi/2}, X_\pi\}$ with $\mathcal{F} = \mathcal{G}$ which includes the empty gate $\{\}$. Each gate G_k can be reconstructed by measuring $\langle\langle E_i|G_k|\rho_j\rangle\rangle$. The GST protocol will characterize the full set $\mathcal{G}$ at once and only requires one initial state ρ and one final measurement E.

The GST algorithm for one qubit is as follows [29]:

1. Initialize to state $|\rho\rangle\rangle$
2. For some i, j, k of $i, j \in \{1...N\}, k \in \{0...K\}$ apply gate sequence $F_i \circ G_k \circ F_j$
3. Measure POVM E which must be a positive semidefinite Hermitian operator with $I - E$ also positive semidefinite
4. Repeat steps 1-3 a large number of times n and per execution r record $n_r = 1$ if measurement is success or $n_r = 0$ if failure
5. Average the results of step 4 to get $m_{ijk} = \sum_{r=1}^{n} \frac{n_r}{n}$ which is a measurement of expectation value $p_{ijk} = \langle\langle E|F_i G_k F_j|\rho\rangle\rangle$

6. Repeat steps 1-5 for all i, j, k

7. Optional for additional independent measurements: repeat steps 1-5 to measure expectation values $p_i = \langle\langle E|F_i|\rho\rangle\rangle$

3.2 Randomized Compiling

Randomized compiling is a method of transforming quantum circuits into a set of logically equivalent circuits by utilizing randomly selected twirling operators [4]. First, a quantum circuit is expressed in cycles, which are each a single time step of parallelized quantum operators within the circuit with no more than one operation per qubit. These cycles are decomposed into "easy" gates which are assumed to have low or negligible error rates and "hard" gates which are assumed to have high error rates. Twirling operators are then injected around the hard gates which have the effect of tailoring the noise in the system to a stochastic Pauli channel. The injected twirling gates must be easy gates, and these are compiled together with the other easy gates in the cycle such that they become a single round of easy gates.

Randomized compiling can be used with a variety of twirling methods. We use Pauli twirling, which is one of the most commonly used twirling techniques. Pauli twirling is a method which turns a quantum operator into a Pauli channel,

$$T_P(\varepsilon(\rho)) = \frac{1}{|\mathrm{P}|}\sum_{P\in\mathrm{P}} P\varepsilon(\rho)P^\dagger = \sum_{P\in\mathrm{P}} c_P P\rho P^\dagger \tag{3.2}$$

where P is a Pauli matrix from the set P and the coefficients c_P define the probability distribution over the Pauli operators. The set P is defined as the Pauli matrices $P^{\otimes n}$ for n number of qubits in the system. The sampling set therefore grows exponentially in the register size, so for systems with large n we may instead use randomized twirling,

$$T_P(\varepsilon(\rho)) = \frac{1}{N}\sum_{n=1}^{N} P_n\rho P_n^\dagger \tag{3.3}$$

where we select a limit of N operators from which to sample. In the limit of the highest possible value of $N = 4^n$, randomized Pauli twirling becomes equal to Eq. (3.2) [42].

Randomized compiling is implemented by adding gates from the twirling group, which in our case are any Pauli gates from Eq. 2.8 and the corresponding correction operator such that the overall unitary of the circuit is preserved. These added gates are compiled with neighboring easy gates which reduces the impact of randomized compiling on the circuit depth. This process is shown in Fig. 3.2. The final output of randomized compiling is a set of quantum circuits with randomly applied operators. The results of a randomly compiled quantum circuit are taken as the sum of the results over the set of these circuits.

Pauli twirling has been used in several different contexts in quantum computing, from experiment reduction in characterization protocols to enhancement of computer performance [42]. In randomized compiling, its purpose is to average the errors in the gate implementations into a stochastic Pauli channel. This has several benefits. Stochastic Pauli channels are more predictable and stable than other types of error such as coherent errors or spatial correlations among quantum components. By averaging the effects of these types of errors into a stochastic Pauli noise channel, we can estimate a description of the noise that is less complex than that of the uncompiled circuit. Randomized compiling is also expected to suppress error overall in the final results of compiled quantum circuits, at least in certain error regimes. For instance, average gate error is reduced in the case of over-rotation noise per gate with a factor of 10^{-2} difference in infidelity between easy and hard gates [4].

3.3 Cycle Benchmarking

Cycle benchmarking (CB) is a methodology for estimating the average process fidelity of a specified quantum process [6]. It is based on defining a circuit in terms of cycles and utilizes Pauli twirling to translate noise in a quantum system into a stochastic Pauli channel, as in randomized compiling [4]. In CB, the benefit of Pauli twirling is to take advantage of this uniform language by estimating the process fidelity under this common noise channel, leading to the ability to benchmark progress and devices. Cycle benchmarking and randomized

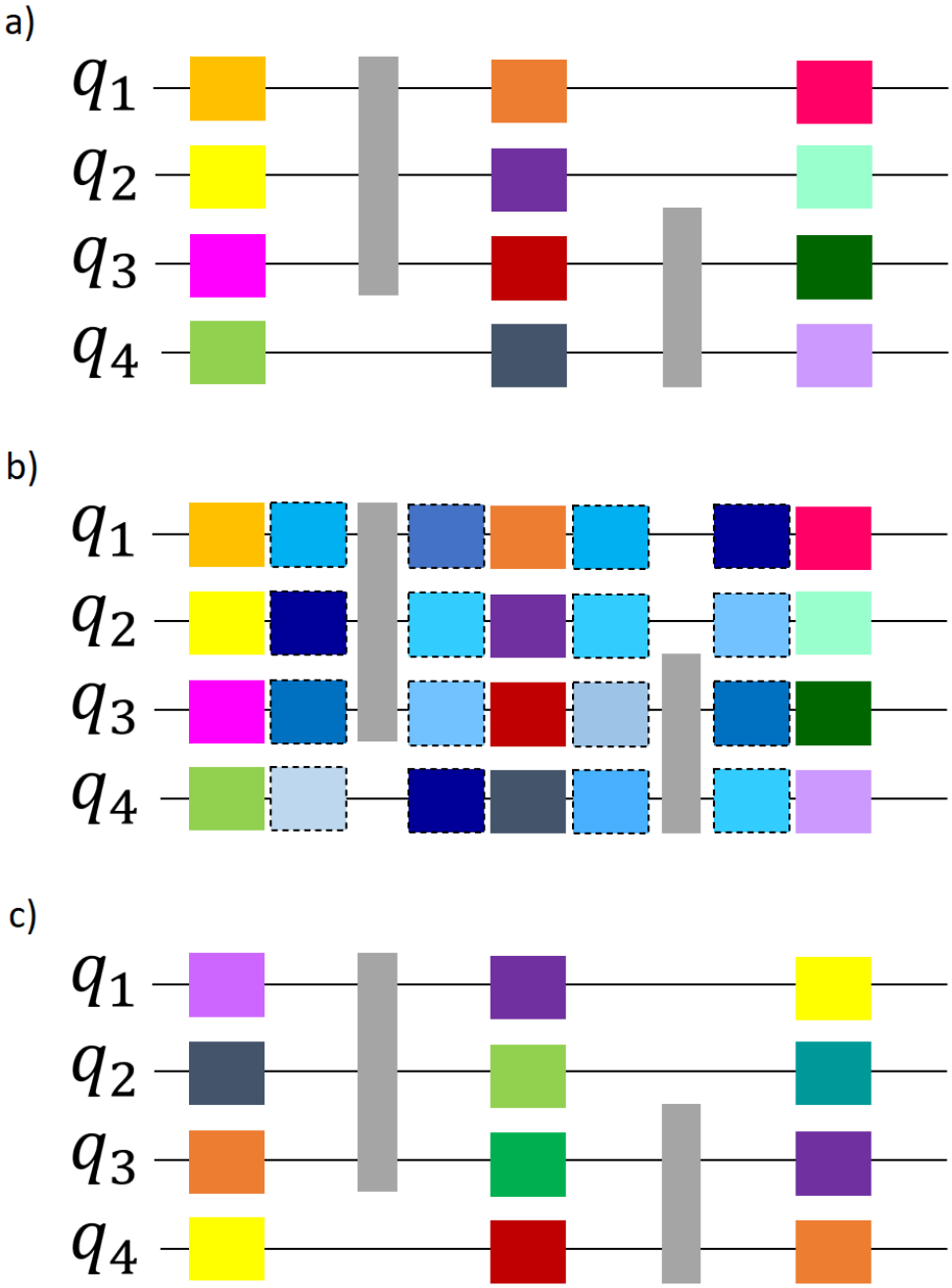

Figure 3.2: Graphical representation of randomly compiling a quantum circuit [4, 5]. Colored boxes represent easy gates; grey multi-qubit gates are considered to be hard gates in this example. Starting with a quantum circuit in a) we inject twirling gates in b) which are depicted by blue squares with dashed lines. Then in c) these gates are compiled together to form a randomly compiled circuit. This process is repeated for a set of randomly selected twirling gates to generate a set of n twirled quantum circuits which together represent the randomly compiled quantum circuit.

compiling are inextricably linked–CB provides a characterization of quantum processes in the context of randomized compiling.

Pauli twirling is used in CB to randomize a cycle within a circuit to "twirl" noise effects into a stochastic Pauli channel. In particular, for a process G in a target circuit, we can apply Pauli twirling to the noisy process $\tilde{G}$ such that the implementation becomes

$$T_P(\tilde{G}(\rho)) = \frac{1}{N}\sum_{n=1}^{N} P_n \tilde{G}(\rho) P_n^\dagger \tag{3.4}$$

In practice, errors occur in the state preparation and the measurement of the final state. To make CB robust to SPAM, it incorporates multiple rounds of randomized Pauli sequences and applications of the process $\tilde{G}$ and then extracts the overall process fidelity of G from the decay of the fidelity as a function of the number of rounds of application.

Cycle benchmarking is implemented as follows [6]. For a noisy quantum process $\tilde{G}$, we first select a set of n-qubit Pauli matrices which we define as set P. We also select two lengths m_1 and m_2 which define how many applications or "cycles" we will use to benchmark $\tilde{G}$. Because we do not want to change the logical output of the process $\tilde{G}$, we constrain m_1 and m_2 to values which satisfy $\tilde{G}^{m_1} = \tilde{G}^{m_2} = I$. Additionally, we define a length L which determines the number of random sequences per P in P. Then for each Pauli string in P, length m, and $l \in \{1, ..., L\}$ we perform the following procedure:

1. Select $m + 1$ random Pauli cycles which are injected to create a randomized circuit. The circuit can be expressed in three segments: a basis change to the selected Pauli string P in P, m cycles of alternating random Paulis and applications of the process $\tilde{G}$ (starting and ending with a round of Paulis), and a basis change back which returns to the original logical state.

2. Calculate the ideal (noiseless) outcome of the sequence

3. Implement the circuit in experiment and collect results

4. Calculate the overlap between the experiment results $\tilde{C}(P)$ and the ideal outcome $C(P)$ according to the expression:

$$f_{P,m,l} = Tr[C(P)\tilde{C}(P)] \tag{3.5}$$

5. Report the final composite process fidelity defined as:

$$F(\tilde{G}, G) = \sum_{P \in \mathrm{P}} \frac{1}{|\mathrm{P}|} \left(\frac{\sum_{l=1}^{L} f_{P,m_2,l}}{\sum_{l=1}^{L} f_{P,m_1,l}} \right)^{\frac{1}{m_2 - m_1}} \tag{3.6}$$

Figure 3.3 [6] demonstrates this procedure.

3.4 Noise Reconstruction

In many ways, CB is an extension of randomized benchmarking. Noise reconstruction, in turn, is an extension of CB. The generalization of NR therefore encompasses the basic algorithms of CB, randomized benchmarking, and other related protocols.

Noise reconstruction is a protocol which enables estimation of process fidelities along with the associated error probabilities. It stems from the relationship between Pauli fidelities f_i which measure how susceptible the Pauli operator P_i is to noise,

$$f_i = 2^{-n}\mathrm{Tr}(P_i \varepsilon(P_i)) \tag{3.7}$$

and the Pauli channel expression of Pauli error rates p_i which express the likelihood of the occurrence of a Pauli operator P_i as an error on state ρ,

$$\varepsilon(\rho) = \sum_i p_i P_i \rho P_i \tag{3.8}$$

These two metrics–Pauli fidelities and Pauli error rates–are related via the Walsh-Hadamard transform

$$\boldsymbol{f_G} = \boldsymbol{W_{G,\mathrm{P}^n}} \boldsymbol{p} \tag{3.9}$$

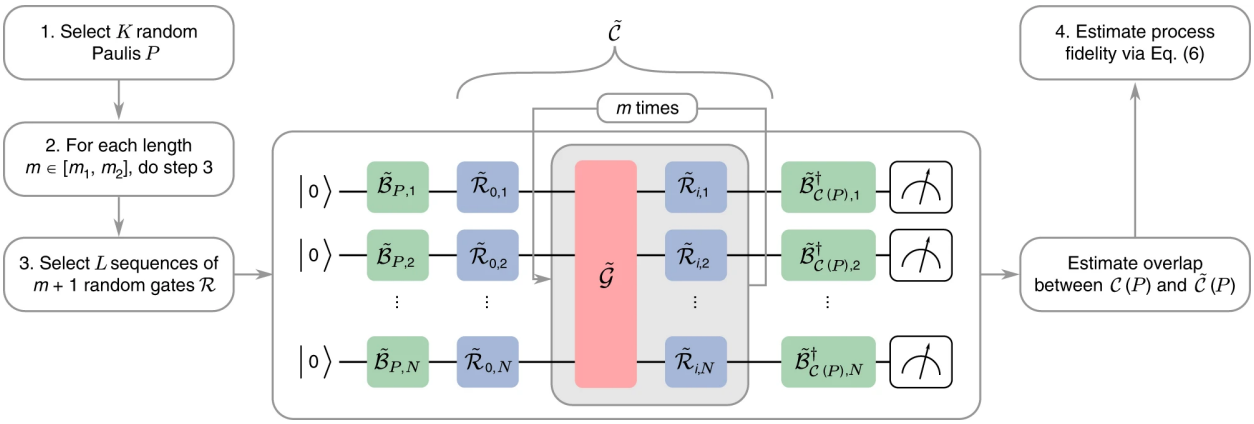

Figure 3.3: Graphical representation of the CB process. The green gates represent the change of basis defined by the Paulis P, blue gates represent the randomized Pauli twirling gates, and the red $\tilde{\mathcal{G}}$ gate represents the cycle of interest. This figure is reproduced with no changes from Erhard et al. [6] under the Creative Commons Attribution 4.0 International license [2].

for a group of Paulis G, where $\boldsymbol{f_G}$ is the vector representation of the fidelities f_g for elements $g \in G$ and $\boldsymbol{p}$ is the vector of Pauli error rates. The transform $\boldsymbol{W_{G,\mathrm{P}^n}}$ maps from group P^n to G, where P^n is the quotient group of Paulis with its center. The columns of this transform that correspond to Paulis that differ only by an element that commutes with all g are interchangeable, and therefore cannot produce the necessary reconstruction between fidelities and probabilities. Instead, we have to restrict the transform to the anticommutant of the group G, such that

$$\boldsymbol{f_G} = \boldsymbol{W_{G,\mathrm{A}_G}} \boldsymbol{p}_{\mathrm{A}_G} \tag{3.10}$$

In practice, applying the inverse Walsh-Hadamard transform to the fidelity vector can yield the corresponding error probabilities [38].

The NR algorithm is described below [43].

1. Choose one- or two-qubit twirling sequences from the Clifford group (Hadamard, phase, and/or CNOT gates)

2. Sample empirically to estimate the probability distribution from measurement outcomes

3. Calculate the Walsh-Hadamard transform of this probability distribution

4. Fit these transformed values to the exponential decay Af^m dependent on sequence length m, yielding the fidelities f

5. Perform reverse transform and project onto probability vector, which will reconstruct the entire list of effective qubit error rates

This procedure converges to the estimate of the average noise [38]. It scales polynomially in the number of qubits and the number of error rates. But since the possible correlations depends on the number of qubits, the number of error rates scales exponentially in the number of qubits. To limit this scaling to polynomial rather than exponential, error correlations are limited in range according to the physically-motivated constraints of error correlations between a qubit and only its nearest neighbors as defined by the topology of the qubit register.

Chapter 4

Empirical Direct Characterization

The Empirical Direct Characterization method is introduced in "Modeling Noisy Quantum Circuits Using Experimental Characterization" which is authored by Megan L. Dahlhauser and Travis S. Humble and published by the American Physical Society in Physical Review A [1] (DOI: https://doi.org/10.1103/PhysRevA.103.042603). It is reproduced here under the Creative Commons Attribution 4.0 International license (CC BY 4.0) [2]. Changes to the formatting have been made but the content is unchanged.

4.1 Introduction

Quantum computing is a promising approach to accelerate computational workflows by solving problems with greater accuracy or using fewer resources as compared to conventional methods [44, 45, 46, 47]. Testing and evaluation of early applications on experimental quantum processing units (QPUs) is now possible using prototypes based on superconducting transmons [48, 21, 49, 50] and trapped ions [51, 52, 53, 54] among other technologies. Although these QPUs lack the fault-tolerant operations required for known computational speed ups, they offer the opportunity to understand the behaviors of noisy quantum computing [18].

Noisy, intermediate-scale quantum (NISQ) devices have enabled a wide range of early application demonstrations [55, 56, 57, 58, 59, 21], but validating program performance in the presence of non-reproducible device behaviors remains a fundamental challenge. NISQ

devices are characterized by noisy and erroneous operations, where gate characterizations often change in time and with the nature of the program being implemented [60, 61]. The experimental characterization of individual gates has relied on high-fidelity physics models for the underlying devices with common methods including quantum state tomography (QST) [62], quantum process tomography (QPT) [63, 64], gate set tomography (GST) [41], and randomized benchmarking (RB) [65, 66, 67]. Physics-driven characterizations offer valuable insights into the underlying noise and errors that can inform the design of new devices and control pulses. However, translating from gate-level characterizations to circuit-level applications is typically resource intensive because these methods often scale exponentially with the size of the qubit register to be characterized. [68].

As NISQ applications evolve toward deeper and wider quantum circuits, characterization methods must also extend to these larger scales. There is also a growing need for characterization techniques that can be executed swiftly and repeatedly to provide context-specific characterization data. Resource-intensive, physics-driven gate characterization techniques are not a scalable solution to characterizing devices and applications which are rapidly increasing in size and generally do not allow for a high level of dynamic tuning. Quantum circuit characterization methods may provide effective models of device behaviors that are efficient to generate and easy to interpret by a supporting programming environment, e.g., a compiler [69, 70, 71]. In particular, the validation of application behavior will require debugging methods and programming techniques that support mitigating computational errors in quantum circuits [72, 43]. Effective models of noisy gates and circuits have already informed robust programming methods that lead to increased application performance [73, 74, 75], but a general method for composing noisy quantum circuit models is still needed.

Here, we introduce methods for generating effective models for noisy quantum circuits in NISQ devices derived from experimental characterization. Our approach is based on modeling application-specific circuits using a suite of characterization tests that build a representative set of noisy subcircuit models. We compose noisy subcircuit models to generate noise models for more complicated circuits at larger scales, and we test the fidelity of the resulting model against experimental data. We show how to iteratively adjust the composite model selected for a noisy application circuit by comparing performance of

the predicted behavior against application observations using the total variation distance (TVD) [73]. The iterative and flexible nature of this modeling approach is demonstrated using applications based on GHZ-state preparation and the Bernstein-Vazirani algorithm for search. We develop model composition for the fixed-frequency superconducting transmon devices available from IBM, though we propose these techniques may extend to other NISQ devices as well.

This characterization method is a coarse-grained yet fast approach to characterization which scales linearly with the number of elements in the device, e.g. qubits and couplings. Furthermore, it allows for dynamic tuning of characterization data to every execution of a particular application and can be tailored to yield desired information, e.g. development of a noise model using depolarizing parameters or performance of an entangling gate creating an equal superposition. The tradeoff compared to physics-driven characterization techniques is less total information received, which in some cases may result in a lower accuracy in the final effective description of the device.

We present the steps in the modeling methodology in Sec. 4.2 followed by a series of examples using the case of n-qubit GHZ states in Sec. 4.3. In Sec. 4.4, we present results from experimental characterization for the GHZ state on NISQ QPUs and discuss the role of model selection for characterization accuracy. In Sec. 4.5 we show the performance of our noise models composed from this characterization on the GHZ state experimental results. In Sec. 4.6, we apply these models to the case of the n-bit Bernstein-Vazirani algorithm, while we offer final conclusion in Sec. 4.7.

4.2 Model Selection Methodology

We begin by detailing the coarse-grain modeling methodology before providing specific examples of its implementation. Consider the input for noisy circuit modeling to be an idealized quantum circuit C that is expressed in the available instruction set architecture (ISA) for a given QPU [45]. While the gates defined by the ISA may not be directly implemented within the QPU, the representation used for the ideal circuit will define the operators available for gate characterization. The input circuit is decomposed into a set

$S(C) = \{S_i\}$ of idealized subcircuits S_i that each represent a subsection of the total area of circuit C. The area of C is defined by its width (register size) and depth (length of the operation sequence). The area of each subcircuit S_i is defined by the selected subcircuit width taken from C and the longest depth of the selected gate sequence. For example, a circuit C composed of one- and two-qubit gates as shown in Fig. 4.1 may be decomposed into a set S of two-qubit subcircuits which have depth of two gates and width of two qubits.

Circuit decomposition is not unique and a given decomposition is selected based on tradeoffs in the cost of characterizing each subcircuit, prior knowledge of the suspected device noise and error processes, and any potential structure or symmetry in the circuit design. A complete characterization requires every gate and register element within the input circuit to be included in at least one subcircuit. In general, the selected subcircuits need not be disjoint. The ability to tune the decomposition enables coarse-graining of the noisy circuit model, which is formed by composing the results from subcircuit characterization.

Next we test each subcircuit to characterize the noise present within the coarse-grained area. Each test circuit specifies an idealized outcome based on the input state and gate sequence for the subcircuit instance. We select test circuits to be informative yet limited in both number and circuit dimensions in order to increase efficiency and improve scalability. To test a subcircuit S_i, we may select the full subcircuit S_i provided the ideal outcome is known, but we may select additional test circuits to gain more information and refine our noise models. The set of test circuits $T = \{T_i\}$ is therefore at least as large as S and generally larger. For example, given a two-qubit subcircuit S_i consisting of a one-qubit gate followed by a two-qubit gate, we may select two test circuits–the first circuit consisting of the one-qubit gate and the second circuit consisting of both gates.

The process for selecting test circuits $T(S) = \{T_i\}$ for each S_i follows a set of guidelines detailed below.

1. Identify the components used in S_i.

   (a) Qubit register of size n with qubit identities $q_j \in \{q_0, q_1, ..., q_n\}$

   (b) State preparation $|\psi_j\rangle$

   (c) Measurement basis B

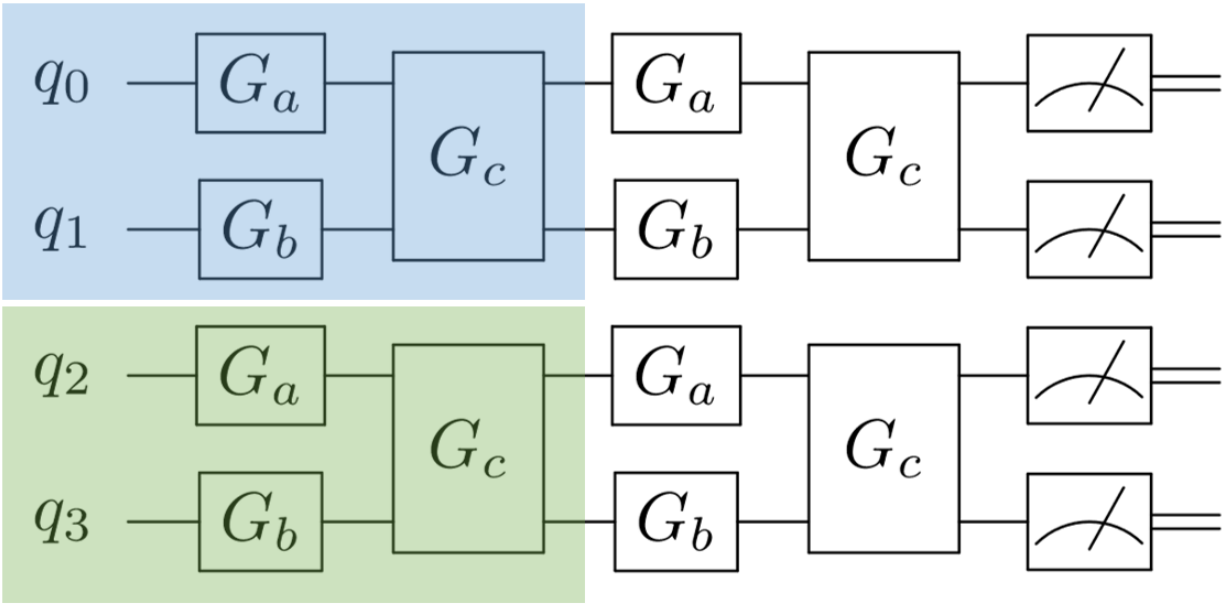

Figure 4.1: An example of a subcircuit decomposition where subcircuit set $S = \{S_{blue}, S_{green}\}$.

(d) Gate sequence $\mathcal{G}$

2. Generate measurement subcircuit T_{meas} consisting of initialization of $|\psi_j\rangle$ and measurement in B for each q_j. If $|\psi_j\rangle$ is unknown or more tests are needed, select or add the computational basis states $|0\rangle$ and $|1\rangle$. Additional input states may include superposition states such as $|\psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ or randomly generated input states $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$.

3. Identify the set $g = \{g_k\}$ of the gates or gate compositions of $\mathcal{G}$ for which the expected outcomes may be calculated for a given input.

4. Select set g' for testing. Elements of g' are gates from g or compositions of gates from g which represent sequences of increasing depth from subcircuit S_i. The selection of g' may be based on tradeoff in the cost of characterization or informed by prior knowledge of expected noise processes or iterative refinement, similar to subcircuit selection.

5. For each element $g'_k \in g'$, generate a circuit $T_k(g'_k)$ which consists of initialization of $|\psi_j\rangle$, application of g'_k applied to the q_j identified from S_i, and measurement in B.

6. The set of test circuits is $T = \{T_{meas}, T_i(g'_k)\ \forall\ g'_k\}$.

The implementation and execution of test circuits on a QPU generates a corresponding set of measurement observations. Each test circuit is executed multiple times to gather statistics from the distribution of results R_i that characterize subcircuit T_i. The i-th characterization is denoted as $H_i = (T_i, R_i)$ and the set of all characterizations is given as H. The number of characterizations is fixed by the number of test circuits $|T|$, while the number of measurement observations acquired for each test circuit is set by the sampling parameter N_s. Assuming the same sampling for all tests, then there are a total of $N_s|T|$ measurement observations, i.e., experiments, required for H.

The results of experimental characterization are used to formulate concise approximate models of the subcircuits' observed behaviors. We model each noisy subcircuit as the idealized subcircuit followed by a quantum channel that accounts for the noise [76]. Let the noisy subcircuit model $M_i = M(S_i, p_i)$ representing subcircuit S_i depend on model

parameters p_i. We estimate the channel parameters using the characterization H_i, where the method of parameter estimation will vary with the selected model. Parameter estimation may be either direct or optimized methods. For example, least-square error estimates may be used to estimate parameters from noisy measurement observations by optimizing the residual model error.

We quantify the error in the resulting models using the total variation distance (TVD) [73], which is defined as

$$d_{\text{TV}}(H_i, M_i) = \frac{1}{2} \sum_k \left| r^{(H_i)}(k) - r^{(M_i)}(k) \right| \tag{4.1}$$

where $r^{(H_i)}(k)$ is the probability of the k-th outcome of the test circuit T_i and $r^{(M_i)}(k)$ is the corresponding probability predicted by the noisy circuit model. The TVD vanishes as the predictions of the model become more accurate in reproducing the observed results and reaches a maximum of unity when the sets are completely disjoint.

After estimating the model parameters $p = \{p_i\}$ for all subcircuits, the corresponding noisy circuit model $M(C, p)$ for the input circuit C is composed. The method of composition of the noisy subcircuit models is paired with the decomposition method to ensure a consistent representation of the original input circuit. In the examples below, we consider modeling methods based on independent noisy subcircuit models that permit separable composition-decomposition methods and defer discussion of non-separable models, e.g., context-dependent noise, to Sec. 4.7.

Final selection of the noisy circuit model is then guided by the accuracy with which the composite model reproduces the performance of the circuit C on the QPU. For clarity, we define the actual executed circuit $A = (C, R_c)$ with R_c the recorded results, and we measure the accuracy of the noisy circuit model as $d_{\text{TV}}(A, M)$. The desired TVD sets an upper bound on the threshold for model accuracy. If this user-defined threshold is not satisfied, selection of the noisy subcircuit models is revisited. This iteration may include refinement of the noisy subcircuit models to improve the accuracy of each M_i or redefinition of the circuit composition-decomposition methods to manage the trade-offs in modeling complexity and accuracy. The former requires repeated post-processing analysis of the characterization H,

whereas the latter requires additional characterization testing. In either case, model selection continues until the threshold has been met. Once the accuracy threshold has been satisfied, noisy circuit modeling is complete.

The noisy subcircuit models can then be tested for robustness in predicting the expected outcome from both the input circuit and other circuits executed on the characterized device. We again use TVD to measure the accuracy for selected models to characterize the behavior of other application circuits within the same QPU context.

We summarize the complete procedure as follows.

1. Identify ideal circuit C.
2. Decompose the circuit into set $S(C) = \{S_i\}$ of ideal subcircuits S_i.
3. Select set of test circuits $T = \{T_i\}$ which define an input state and ideal outcome for each element in S.
4. Propose a noisy subcircuit model $M_i = M(S_i, p_i)$ for each element in S parameterized by p_i.
5. Implement and execute T on QPU to generate experimental characterizations $H_i = (T_i, R_i)$ using results R_i returned from QPU.
6. Using set of characterizations $H = \{H_i\}$, fit noise parameters p_i based on calculated expected probabilities for each M_i.
7. Compose the noisy circuit model $M(C, p)$ for the target circuit and compare the actual executed circuit $A = (C, R_C)$ with recorded results R_C from the QPU to the noisy circuit model using $d_{TV}(A, M)$.
8. If d_{TV} is not at threshold return to 2, apply refinements to 2, 3, and 4, and continue to 7 until threshold is met.

For step 8, refinements to step 2 include additional elements selected from the set g, addition of compositions of elements in g such that the test components are larger, or addition of elements to g not explicitly represented in $\mathcal{G}$. Refinements to step 3 include

additional initializations as test circuits. Refinements to step 4 include additional noise model parameters p_i or different noise channels to define M.

4.3 Application to GHZ States

We next illustrate the methodology of Sec. 4.2 using the example of a GHZ-state preparation and measurement circuit. We generate noisy quantum circuit models for this application for various circuit sizes executed on the IBM `poughkeepsie` QPU, which has a register and layout as shown in Fig. 4.2. All data for characterization tests and applications is collected in a single job sent to `poughkeepsie`, a process which typically required under 30 minutes of execution time after queuing. As the `poughkeepsie` device is periodically calibrated, our experimental demonstrations ensure that all data is collected within one calibration window to preserve the QPU context. The software implementation of our examples below as well as all experiment and simulation details such as subcircuits and noise models is available publicly [77].

We consider the example of preparing the n-qubit GHZ state

$$|\text{GHZ}(n)\rangle = \frac{1}{\sqrt{2}}\left(|0_1, 0_2, ..., 0_n\rangle + |1_1, 1_2, ..., 1_n\rangle\right) \tag{4.2}$$

where the subscript denotes the qubit and the schematic representation of the input circuit C is given in Fig. 4.3. The instruction set for this circuit is limited to the one-qubit Hadamard (H) and two-qubit controlled-NOT (CNOT) unitaries along with the initialization and readout gates acting on a quantum register of size n. We study this example for a range of register sizes from $n = 2$ to 20 by composing a noisy circuit model that represents GHZ-state preparation on a QPU based on superconducting transmon technology [78, 79]. This example demonstrates the unique features of superposition and entanglement using a circuit depth that is within the capabilities of the NISQ devices [80, 81].

We decompose the GHZ-state preparation circuit from Fig. 4.3 into a set of subcircuits S based on the procedure detailed in Sec. 4.2. In this example, we identify a series of overlapping 2-qubit subcircuits for coarse-graining the n-qubit state preparation. Spatial

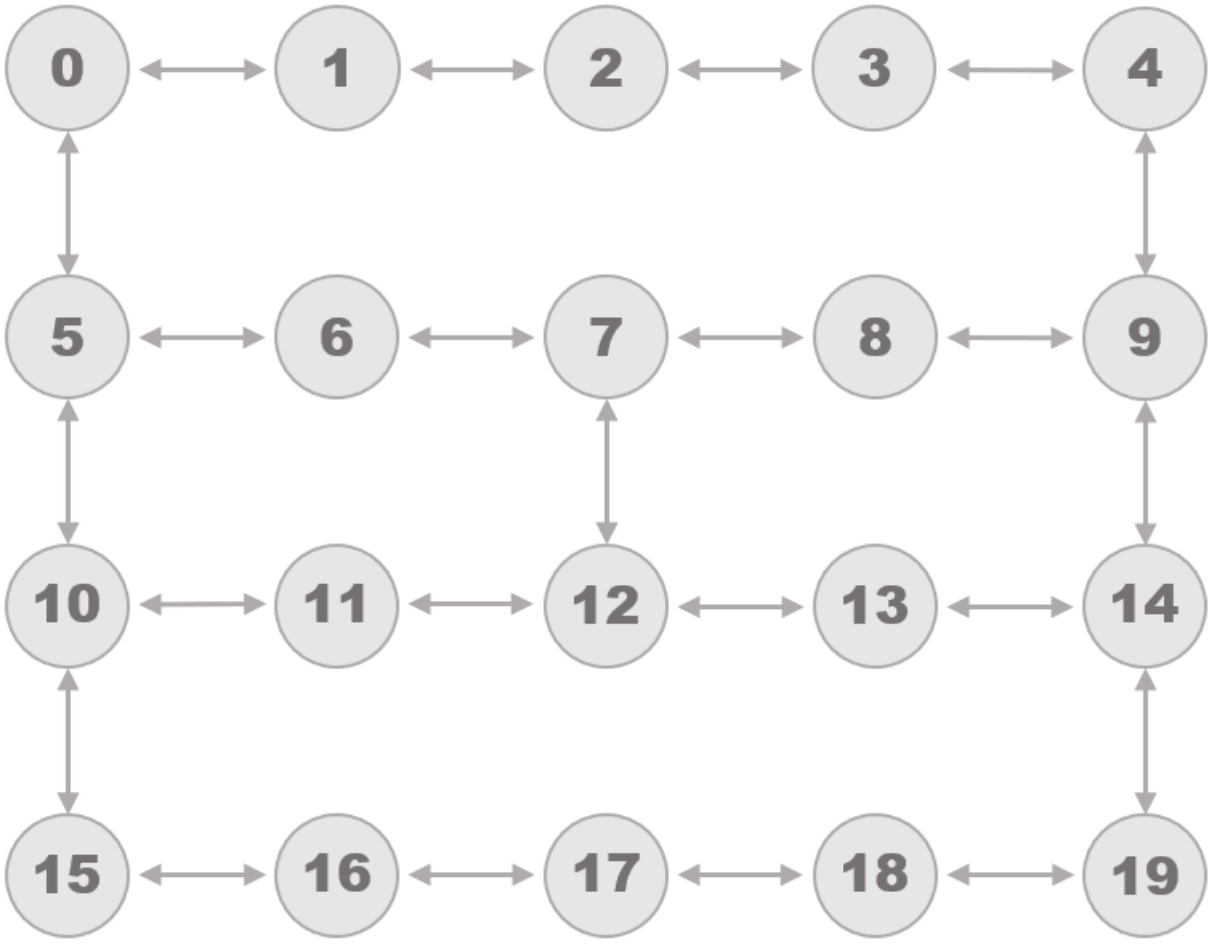

Figure 4.2: A graphical representation of the register connectivity in the `poughkeepsie` QPU at the time of data collection, in which each node corresponds to a register element and directional edges indicate the availability of a programmable two-qubit cross-resonance gate.

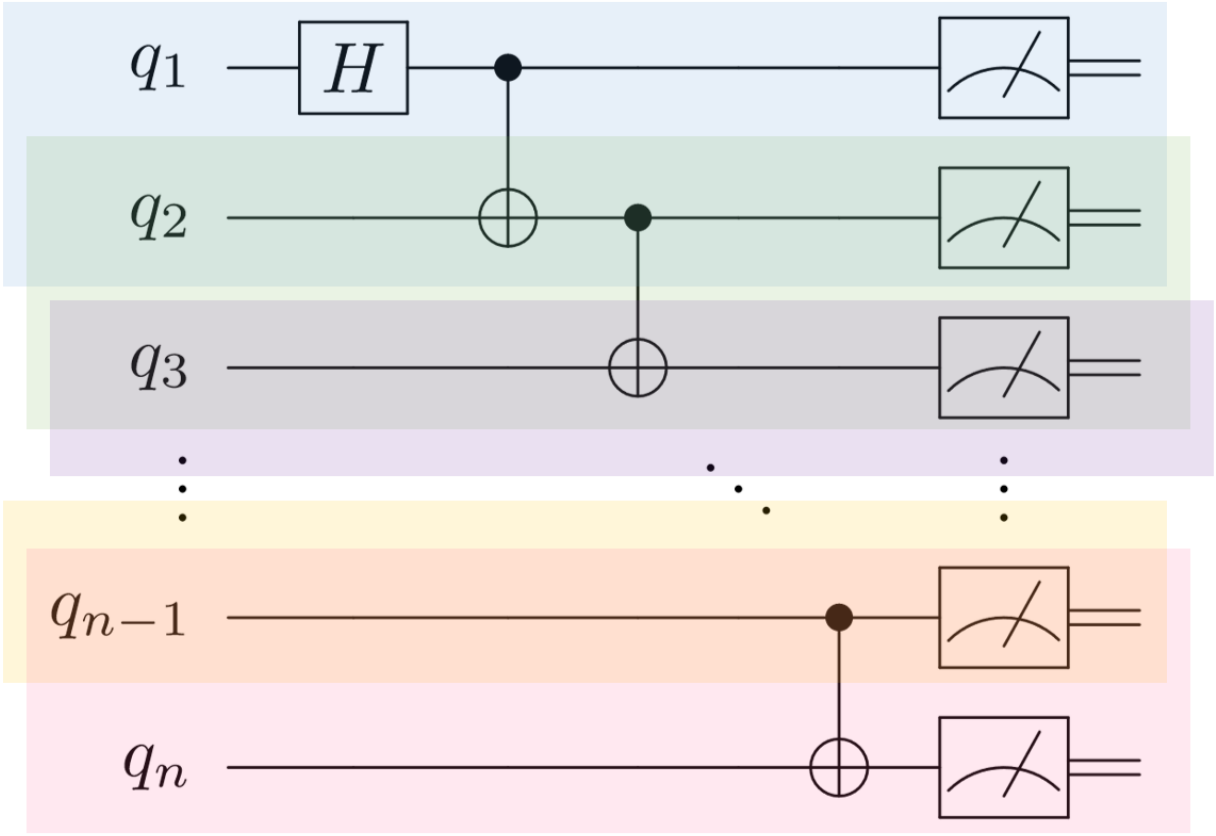

Figure 4.3: The schematic representation of the quantum circuit used for preparation of the n-qubit GHZ state defined by Equation 4.2. The circuit layout satisfies the connectivity constraints of the IBM `poughkeepsie` QPU shown in Fig. 4.2. The circuit uses a total of $n-1$ CNOT gates and n measurement gates. Colored boxes denote subcircuit selections.

variability in the device noise motivates a decomposition based on each register element q_i. We extend these subcircuits to generate a corresponding set of test circuits T by the set g given as

$$g = \{H_{q_0}, \text{CNOT}_{q_0,q_1}\} \tag{4.3}$$

from which we select

$$g' = \{H_{q_0}, H_{q_0} \circ \text{CNOT}_{q_0,q_1}\} \tag{4.4}$$

The expected outcomes of these particular test circuits are simple to calculate from the truth tables for each operator [23]. We examine the models using these test circuits.

4.3.1 Noisy Measurement Model

We begin by characterizing the initialization and measurement test circuits, which are necessary for modeling noisy unitary gate behavior. The measurement process for each register element discriminates an analog signal to generate a classical bit [82], and errors in signal discrimination may lead to the wrong value. Characterization of measurement records the number and type of outcomes observed for each initial state. We characterize each register element with respect to both the 0 and 1 output states. The leading errors in the observed results occurs when the j-th register element maps an expected output value to its complement, i.e., $0 \rightarrow 1$ and $1 \rightarrow 0$.

We model measurement of the j-th element as a binary process subject to errors which act on the post-measurement classical bit string, and we consider two models for the measurement error process: symmetric readout noise (SRO) and asymmetric readout noise (ARO). The SRO model is defined by a single parameter p_{sro} that specifies the probability for a bit to flip, and we define a test circuit to characterize this process as measurement immediately after initialization to state $|0\rangle$. We directly estimate the value of p_{sro} from the number of errors when preparing this computational basis state as $p_{\text{sro}} = r(1)$, where $r(k)$ is the observed probability of k errors recorded. This model implicitly delegates initialization errors to the readout error model. The SRO model is developed by test circuits $T = \{T_{meas}(|0\rangle)\}$ where the final SRO model is defined by $M_{SRO} = M(T_{meas}, p_{sro})$.

By contrast, the ARO model uses two parameters: p_0 for the probability of error in readout of $|0\rangle$ and p_1 as the probability of error in readout of $|1\rangle$. The ARO model therefore represents a refinement of both the noise model parameters p_i and the test circuit suite T. We may estimate p_0 using the same test circuit above, but we must extend the characterization to preparation and measurement of $|1\rangle$ to estimate p_1. These additional test circuits will require inclusion of the single-qubit X gate, and we also add a test circuit for the XX operation of two successive X gates applied to a single qubit. The latter reproduces the initial state $|0\rangle$, enabling the error in readout of state $|1\rangle$ to be isolated from the error associated with the X gate. The ARO model is therefore defined by $M_{ARO} = M(T, p_0, p_1)$ where $T = \{T_{meas}(|0\rangle), T_{meas}(|1\rangle), T_{XX}(|0\rangle)\}$.

We model the test circuits for the ARO process using an isotropic depolarizing channel parameterized by p_{x} to describe noise in the X gate,

$$\epsilon_{DP}(\rho) = (1 - p_x)I\rho I + \frac{p_x}{3}(X\rho X + Y\rho Y + Z\rho Z) \tag{4.5}$$

where I, X, Y, and Z are the Pauli operators. Characterization of the ARO model yields an overdetermined system of equations relating the four experimentally observed probabilities $r^{(X)}(0)$, $r^{(X)}(1)$, $r^{(XX)}(0)$, and $r^{(XX)}(1)$ to the parameters p_0, p_1, and p_{x}. Of these parameters, only the latter two are unknown since p_0 is determined by the same method outlined above for p_{SRO}. Because the experimental observations directly relate to each other via $r^{(X)}(0) + r^{(X)}(1) = 1$ and $r^{(XX)}(0) + r^{(XX)}(1) = 1$, we select the following system of equations for each register element based on counts of $r^{(\cdot)}(0)$.

$$r^{(X)}(0) = \frac{2p_{\mathrm{x}}}{3}\Big(1 - p_0\Big) + p_1\Big(1 - \frac{2p_{\mathrm{x}}}{3}\Big) \tag{4.6}$$

$$\begin{aligned} r^{(XX)}(0) =& (1 - p_0)\left[\left(1 - \frac{2p_{\mathrm{x}}}{3}\right)^2 + \left(\frac{2p_{\mathrm{x}}}{3}\right)^2\right] \\ &+ p_1\left[\frac{4p_{\mathrm{x}}}{3}\left(1 - \frac{2p_{\mathrm{x}}}{3}\right)\right] \end{aligned} \tag{4.7}$$

This system of equations is solved using the SciPy function `fsolve`, which is based on Powell's hybrid method for minimization [83].

4.3.2 Noisy Subcircuit Models

Test circuits for characterizing noisy subcircuits generate results that include measurement noise. We use the noisy measurement model above to account for these behaviors when modeling the results from test circuits. For the SRO and ARO models discussed above, this directly estimates the probabilities expected to be observed for each register. We use this procedure when discussing the characterization below.

We first characterize the subcircuit representing the Hadamard operation. The test circuit for a single Hadamard is defined with respect to the expected values for input states drawn from the computational basis, which yield a uniform superposition of binary results upon ideal measurement. We also use even-parity sequences of Hadamard gates as a second test to estimate noise in the subcircuit. These test circuits $T = \{T_H(|0\rangle), T_{HH}(|0\rangle), T_{4H}(|0\rangle), T_{6H}(|0\rangle), ..., T_{nH}\}$ are used to characterize the Hadamard gate to yield $M_H(T, p_H)$.

We define test circuits for the CNOT operations that mirror the subcircuits used in the target application. For GHZ-state preparation, these are based on characterization of Bell-state preparation. The test circuit specification shown in Fig. 4.4 produces the idealized result of a uniform distribution over perfectly correlated binary values. These test circuits may be defined across all pairings of register elements as represented by Fig. 4.3. In particular, additional CNOT test circuits may be added to the set g' from the set g, and additional CNOT test circuits for couplings not explicitly in $\mathcal{G}$ may be added as well. For convenience, we will denote the Bell-state preparation subcircuit as $U^{\mathrm{BS}}_{(j,k)} = U^{(\mathrm{CNOT})}_{(j,k)} H_{(j)} |0_j, 0_k\rangle$.

The noisy test circuits for Bell-state preparation are modeled by a pair of identical, independent depolarizing channels. Each channel, together defined as $\epsilon^{\mathrm{DP}}_{j,k} = \epsilon^{\mathrm{DP}}_j \otimes \epsilon^{\mathrm{DP}}_k$, is parameterized by p_{CNOT}, which represents the probability of a depolarizing error determined independently for each qubit in the two-qubit CNOT gate. We therefore use the test circuit $T = \{T^{\mathrm{BS}}_{(j,k)}(|0_j, 0_k\rangle)\}$ to compose model $M_{\mathrm{CNOT}} = M(T, p_{\mathrm{CNOT}})$.

The probability of observing bits a and b is given by

$$r_{j,k}(ab) = \mathrm{Tr}\left[\Pi_{ab} \epsilon^{\mathrm{DP}}_{j,k} \left(U^{\mathrm{BS}}_{(j,k)} |0_j, 0_k\rangle \langle 0_j, 0_k| U^{\mathrm{BS}\dagger}_{(j,k)} \right) \right] \tag{4.8}$$

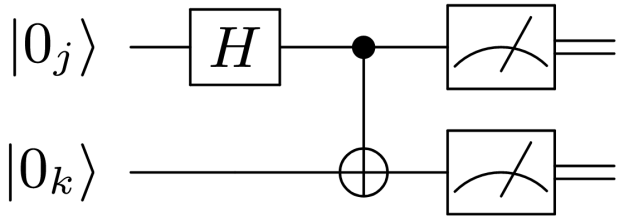

Figure 4.4: The test circuit for characterizing the CNOT operation acting on register elements q_j and q_k. This test prepares the two-qubit Bell state as an instance of $n = 2$ in Fig. 4.3.

where the operator Π_{ab} projects onto the state $|a,b\rangle$, and the resulting trace yields the probability of the ideal measurement. The probabilities expected from the noisy Bell state subcircuit on qubits j,k with ideal measurement is then given by

$$\begin{aligned} r_{j,k}(00) = r_{j,k}(11) &= \frac{1}{2} - \frac{2}{3}p_{\text{CNOT}} + \frac{4}{9}p_{\text{CNOT}}^2 \\ r_{j,k}(01) = r_{j,k}(10) &= \frac{2}{3}p_{\text{CNOT}} - \frac{4}{9}p_{\text{CNOT}}^2 \end{aligned} \tag{4.9}$$

Errors in readout transform these probabilities according to the noisy process, which may be either the SRO or ARO model. For example, the probability following readout $s_{j,k}(00)$ under the ARO channel is given by

$$\begin{aligned} s_{j,k}(00) =& (1-p_0^j)(1-p_0^k)r_{j,k}(00) \\ &+ (1-p_0^j)p_1^k r_{j,k}(01) \\ &+ p_1^j(1-p_0^k)r_{j,k}(10) \\ &+ p_1^j p_1^k r_{j,k}(11) \end{aligned} \tag{4.10}$$

From the system of four equations generated by the readout probabilities $s_{j,k}(cd)$, we use the method of least squares to estimate p_{CNOT}. We minimize the sum of the squared residuals,

$$\sum_{cd} \Big(s_{j,k}(cd) - h_{j,k}(cd) \Big)^2 \tag{4.11}$$

where each residual is defined as the difference between the modeled probability $s_{j,k}(cd)$ and the experimentally observed probability $h_{j,k}(cd)$ for each state result cd. The value $h_{j,k}(cd)$ represents the counts of state cd on qubits j,k measured during a total number of experiments N_s. The value returned for p_{CNOT} is found using the SciPy `fsolve` function and bounded between 0 and 1 [83].

4.4 Experimental Characterization

In this section, we report on the results of experimental characterization and noisy circuit modeling of GHZ-state preparation using a QPU based on superconducting transmon

technology developed by IBM. The IBM `poughkeepsie` device has a register of 20 superconducting transmon elements that encode quantum information as a superposition of charge states [84]. Microwave pulses drive transitions between the possible charge configurations and induce single-qubit gates. Coupling between register elements uses a cross-resonance gate that drives a mutual transition between transmons and therefore only occurs between two spatially connected elements [78].

The layout of the 20-qubit register in `poughkeepsie` at the time of data collection is shown in Fig. 4.2. A common edge in the connectivity diagram specifies those register elements that may interact through the cross-resonance operation. Individual registers are measured through coupling to a readout resonator, which results in a state-dependent change in the resonator frequency. Amplification of the readout signal then enables discrimination of the state using a quantum non-demolition measurement [85, 48].

Circuits are sent to the backend where they are translated into the appropriate ISA. The ISA for `poughkeepsie` consists of the gates U_1, U_2, U_3, CX, and ID [86]. The U_1, U_2, and U_3 gates are unitary rotation operators, of which U_1 is a "virtual" gate performed in software and U_2 and U_3 are performed in hardware. The identity gate ID is used as a placeholder to create a timestep since it does not alter a quantum state. CX represents the CNOT gate [87]. These instructions are implemented using low-level hardware operations. For instance, the CX operator is implemented in hardware using a sequence consisting of cross-resonance gates and single-qubit rotation gates [86, 88, 89].

The `poughkeepsie` QPU is accessed remotely using a client-server interface. We employ the Qiskit programming language to specify the input circuit and test circuits for the GHZ-state preparation application [90]. These Pythonic programs are transpiled to the specifications and constraints of the backend, including ISA, connectivity layout, and register size. Additional inputs to the transpiler may include optimization protocols for minimizing circuit operations or noise levels. The transpiled programs are executed remotely on the `poughkeepsie` device, which returns the corresponding measurements along with job metadata.

We use a shot count of 8,192 for all of the circuits executed on `poughkeepsie` which represents the number of times each circuit is individually executed and generates the

distribution of output states from the input circuit. Therefore each probability estimated by experiment is given by $r(k) = C(k)/N_s$, where $C(k)$ is the number of events observed for each measurement and N_s is the shot count of 8,192. These measurements are subject to error due to variability in sampling in experiment from the QPU distribution. We restrict our sample size to a single experiment of 8,192 shots to avoid introducing effects from drift in the `poughkeepsie` QPU. We use the standard deviation of these measurements to report error and statistical fluctuations, which is given by $\sqrt{(p(1-p)/N_s)}$ where p is the binomial distribution probability parameter measured from experiment.

We characterize measurement of all register elements in `poughkeepsie` and analyze the results using the SRO and ARO models. The results for direct estimation of the ARO model parameter p_0 and p_1 are shown in Fig. 4.5. The results for the SRO model correspond with $p_{\text{sro}} = p_0$. From these results, we observe a large spatial variability in readout error as well as asymmetry per register element. The readout of state $|1\rangle$ is almost always more error-prone than readout of state $|0\rangle$.

The results of estimating the parameter p_x for the depolarizing noise model of each X gate are shown in Fig. 4.6. From these results, we see spatial variability in the recovered error parameter. We observe one case of a negative error rate for qubit 17 recovered from direct estimation using Eqs. 4.6 and 4.7. Because an estimated error rate of zero is within the experimental error, this is most likely due to statistical fluctuations. However, it could also be attributable to inconsistencies in the error behavior for the test circuits such that the model cannot estimate a feasible parameter based on the results, or to errors for this register that are not well described by a depolarizing channel such that a different model may yield a better solution. All other error rates are relatively small and therefore we have not investigated model refinement for this case because of the negligible contribution to the noise.

We next characterize the Hadamard gate. We characterize error rates using test circuits generated from long sequences of Hadamards acting on a single element. We observe small error rates which correspond on average to 0.1% error per gate. We attempted to model the Hadamard noise using a depolarizing channel but it did not lead to a better TVD than using a noiseless model for the gate.

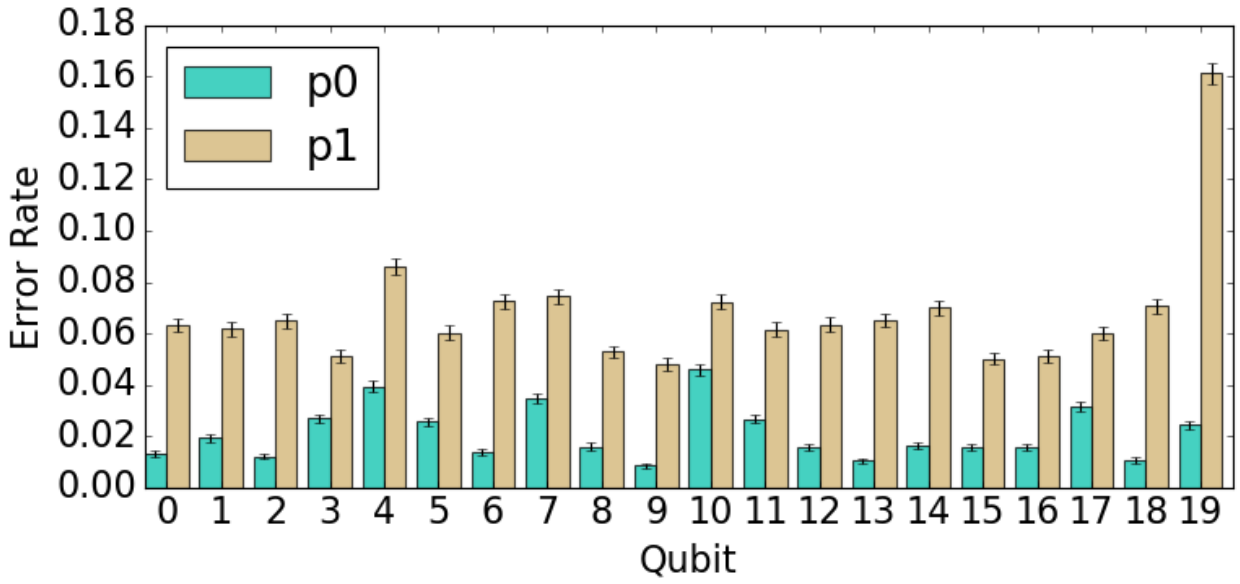

Figure 4.5: Error rates under the ARO channel for each qubit of *poughkeepsie*. The SRO channel is given by the error rates for state 0 shown here. Average p_0 value is 0.0212 (standard deviation of 0.0101 across all qubits) and average p_1 value is 0.0681 (standard deviation of 0.0233). Each qubit is evaluated in a separate circuit, e.g. $X_0 \left|0_0, 0_1, ..., 0_{19}\right\rangle$ to generate Eq. 4.6 for qubit 0.

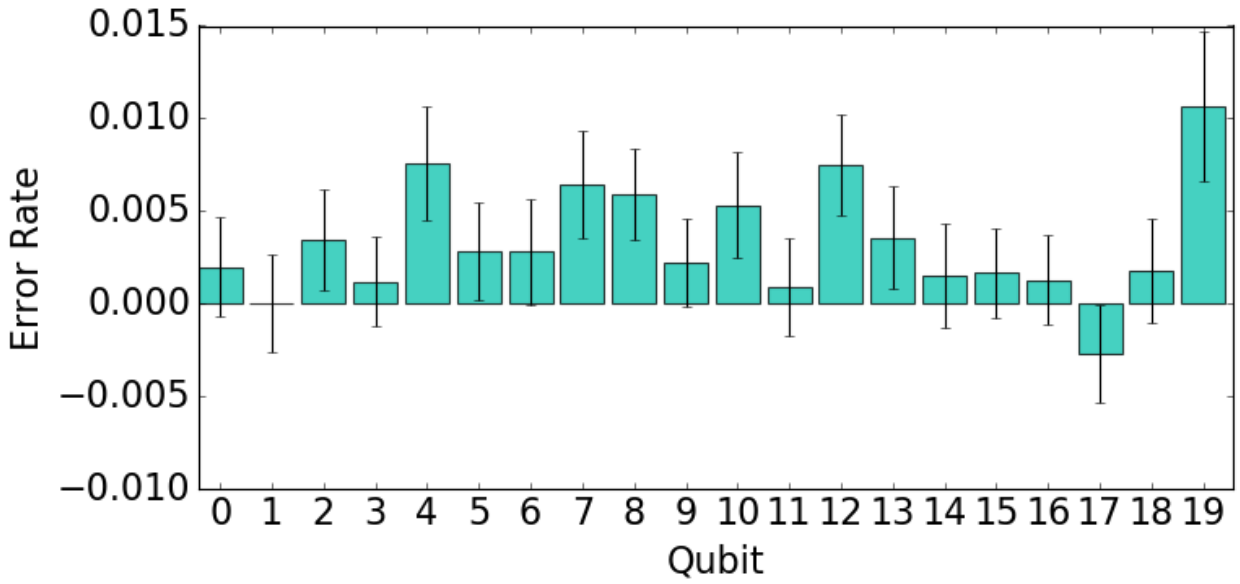

Figure 4.6: Depolarizing error rates associated with X gate application for each qubit of *poughkeepsie*. Average p_x value is 0.0033 with standard deviation 0.00303.

We also characterized gate error models based on unitary rotation noise in X, Y, and Z for the Hadamard gate which represents coherent errors. These characterizations did not yield a smaller TVD than using a noiseless model. Our choice to restrict characterizations to computational basis measurements significantly limits the achievable accuracy or effectiveness of this model. In general, such characterizations are not capable of identifying arbitrary coherent noise and are limited, e.g. only X and Y noise have an observable effect in the Z measurement basis. Additional test circuits could address this limitation at the expense of increased experiment count. For our purposes, we concluded that error rates associated with the Hadamard operation were negligible as this noise was 100 times smaller than the next leading gate error.

We next characterize the Bell-state preparation circuits for each pair of possible interactions shown in Fig. 4.2. We select the depolarizing noise model because it is a well-understood model for quantum noise that captures several different fundamental aspects of quantum behavior. We do not expect the depolarizing model to be a perfect fit to experimental data but this model provides a useful method to understand noise levels in the system and how noise from different components interacts. We use least-squares error estimation to find the value of depolarizing parameter p_{CNOT} that best fits the results while accounting for readout error as in Equation 4.10. This approach yields more consistent results than solving each equation in the system explicitly and using a selection process to determine the final p_{CNOT} value from among these solutions which are often highly varied. The estimated parameter values are shown in Fig. 4.7. The magnitude of the error bars for the parameter estimations highlights the relative magnitude of gate noise to readout noise.

We test the accuracy of the noisy subcircuit models with estimated parameters from experimental characterization. For these tests, we use explicit numerical simulation of the quantum state prepared by each noisy subcircuit model. We estimate the measurement outcomes for these modeled circuits using the simulated quantum state, and we compare these simulated observables with the corresponding experimental observations from the `poughkeepsie` device. The accuracy of the noisy subcircuit model is quantified using the total variation distance (TVD) defined in Equation 4.1.

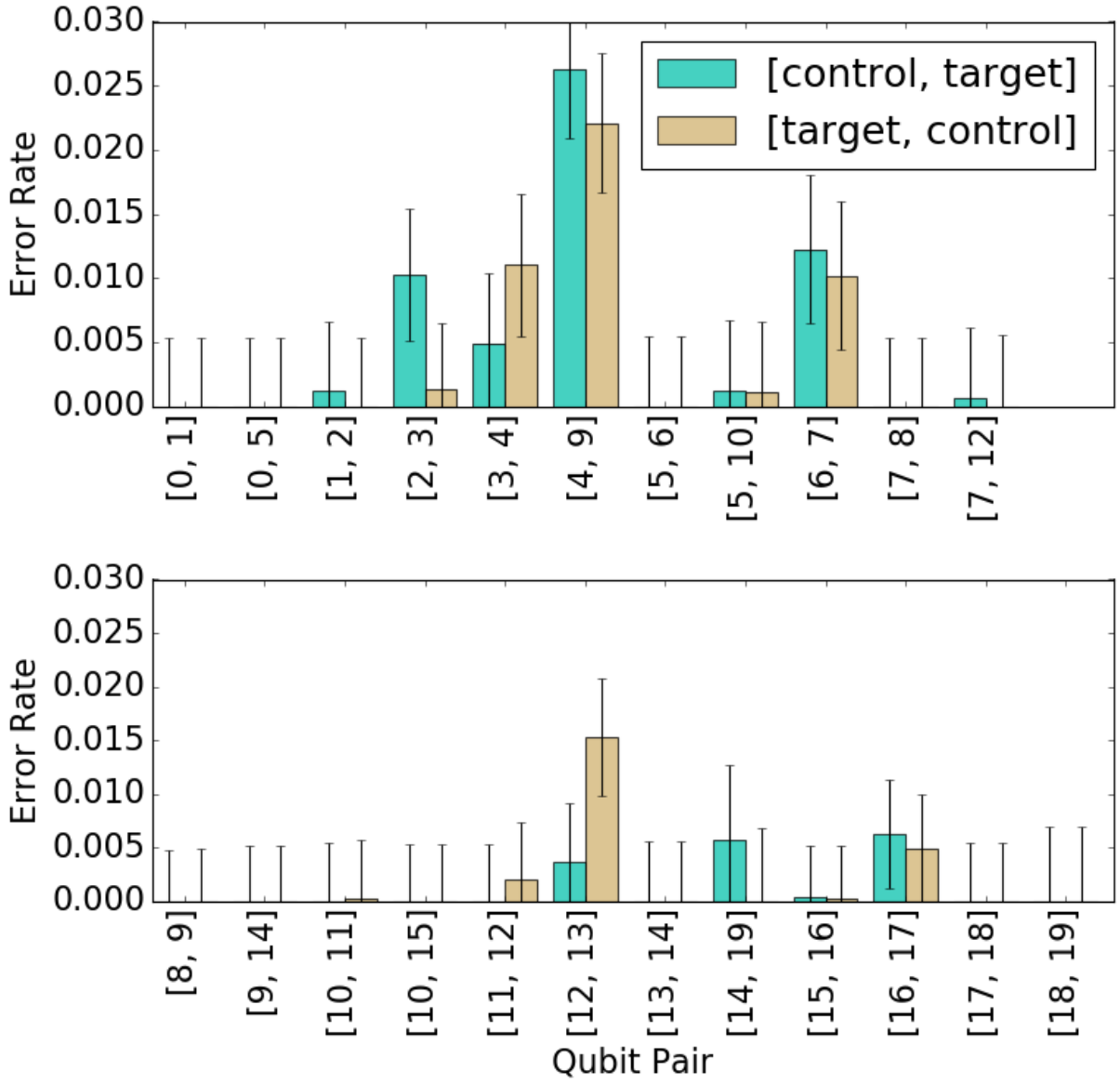

Figure 4.7: Error rates for CNOT gates under the depolarizing channel for each coupled qubit pair of *poughkeepsie*. These values are fitted to include the ARO channel noise with rates shown in Fig. 4.5. Reported error bars represent the upper limit of the error from the least squares calculation.

Our simulations of the quantum state use a numerical simulator bundled into the Qiskit software framework. The Aer software simulates both noiseless and noisy quantum circuits using the same Qiskit programs sent to the `poughkeepsie` device as input. We constrain the simulator to a statevector simulation method. Within Aer, we input the noise models using the error rates and noise operators of depolarizing and readout channels as defined in Sec. 4.3. Aer models gate noise using error functions parameterized by these error rates which create noisy descriptions of gates for simulation. When a noisy simulation is run, these functions sample errors and inject them as operations within the circuit. We tailor the simulations to match the developed noisy subcircuit models. Each test case acquired N_s samples in order to mimic the finite statistics from experimental characterization. We generate a number of simulation samples of 8,192 shots per sample to create a sampling distribution. We report the standard deviation of this distribution which represents error due to variability in sampling in simulation.

A comparison of accuracy for different noisy subcircuit models is shown in Fig. 4.8 for simulating the Bell state circuit on qubits 0 and 1 on the `poughkeepsie` device. We calculate the TVD between experiment and simulation using six different noise cases. We consider symmetric readout only (SRO), asymmetric readout only (ARO), CNOT depolarizing error only (DP), symmetric readout with CNOT error (SRO+DP), and asymmetric readout with CNOT error (ARO+DP). The error rate parameters are optimized for each composite noise model, e.g. the optimal depolarizing parameter in the SRO+DP case may not be the same value found for the ARO+DP case. We also simulate a noiseless Bell state for a baseline comparison.

The results shown in Fig. 4.8 clarify the noisy circuit model yielding the smallest TVD is composed from the asymmetric readout channel with a CNOT depolarizing channel (ARO+DP). Since each noise model achieves a clear improvement in TVD as measured by a decrease from the noiseless case that is outside of error bars, we can be confident that each selected model is capturing some of the noise behavior present in the system while also illustrating which models provide the best descriptions of the noise. For example, in the noise model case 'DP' we have modeled a depolarizing channel for which the p_{CNOT} parameter is calculated to account for all noise in the system. This model has a clear improvement

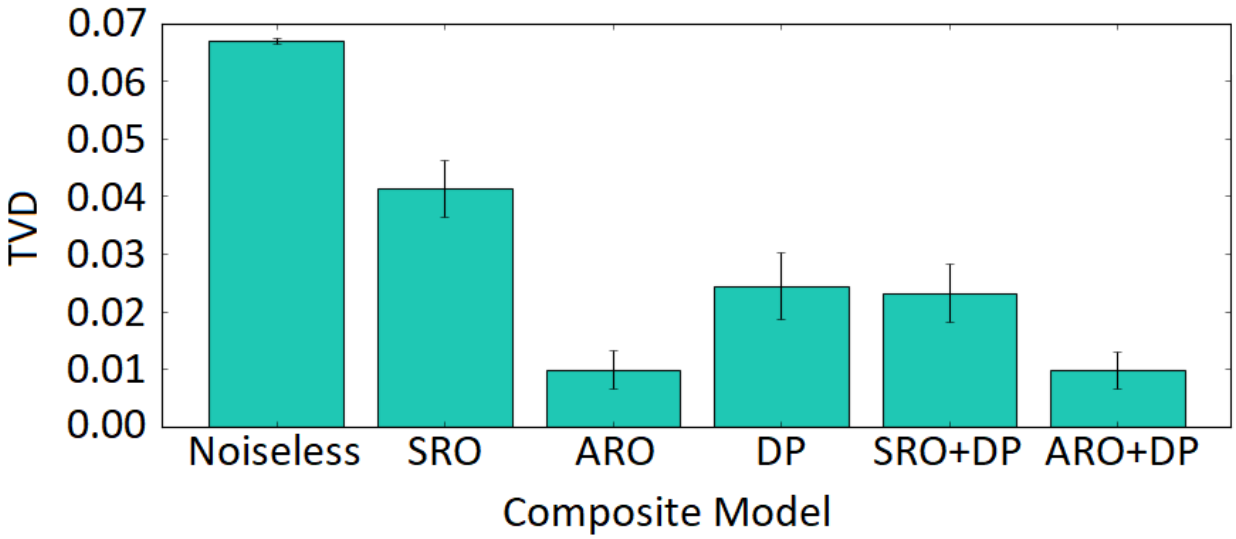

Figure 4.8: Comparison of possible choices for composite model. The best performance is achieved in the ARO+DP case. Error bars represent the distribution of TVD values across 100 sets of 8,192 samples per simulation case.

on TVD and therefore is likely to be an effective description of the noise in the system. However, the addition of readout noise models for the 'SRO+DP' and 'ARO+DP' cases is evidently a more accurate noise model because these models achieve further improvements in TVD.

4.5 Performance Testing Results

We now present the performance of the selected composite model on n-qubit GHZ-state preparation circuits. Using the estimated ARO and CNOT error rates, we demonstrate iterations of this composite noise model which represent varying model complexity and experimental efficiency to achieve a particular accuracy. These iterations are shown in Fig. 4.9. The 2-qubit average case represents the performance of a noise model with only three parameters–$p_0, p_1, p_{\text{CNOT}}$–which are taken as the average of the error rates for only qubits 0 and 1. This represents a case of characterization using the fewest quantum resources, requiring only 7 experiments. We also consider a case which uses these same three parameters averaged over the entire register which retains low model complexity of only three noise parameters but requires the full suite of experiments. Our most detailed model accounts for spatial variations in the error parameters and uses individualized readout error rates for each qubit and CNOT error rates for each coupling. As with the Bell state example in Fig. 4.8, we show the noiseless case for the sake of context and comparison. Finally, we also show the sum of the minimum TVD achieved for noisy simulation of the Bell state across each qubit pair for which a CNOT was applied in the GHZ preparation circuit.

Figure 4.9 demonstrates a significant improvement in model accuracy for GHZ state preparation using our composite noisy circuit model. The improvement is a 3-fold decrease in TVD as compared to the noiseless simulation. Our fully spatial model performs better than the coarser-grained models, such as the average two-qubit model, particularly for larger sizes of GHZ state preparation. We also examine the scaling in the error with respect to the area of the circuit. We normalize the computed TVD by the number of CNOT gates in each GHZ preparation circuit, and we find that the per-qubit model accuracy is nearly constant across all GHZ circuit instances, as shown in Fig. 4.10. This trend would also hold when

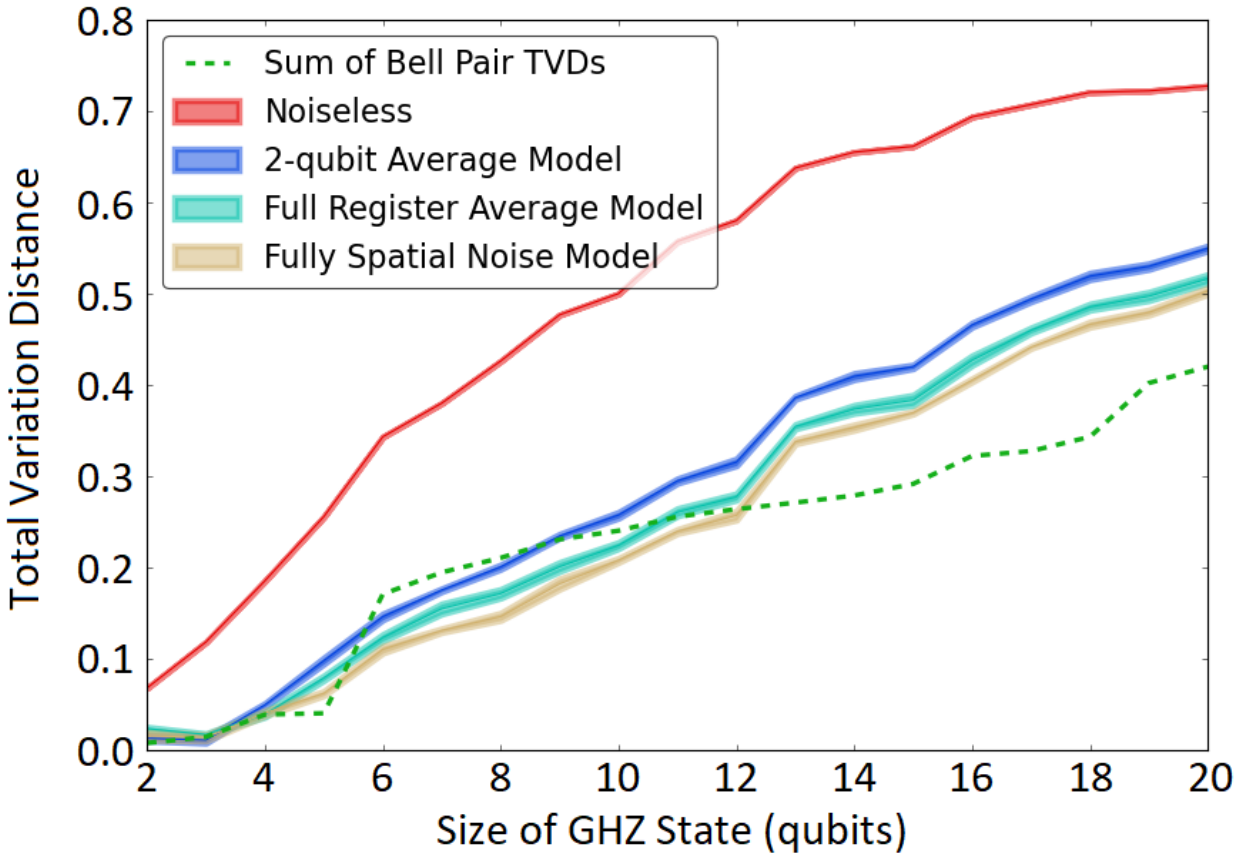

Figure 4.9: Performance of selected noise model on n-qubit GHZ states. The best performance is achieved with the fully spatial noise model. Error bars represent the distribution of TVD values across 6 sets of 8,192 samples per simulation case.

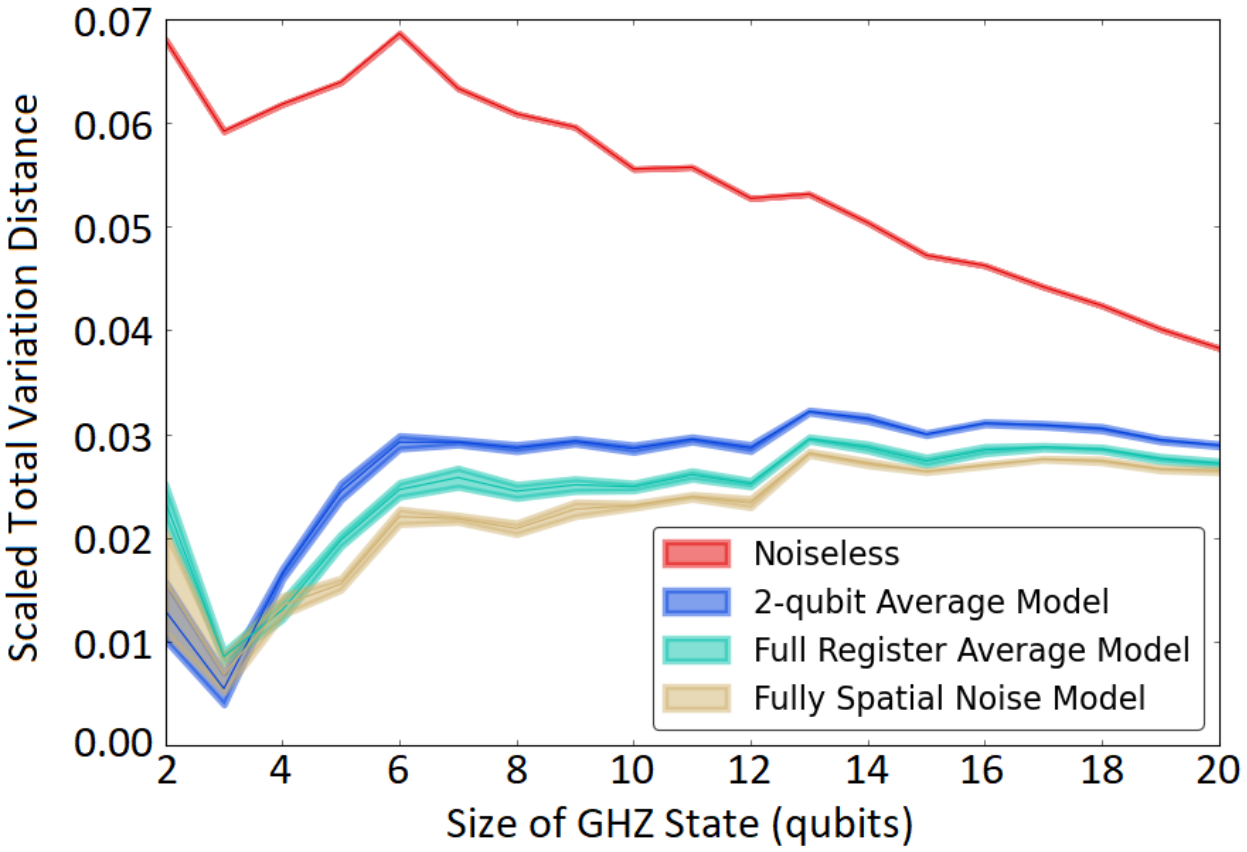

Figure 4.10: Scaled performance of selected noise model on n-qubit GHZ states, where TVD is divided by the number of CNOT gates in each circuit. Error bars represent the distribution of TVD values across 6 sets of 8,192 samples per simulation case.

TVD is scaled by qubit count, since qubit count and CNOT count are strongly linked in the GHZ example. Since the TVD increases at a rate commensurate with CNOT count or qubit count, this may indicate that higher levels of entanglement or larger Hilbert spaces impact the predictability of noise in the device.

4.6 Bernstein-Vazirani Application

We next test the performance of this noisy circuit model on a different application to evaluate its ability to capture fundamental characteristics of the device. We test the performance by modeling several quantum circuit instances of the Bernstein-Vazirani algorithm. This algorithm considers a black box function that is encoded by a secret binary string which the Bernstein-Vazirani algorithm finds in one query [91]. Figure 4.11 shows an example of our circuit implementation of this algorithm using a three-bit string. We use a phase oracle qubit as the black box function encoded with the secret string. Upon measurement of the non-oracle qubits we obtain the secret binary string. We select the Bernstein-Vazirani algorithm because it is implemented using the same gate set we have characterized for the GHZ example, so we do not require additional characterization circuits.

Given the connectivity constraints of the `poughkeepsie` device, the maximum bit string we can test without introducing SWAP operations is of length three. We choose qubits 6, 8, and 12 with oracle qubit 7 because this set has among the lowest error parameters. We execute the Bernstein-Vazirani algorithm for every possible encoding of the three-bit secret string and record the accuracy as the probability that the encoded string was observed. We include collection of these measurements during the same job used to characterize the device.

Figure 4.12 plots the simulated accuracy of the circuit outcome using the fully spatial noise model alongside the experimental accuracy. Our model captures the decrease in experimental observed accuracy across the various binary strings. The loss in accuracy scales with the number of 1 bits in the secret string for both the experiment and simulation. However, the accuracy predicted by simulation is consistently higher than the accuracy observed experimentally, indicating a state-dependent noise source remains missing from this model.

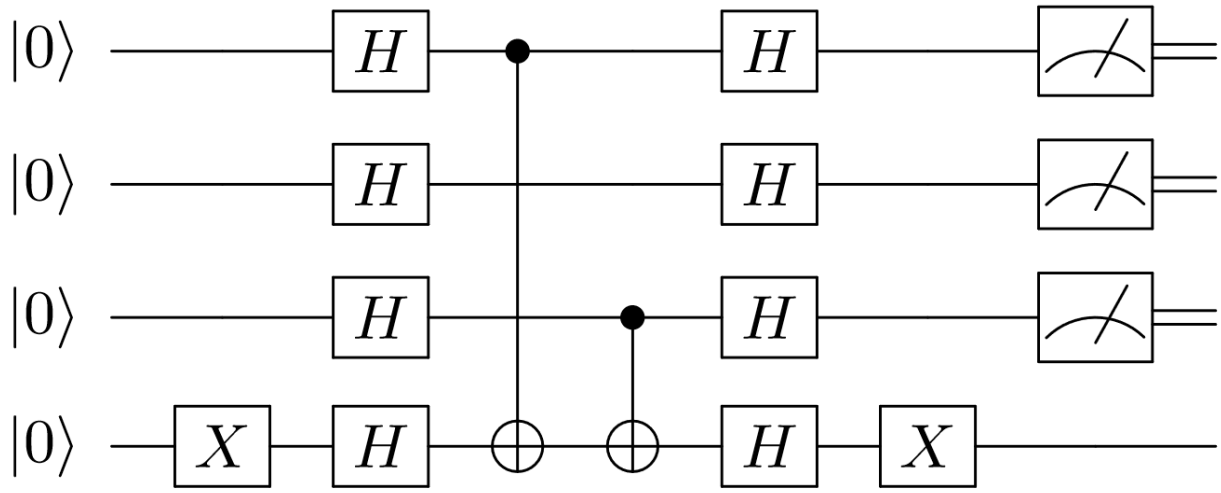

Figure 4.11: Circuit implementation of the Bernstein-Vazirani algorithm. The bottom qubit of the register is the oracle; the top three yield the secret string, here given as 101 as example. Other secret strings are produced by changing the CNOT gate sequence such that control qubits correspond to output bits of 1.

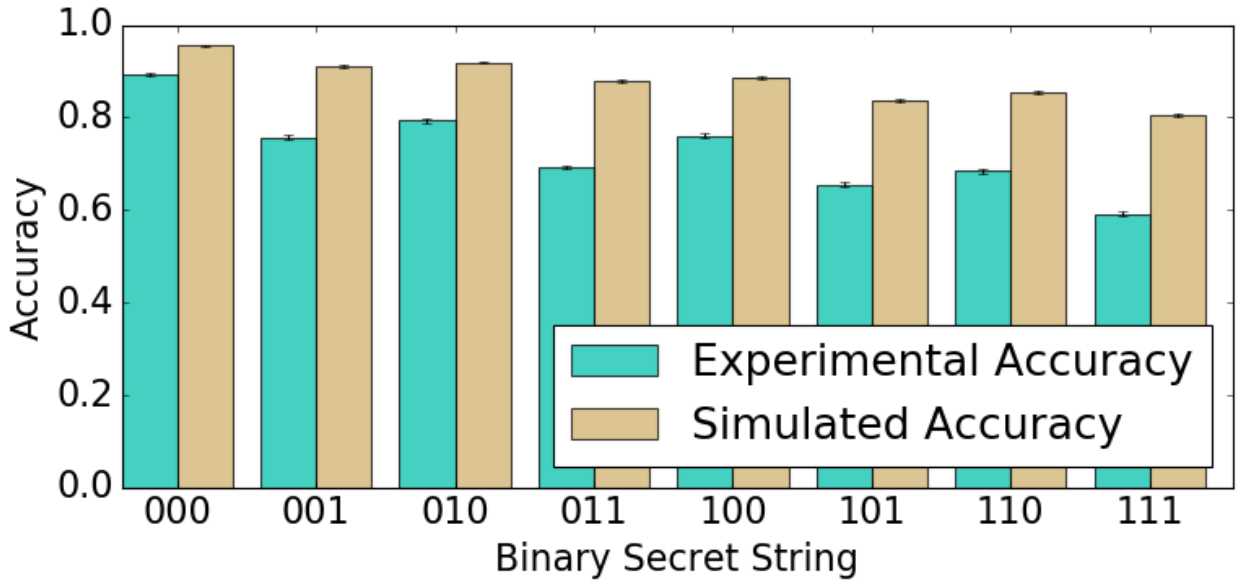

Figure 4.12: Performance of Bernstein-Vazirani algorithm evaluated as the measured probability of the prepared secret string. Simulation is subject to noise defined by the fully spatial model.

4.7 Conclusion

We have presented an approach to noisy quantum circuit modeling based on experimental characterization. Our approach relies on composing subcircuit models to satisfy a desired accuracy threshold, model complexity, and experimental efficiency, which we implement using the total variation distance. We have tested our ideas using the IBM `poughkeepsie` device, which enables evaluation of our characterization methods as well as the comparison of predicted performance for GHZ-state preparation and an instance of the Bernstein-Vazirani algorithm. The initial example focused on GHZ-state preparation examined model fidelity with respect to both width and depth of an input circuit. Models for the readout and CNOT subcircuits accounted for a majority of the model error. Our analysis of a second test circuit using instances of the Bernstein-Vazirani algorithm reveals additional sources of errors not captured in the original GHZ circuit characterization. Because both tests depend on the same gates for state preparation, the appearance of new errors suggests a possible state-dependent noise model that warrants further investigation. While our demonstrations have focused on specific devices and input circuits, the methodology provides a robust and flexible framework by which to generate noisy quantum circuit models on any device.

A significant feature of this approach to noise model decomposition is to iteratively adjust the models until sufficient accuracy is obtained. Improvements in accuracy may be obtained by changing characterization circuits or parameter estimation. The Bell-state and GHZ-state preparation examples demonstrate how this model adjustment may be performed by varying the experimental efficiency and the input to the model to change the accuracy of the final composite model. Our demonstrations have focused on the depolarizing channel for gate modeling, but circuit characterization can be directly extended to account for new noise models, components, applications, and algorithms. For example, in both the GHZ and Bernstein-Vazirani results, we observe an increase in TVD that scales with the number of CNOT gates applied in the circuit. A more sophisticated CNOT noise model may improve accuracy of the final noise model. Since placing limitations on coarse-graining may introduce insensitivities to certain error types, for instance measurement only in the computational basis creates insensitivity to Z error types, it will likely be necessary to refine test circuits to

address more sophisticated models. Additionally, this methodology assumes separability in composition-decomposition, i.e. it assumes that the noise present in the decomposed subcircuits is not substantially different from that of the composed circuit and that any differences may be tuned away by refinement. If this assumption is not true, there may be an upper limit to the achievable accuracy of noise modeling using subcircuit testing. Further model refinement and testing would be necessary to demonstrate this non-separability.

Our original motivation was to address the growing challenge of characterizing NISQ applications, for which efficient and scalable methods are necessary. We have shown how to construct a set of test circuits that scales with the area of the input circuit C and the underlying decomposition strategy. In the GHZ-state preparation example, the number of total experiments needed for full spatial characterization scales with the size of the register q and the number of couplings c according to $N_s(2q+2c+1)$. This resource requirement enables characterization to be run alongside the state preparation circuit when the job is sent to the QPU. This efficiency should help ensure noise characterization is performed within the same processor context as the sought-after circuit. We anticipate such real-time characterizations to be valuable for dynamic compiling and tuning of quantum programs [75, 92, 24].

Our approach to characterization has relied on model selection using minimization of the total variation distance (TVD) between noisy simulation and experimental results. This demonstration used a small set of the possible models for characterizing the observed QPU behavior, and expanding the set of potential models is possible for future work. There is a necessary balance, however, between the sophistication of the model and the utility for characterizing QPU behavior. While fine-grain quantum physical models are capable of capturing a more detailed picture of the dynamics present on small scales, the dawning of the NISQ era requires the addition of new techniques to our toolbox that have a higher-level and larger-scale approach. For scalable numerical analysis of quantum computational methods, it is essential that we develop coarse-grained, top-down approaches to capture the core behavior of QPUs.

Chapter 5

Characterization and Benchmark Experiment Design

5.1 Overview

The methodologies described in Chapters 3 and 4 are components of the toolset for characterization and benchmarking of quantum computers. Our goal is to evaluate the performance of these quantum computing characterization protocols. We test the performance of GST, CB, NR, and EDC on a variety of components and contexts. These protocols are selected because their motivations are different, as are their advantages, disadvantages, and resource consumption, but their outputs are complementary. They have commonalities which we use in developing comprehensive comparisons among these protocols. In particular, they utilize the description of Pauli noise, as described in Chapter 3. We use this common language to design tests which identify the effectiveness of the protocols at characterizing quantum computers.

We design several tests for metrics of interest. The primary metric is the accuracy of each method in capturing the fundamental behavior of the device. We evaluate this in two ways. First, we calculate the distance between the empirical results and results estimated using the selected protocols using noisy simulation with noise models parameterized by the characterization results. We use the total variation distance (TVD) defined in Eq. 4.1 as the metric for this calculation. Second, we evaluate the ability of the protocols to predict

performance of a QPU on a benchmark application. To do this, we identify quantum circuit implementations which are composed of components we have characterized. We gather experimental data for these applications and simulation data under noise models designed from characterization information from each protocol. We compare simulated results to empirical results and evaluate how close our simulation is to experiment using TVD.

Another metric of interest is efficiency, specifically how these protocols scale with the size of the quantum register. The scalability is often based on the dependencies of the algorithm, and the number of quantum experiments needed for each of our selected protocols to characterize a particular gate set on a selected qubit register is known. However, more precisely establishing the tradeoff between experiment count and accuracy of the characterization measured by TVD is a key metric for evaluating these methods. In particular, we measure the relationship between the experiment count of implementations of each protocol and the TVD between these characterizations used in noisy simulation and their associated empirical results. This relationship helps to identify thresholds for the achievable accuracy under a particular experiment count limit, for example, the practical limitation of maximum experiment count per job sent to a QPU.

Classical processing and computing efficiencies are important considerations as well. For instance, classical computational resources are used in processing characterization data to generate protocol output. The efficiency of GST, CB, NR, and EDC is dominated by the quantum computational resources rather than classical computational resources, but classical resource costs may be prohibitive for large quantum circuit simulations and optimization over large data sets, for example.

Our experiment design is outlined as follows.

1. Select characterization protocols–GST, CB, NR, EDC–which generate metrics such as process fidelity and error rates that predict low-level performance.

2. Select a suite of test circuits to characterize. We use Bell-state preparation circuits and GHZ-state preparation circuits.

3. Select a suite of circuits to test the predictive capacity of each protocol's characterization output. We use the Bernstein-Vazirani algorithm, implemented for all accessible secret string encodings.

4. Select QPUs and collect experimental data for each protocol and each application circuit. We use the IBM Q suite of QPUs [3].

5. Analyze characterization data to generate protocol output and noise models. This analysis includes calculating noise parameters that best fit the data and metrics such as process fidelities and noise rates per component, for example.

6. Report on metrics of these results. This includes:

    - Accuracy of noisy simulation based on measured characterization parameters in both application circuit performance and predicted performance in additional applications.
    - Efficiency and scaling of methodology in computational resources, including time, quantum experiments and classical processing and analysis.
    - Effectiveness of the translation of characterization data to a performance benchmark.

5.2 Devices Tested

To gather empirical data for testing our benchmarking protocols we use the IBM Q suite of quantum processors (QPUs) [3]. All of our selected characterization protocols may be straightforwardly executed on any QPU which has a gate-level interface, but we select the IBM suite because they are publicly available and provide an array of QPUs of differing register properties. We focus our experiments on `toronto`, a 27-qubit superconducting transmon device with layout as shown in Fig. 5.1, which has a limit of 900 circuits per job and the option to reserve dedicated time [3]. The relatively large register size of `toronto` compared to other QPUs available makes `toronto` a good choice for testing the scalability of these protocols while also remaining well within the limits of classical simulation of quantum

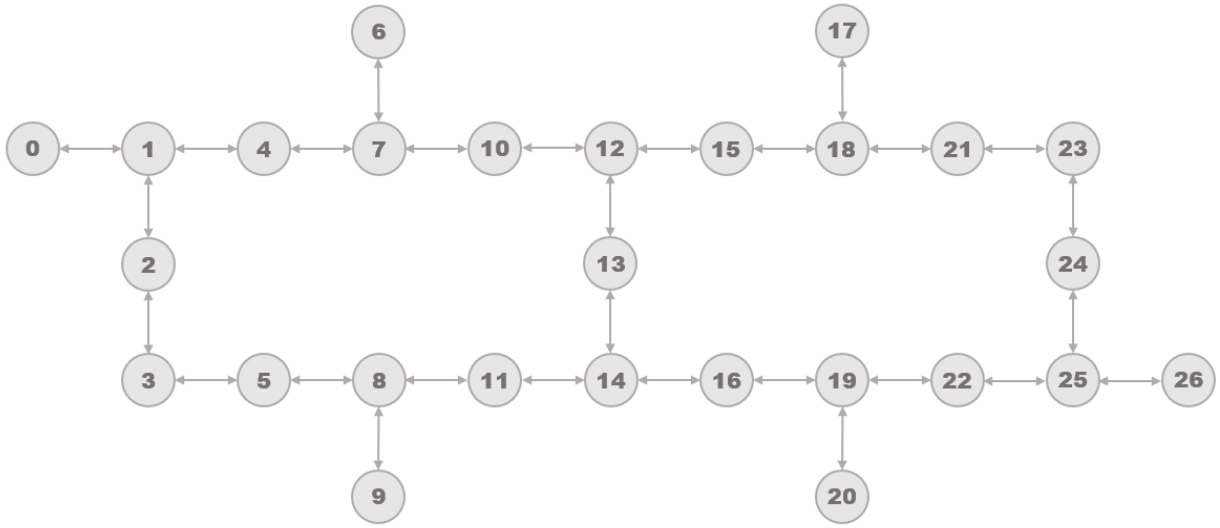

Figure 5.1: A graphical representation of the register layout of the 27-qubit `toronto` QPU at the time of data collection. Each node corresponds to a register element and directional edges indicate the availability of a programmable two-qubit cross-resonance gate.

computers. The importance of a high circuits-per-job limit and dedicated QPU time is to keep a high throughput, which prevents the introduction of drift in the system noise [93].

We map GHZ-state preparation circuits onto `toronto` as illustrated in Table 5.1. This mapping is not unique nor is it optimized for any performance gains. The use of SWAP gates would enable less rigid adherence to the spatial topology of the device, but their addition would likely introduce additional noise sources so we restrict gate selections to the layout as shown in Fig. 5.1.

For the Bernstein-Vazirani (BV) algorithm circuits defined in Section 4.6, we select oracle qubit 25 and secret string encoded qubits 22, 24, and 26. We limit our BV algorithm implementation to a total of 4 qubits because this is the maximum number of qubits we may use without introducing SWAP gates. These qubits are selected because they have comparable or slightly lower error rates than other identically-connected 4-qubit groupings on `toronto` as measured by IBM's routine calibration data [3].

5.3 Characterization Experiments

5.3.1 Empirical Direct Characterization

We utilize a set of quantum circuits for EDC characterization experiments as outlined in Chapter 4. To characterize asymmetric readout, we use circuits of X and XX gates done in parallel and in isolation with one operation per circuit. We also use a blank circuit with no operations which will return a zero state in the absence of noise because IBM QPUs are initialized to the all-zero state. To characterize the error on CNOT gates we use a set of Bell-state preparation circuit tests which are applied to each qubit coupling of `toronto` according to Fig. 5.1. We use the Bell state because it is a subcircuit of the GHZ state and therefore a good candidate to characterize the GHZ-state preparation circuits. We use EDC to characterize GHZ-state preparations of qubit register size 2-27 on `toronto`. This methodology is outlined in Section 4.2.

Table 5.1: GHZ-state preparation circuit mapping onto `toronto`'s topology. For each GHZ size n, the preparation circuit is built by applying the gates of all sizes $[2, n]$ in series.

GHZ Size	**Gate Added**	**Qubit Added**
2	H(0) CNOT (0,1)	0 1
3	CNOT (1,2)	2
4	CNOT (2,3)	3
5	CNOT (3,5)	5
6	CNOT (5,8)	8
7	CNOT (8,9)	9
8	CNOT (8,11)	11
9	CNOT (11,14)	14
10	CNOT (14,13)	13
11	CNOT (13,12)	12
12	CNOT (12,10)	10
13	CNOT (10,7)	7
14	CNOT (7,6)	6
15	CNOT (7,4)	4
16	CNOT (12,15)	15
17	CNOT (15,18)	18
18	CNOT (18,17)	17
19	CNOT (18,21)	21
20	CNOT (21,23)	23
21	CNOT (23,24)	24
22	CNOT (24,25)	25
23	CNOT (25,26)	26
24	CNOT (25,22)	22
25	CNOT (22,19)	19
26	CNOT (19,20)	20
27	CNOT (19,16)	16

5.3.2 Cycle Benchmarking and Noise Reconstruction

Cycle benchmarking and Pauli channel noise reconstruction characterize noise of randomly compiled (RC) circuits. These characterizations rely on Pauli twirling and utilize a similar structure of experiment design for quantum circuit characterization. These are outlined in Sections 3.3 and 3.4.

For both CB and NR, we use the True-Q software to generate circuit collections for execution on IBM QPUs and to calculate the estimated process fidelities from cycle benchmarking (CB) and the Pauli channel descriptions from NR [5]. This software is developed by the company Quantum Benchmark. Noise reconstruction is referred to as k-body noise reconstruction (KNR) in True-Q, so we use KNR for clarity in reporting our results. K-body refers to the number of gates for which an error description is estimated. For instance, a cycle with three parallel gates could be defined with up to $k = 3$. Then if $k = 2$, Pauli channels would be estimated for every two-gate subset within the cycle. We have a software-enforced limit of 20 qubits for experiment design, circuit generation, and results in True-Q, so we limit our experiment design of CB and KNR for GHZ cycles and the RC GHZ circuits to the first 20 qubits of the 27-qubit GHZ mapping we use on `toronto`. Our CB and KNR for BV cycles and RC BV circuits are executed on a 4-qubit subset on `toronto` and therefore do not reach this limit.

We design experiments using KNR to characterize the components of the GHZ-state preparation and BV circuits. Specifically, we use KNR to characterize the Hadamard and CNOT gates for the qubits used in the n-qubit GHZ-state preparation as well as the Hadamard, CNOT, and X gates used in the BV algorithm circuits. The Pauli error rates estimated with KNR can then be used as input to noisy simulation, which we compare to experiment to evaluate the accuracy of the KNR characterization. Similarly, we use CB to estimate the process fidelities of these components.

The experiments for KNR and CB are defined in terms of cycles. Because cycles must be one time step of a circuit, i.e. only one round of parallel gates, we select two different types of cycles to characterize for GHZ. We use a per-gate cycle design which defines one cycle per gate of the GHZ circuit. In the GHZ circuit example, each gate is necessarily a separate time

Table 5.2: Alternate cycles for characterizing the GHZ-state preparation circuits.

Cycle Name	Gates	Gate Count
A1	CNOT (0,1) CNOT (2,3) CNOT (5,8) CNOT (10,7) CNOT (11,14) CNOT (13,12) CNOT (15,18) CNOT (21,23)	8
A2	$H(0)$ CNOT (1,2) CNOT (3,5) CNOT (7,6) CNOT (8,9) CNOT (12,10) CNOT (14,13) CNOT (18,17)	8
A3	CNOT (7,4) CNOT (8,11) CNOT (12,15) CNOT (18,21)	4

step, so this cycle selection is the most natural decomposition for the GHZ-state preparation circuits. This yields a total of 20 unique cycles for our 2-20-qubit GHZ-state preparation circuits. The second set of cycles that we test is a set of maximally parallelized CNOT operations. This yields 3 unique cycles which are listed in Table 5.2. Three cycles is the minimum cycle count that characterizes all gates used in the GHZ circuits. The results from these larger cycles may capture noise effects which are only observable in the context of the other applied gates, so these cycles may provide a helpful description of noise behaviors in the largest systems. However, for building our primary noise models we will use the per-gate cycles so that we may develop a noise model for the device components and model all sizes of GHZ-state preparation circuits.

For Bernstein-Vazirani algorithm circuits, each timestep of the circuit is defined as one cycle. Preparation of secret bitstring encodings uses three CNOT gates which are applied such that the control qubit corresponds to any encoded '1's in the bitstring. These CNOTs are characterized as one cycle each. Every BV secret string encoding is preceded and succeeded by parallel Hadamard gates on every qubit and an X gate on the oracle qubit. The Hadamards are characterized as one cycle together and the X gate is characterized as one cycle.

5.3.3 Gate Set Tomography

Because GST is prohibitively intensive for qubit registers beyond a couple of qubits [39, 29], we will limit characterization with GST to 2 qubits. We can use GST as described in Section 3.1 to characterize a gate set which contains a collection of single-qubit and CNOT gates and use the results to generate a Pauli noise model from the process matrix. This data will represent a standard to which we can compare our other techniques, as GST should yield the most accurate picture of the noise present in the Bell-state preparation example.

To run the GST protocol, we use the python implementation called pyGSTi, which stands for Python Gate Set Tomography Implementation [8]. This implementation provides a software code framework for generating a circuit collection for execution on a QPU and data analysis of quantities of interest including average gate fidelity and estimated process matrices. pyGSTi is developed by a team based at Sandia National Laboratories.

For our GST experiments, we use a standard model within the pyGSTi framework which contains the gate set $\{R_X(\frac{\pi}{2}),R_Y(\frac{\pi}{2}),R_Z(\frac{\pi}{2}),I,\text{CNOT}\}$. We perform the standard GST analysis on our data set (maximum likelihood gate set tomography, or MLGST). This process estimates the gate set that is the best fit to the experimental data by maximizing the log-likelihood with the gate set probabilities [8].

5.4 Benchmark Experiments

5.4.1 Noise Models

For EDC, our estimated noise models include isotropic depolarizing two-qubit channels. This channel ϵ_{DP} is defined in terms of p_{DP} such that

$$\epsilon_{DP}(\rho) = (1 - p_{DP})I\rho I + \frac{p_{DP}}{3}(X\rho X + Y\rho Y + Z\rho Z) \tag{5.1}$$

where $\epsilon_{\text{DP}}^{j,k} = \epsilon_{\text{DP}}^{j} \otimes \epsilon_{\text{DP}}^{k}$ for qubits j, k.

For KNR, our estimated noise models include stochastic Pauli channels of one and two qubits. In all of our experiments, we consider only $k = 1$ because almost all of our cycles are defined with just one gate based on the structure of the GHZ and BV circuits. Incorporating correlated errors among subsets of gates in the cycles with parallelized gates might enhance the detail of the final noise models, but using $k = 1$ is a necessary first step for characterizing all our selected cycles and is most comparable to other methods.

As defined from Eq. 3.2, stochastic Pauli noise channels are of the form

$$\epsilon_{SP}(\rho) = \sum_{P \in \mathbf{P_d}^{\otimes n}} c_P P\rho P^\dagger \tag{5.2}$$

where each P is an n-qubit Pauli matrix with dimension $d = 2$ for qubits. The KNR protocol provides estimates of a set of probabilities c_p and Pauli matrices P which describe the noise of a cycle.

For both EDC and KNR noise models, we also estimate an asymmetric readout channel. The asymmetric readout channel is defined in terms of p_0 and p_1 which are the probability

of a bit flip in the measurement of state 0 and state 1, respectively. The probability of a correct measurement then follows directly as $(1 - p_0)$ and $(1 - p_1)$, respectively.

For GST, we use the Pauli Transfer Matrix (PTM) to describe the noise model of our quantum gates. The elements of the PTM are defined as

$$\mathrm{PTM}_{i,j} = \frac{1}{d}\mathrm{Tr}\{P_i \Lambda(P_j)\} \tag{5.3}$$

for dimension $d = 2$, Pauli matrices P and quantum operation Λ. This PTM represents the noisy gate and can be applied directly in simulations.

Gate set tomography also estimates state preparation and measurement (SPAM) errors. For two-qubit tomography experiments these SPAM parameters provide estimates of error on each two-bit measurement output in a 4x4 matrix. The matrix elements represent the probabilities of measuring each classical two-bit outcome given an expected outcome.

5.4.2 Simulation Methods

For our simulations, we use Qiskit Aer [94]. Aer is a quantum circuit simulator which can simulate ideal or noisy quantum circuits with a variety of methods. For our simulations we use Aer's statevector simulator which simulates quantum circuits by applying operators to the statevector which describes the quantum state of the qubit register. It can simulate any of the gates and noise models that we use for our tests but the size of the computation scales exponentially in the size of the qubit register. Consequently, for our GHZ-state preparation circuits with register sizes around 20+ qubits we use the Aer statevector simulator on the IBM Q backend. This is a dedicated classical computing resource which is optimized for quantum circuit simulation such that large simulations can be completed more quickly than on a personal computer.

We model the noise in quantum circuits as an ideal quantum operator followed by a noise operator which represents the noise associated with the ideal operator when applied in experiment. This is a common but not unique method to describe noise in quantum systems [90]. The quantum error functions that are native to the Aer simulator methods utilize this

expression of quantum noise. We define our noise models in the Aer framework to implement them in simulation.

For simulations of the Bell-state preparation circuit using the GST estimated noise model, we use the pyGSTi simulation capability. pyGSTi supports quantum circuit simulation that uses the estimated model results calculated directly from the GST protocol. Because GST reports a more complex model of the characterized gate set than the other methods, simulating the Bell state directly in pyGSTi provides the most accurate translation of GST model results to circuit outcomes.

Because GST simulations are limited to the two-qubit example, we do not simulate the GHZ or BV circuits using the GST model. For the Bell-state preparation circuit, our GST model defines a noisy CNOT gate and a noisy Hadamard gate which is defined as a decomposition into a rotation about Y by $\pi/2$ and two rotations around Z by $\pi/2$. The GST model also includes the state preparation and measurement error which maps the probability of every two-qubit input state to be observed as each two-qubit output state.

The code and data used in these experiments can be found at the public repository [95].

5.4.3 Application Testing

The outcome of any measured quantum circuit is a bitstring of zeroes and ones. To evaluate the distance between two distributions of bitstring outcomes, we use the total variation distance (TVD). The TVD is given by

$$d_{\mathrm{TV}}(H, M) = \frac{1}{2} \sum_{k} \left| r^{(H)}(k) - r^{(M)}(k) \right| \tag{5.4}$$

for two distributions H and M with probability r of state k, just as in Eq. 4.1. The probability r is calculated by the number of times the state k is returned divided by the total number of measurements which comprise the distribution.

Error propagation in the TVD calculation is given by

$$\delta\mathrm{TVD} = \frac{1}{2} \sqrt{(\delta\alpha_i)^2 + (\delta\alpha_j)^2 + (\delta\beta_i)^2 + (\delta\beta_j)^2 + \ldots} \tag{5.5}$$

for states α, β, ... of two distributions labelled i and j. The error of each state is given by

$$\delta\alpha = \sqrt{\frac{p(1-p)}{N}} \tag{5.6}$$

for probability p of measuring the state out of N total measurements.

The Bernstein-Vazirani (BV) algorithm is our selected application test. The circuits which implement the BV algorithm utilize a gate set closely related to the GHZ-state preparation circuits. We use this algorithm as a benchmark of performance. The output of a BV circuit in the absence of noise is the encoded secret string, so we compare the accuracy of our noisy simulation in returning the encoded secret string to the accuracy obtained in experiment from the QPU. The accuracy is defined as the number of times the encoded secret string is observed out of the total shot count of the circuit. This provides a means to benchmark the noise models used in simulations–the closer the accuracy agrees with experiment, the more likely the noise model accurately describes the QPU.

Chapter 6

Characterization and Benchmark Results

We report results of characterization and performance testing using our selected methodologies as presented in Chapter 5. We executed GST, CB, NR, and EDC protocols on `toronto` over a 12-hour period of dedicated QPU time on February 14, 2021. We executed the GST circuits first. Next we ran KNR and CB experiments for GHZ cycles followed by the RC GHZ circuits. Then we executed the circuits for KNR and CB for the BV cycles followed by the RC BV circuits. Interspersed among these were multiple runs of EDC circuits. Uncompiled GHZ and BV circuits were included in the jobs that execute EDC circuits. We refer to these uncompiled circuits as bare circuits (BC).

6.1 Quantum Resources Usage

A central feature of characterization methods is their resource use and scalability. In Table 6.1, we summarize the resource requirements of our experiments, in particular the amount of time taken to acquire results and the size of the computational jobs. All quantum experiments are sent to IBM Q devices as jobs with a limit of 900 circuits per job. The number of shots per circuit on these devices is limited to 8192. Because data was taken during a 12-hour window of dedicated QPU time, there were no queue wait times for any experiments. We record the amount of time taken for an experiment set as the wall clock time

Table 6.1: Quantum resources used in our selected protocols for all experiments executed. Sequence lengths in CB and KNR are the experimental values of m from Fig. 3.3.

Method	**Details**	**Circuits**	**Shots**	**Time**
GST	2-qubit gate set $\{R_X(\frac{\pi}{2}), R_Y(\frac{\pi}{2}), R_Z(\frac{\pi}{2}), I, \text{CNOT}\}$	20094	1024	2.28 hours
CB (GHZ)	Per-gate cycles (20) Sequence lengths 4,12	10440	128	25 minutes
KNR (GHZ)	Per-gate cycles (20) Sequence lengths 4,12	10440	128	17 minutes
CB (GHZ)	Parallelized cycles (3) Sequence lengths 4,12	3300	128	16 minutes
KNR (GHZ)	Parallelized cycles (3) Sequence lengths 4,12	1620	128	11 minutes
RC GHZ	2-20-qubit GHZ circuits compiled into 32 RC circuits each	608	128	12 minutes
CB (BV)	Time step cycles (5) Sequence lengths 4,12	2520	128	9 minutes
KNR (BV)	Time step cycles (5) Sequence lengths 4,12	1980	128	6 minutes
RC BV	All 3-bit strings compiled into 32 RC circuits each	256	128	6 minutes
EDC	27-qubit characterizations	205	8192	11 minutes
BC GHZ	2-27-qubit GHZ circuits	26	8192	
BC BV	All 3-bit strings	8	8192	

from the creation of the first job containing experiments for the protocol to the completion of the last job containing experiments for the protocol. In the cases of CB, KNR, and RC experiments, the job creation and validation are parallelized by the True-Q software interface which substantially decreases the total time taken for these experiments compared to that of GST and EDC.

For GST, the 2-qubit 5-gate set we characterize is computationally expensive. The qubit count and gate count are the primary drivers of the total experiment count necessary to build a GST estimate. For instance, reducing this to just a single qubit example of the same gate set without CNOT would reduce the circuit count by ten times. We use a shot count of 1024 which is the default shot count setting and generally ensures sufficient statistics as shown in Appendix A.

For KNR and CB, the primary factors which determine computational expense are the sequence lengths and the number of cycles. There are 20 cycles needed to characterize every component of the GHZ circuits, so we utilized a minimum sequence count to keep resource costs manageable. We use sequence lengths of 4 and 12 because the error rates of CNOT gates tend to be high so the performance degrades after a short sequence of gates. The error bars on these estimates are consequently larger however, as a result of fewer data points to fit the decay curve over multiple sequence lengths. We use a shot count of 128 because these protocols calculate estimates based on the decay functions, so the sampling size of each individual data point may be reduced [5]. This also helps to manage the resource cost.

The EDC circuit count includes the circuits which we use to characterize readout errors which are applied for both EDC and KNR noise models. The EDC circuit count increases linearly with the number of qubits and the number of operators to characterize. We use the maximum shot count for these tests because there are few enough circuits that the resource cost is still low.

6.2 Experimental Data

We next report device characteristics across the time period these experiments ran. IBM Q devices are periodically calibrated, and this calibration includes sampling for measurement

error approximately every hour. The results of these tests inform the calculation of the discriminator plane that distinguishes a measurement result of 0 from a result of 1 [96]. Figures 6.1, 6.2, and 6.3 show results from EDC readout error analysis using the methodology outlined in Chapter 4.

We use this data to inform our selection of data set for further analysis. We avoid data sets with outliers in the error results such as those from 0805 or 1917. We also want to select a GHZ-state preparation data set with good performance as measured by close agreement to an ideal GHZ-state preparation, as this would be a typical first-pass approach for running quantum algorithms to select the best available data. We then select the readout data closest to the GHZ-state performance data we select.

Figures 6.4 and 6.5 show the performance of RC and BC GHZ circuits across the collected data sets reported as the rate that the all-zero and all-one states were observed in experiment. For BC GHZ circuits this rate is calculated out of a total experiment count of 8192, and for RC GHZ circuits this rate is calculated out of a total experiment count of 4096 from 32 randomly compiled circuits with 128 shots each.

From these results, we see that in the BC GHZ circuit performance there are decreases in the rate of expected outcomes at register sizes 10 and 16. These correspond to the addition of a particularly noisy qubit in the entangled GHZ-state–for size 10, qubit 13 is added and for size 6 qubit 15 is added. From the readout error rates shown in Figs. 6.1 and 6.2 we can see that these two qubits are well above the typical error rates of the register components. In the RC GHZ-state preparation circuits these decreases are not as noticeable, which may indicate that randomized compiling has the effect of reducing the impact of outlier qubits.

For our selected RC and BC GHZ-state preparation circuits, we plot the TVD between the experimental results and noiseless results, i.e. ideal outcomes, in Fig. 6.6. For the noiseless case of GHZ-state preparations, we use an equal split of $N_s/2$ counts in state $|0_0, ..., 0_n\rangle$ and $N_s/2$ counts in state $|1_0, ..., 1_n\rangle$ for N_s total shots and n qubits. From these results, we see that the BC circuits are closer to the ideal outcomes than the RC circuits because their TVD remains closer to zero. This is likely because injecting twirling gates in the GHZ circuits can lead to a dramatic increase in the total gate count, which in this instance is most likely increasing the overall error rate of the circuits. The randomized compiling protocol

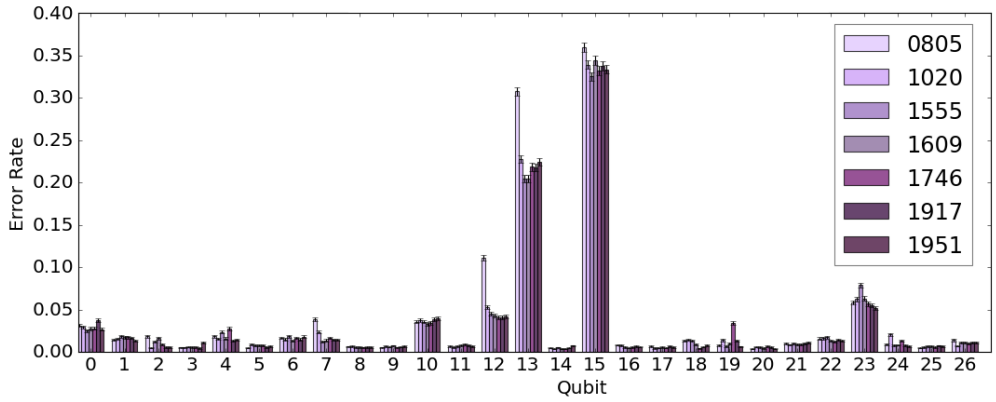

Figure 6.1: Error rate in readout of state 0 on `toronto` using the EDC methodology.

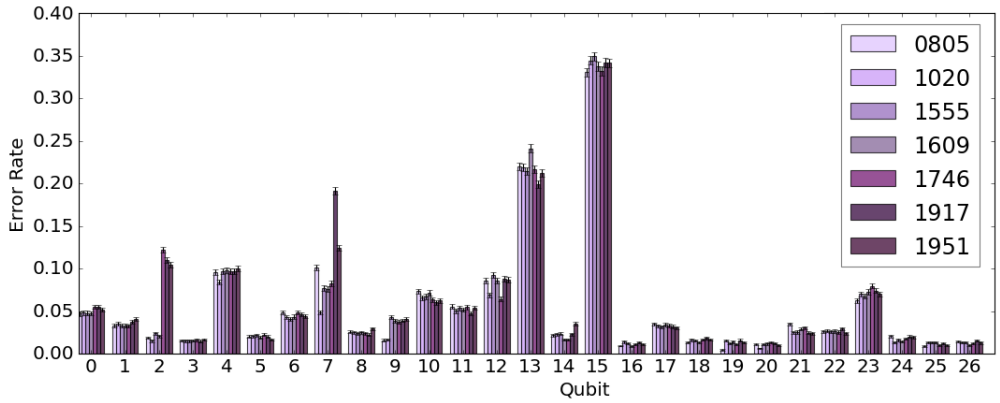

Figure 6.2: Error rate in readout of state 1 on `toronto` using the EDC methodology.

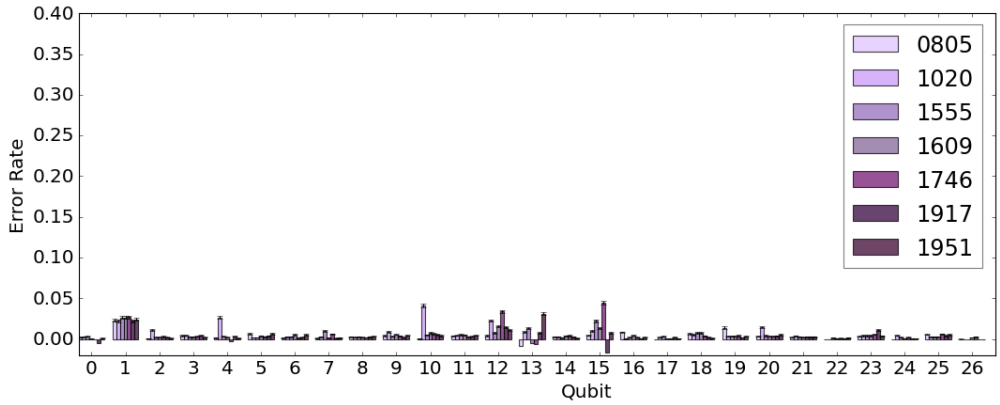

Figure 6.3: Depolarizing error rate for the X gate on `toronto` using the EDC methodology.

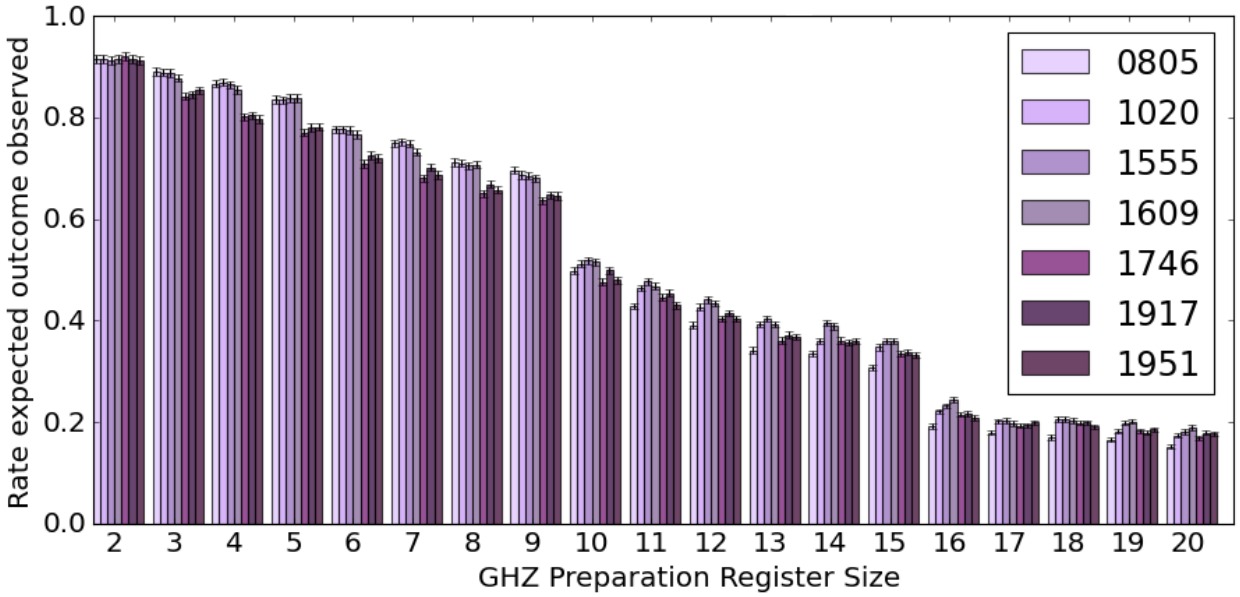

Figure 6.4: Rate expected outcomes were observed from BC GHZ-state preparation circuits executed on `toronto` across a 12-hour period. The decay function of the best performing set (1555) is $1.112e^{-0.0834x}$ with $R^2 = 0.989$ for register size x [7].

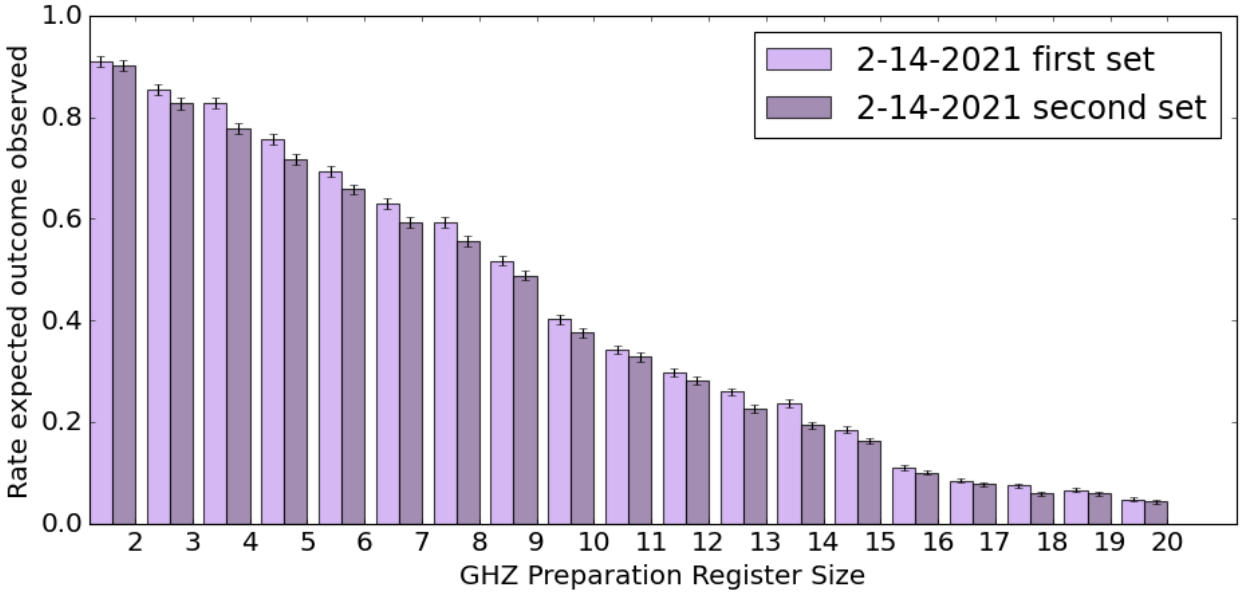

Figure 6.5: Rate expected outcomes were observed from RC GHZ-state preparation circuits executed on `toronto` across a 12-hour period. Decay function of the best performing set (first set) is $1.148e^{-0.12x}$ with $R^2 = 0.9855$ for register size x [7].

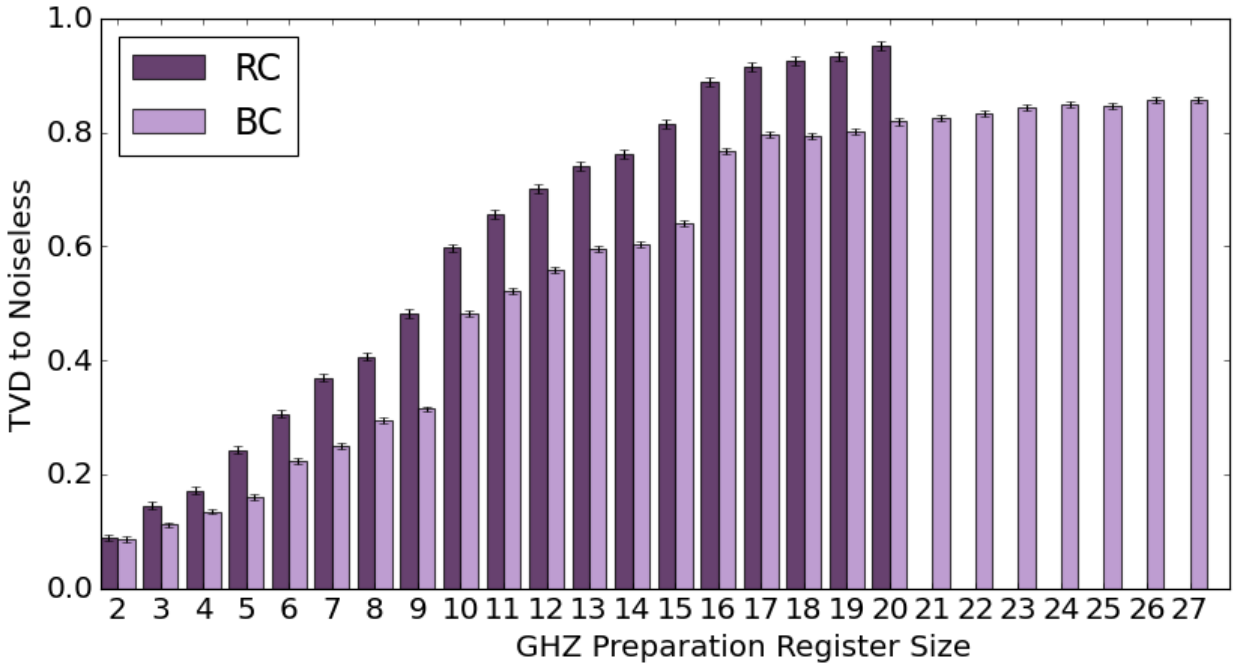

Figure 6.6: TVD between experiment results and noiseless GHZ-state preparations. RC circuits are limited to the first 20 qubits. The RC TVD increases with register size x as $0.2418e^{0.0801x}$ with $R^2 = 0.978$ and the BC TVD increases as $0.2625e^{0.053x}$ with $R^2 = 0.963$ [7].

compiles the twirling gates with neighboring single-qubit gates, but in the case of GHZ-state preparation all circuits consist only of two-qubit gates which are all twirled around. This is corroborated by the close agreement of RC and BC TVDs for the smallest GHZ states, when the total number of added twirling gates is lowest compared to the total gate count of the uncompiled circuit.

In Fig. 6.7 we show the TVD between experiment results and noiseless GHZ-state preparations trimmed to only qubits 0 and 1 such that the full bitstring of each state observed is classified by the first two bits. We show these results to address the probabilistic decrease in observing a fully all-zero or all-one state from the largest register sizes. It becomes more likely that at least one bit of the bitstring outcome is flipped due to an error as the measured register size increases. The TVD of these results is reduced over the TVDs shown in Fig. 6.6, which may also suggest that larger qubit registers correlate with higher error rates in experiment. However, the TVD increases steadily for larger sizes of GHZ-state preparation, which may capture the effects from decoherence on the first two qubits which idle while CNOT gates are performed on the other qubits of the register.

6.3 Characterization Results

We report the characterization results of GST, CB, KNR, and EDC of our experiments.

6.3.1 Gate Set Tomography

Gate set tomography provides a detailed picture of the characterization of a defined gate set on a selected qubit subspace. We executed GST on qubits 0 and 1 of `toronto` and obtain estimates of SPAM operators and the gates $\{R_X(\frac{\pi}{2}), R_Y(\frac{\pi}{2}), R_Z(\frac{\pi}{2}), I, \text{CNOT}\}$, as well as estimates of the model fit and metrics of gate performance such as process fidelity. We calculate the completely-positive trace-preserving (CPTP) map that best fits the GST experiment data [8, 26]. The CPTP estimate results are about 45 standard deviations away for the shortest circuits of length 1 and 2 gates and about 250 standard deviations away from a Markovian gate set for the longest circuits of length 32 gates. This indicates the presence of non-Markovian noise, especially for longer gate sequences.

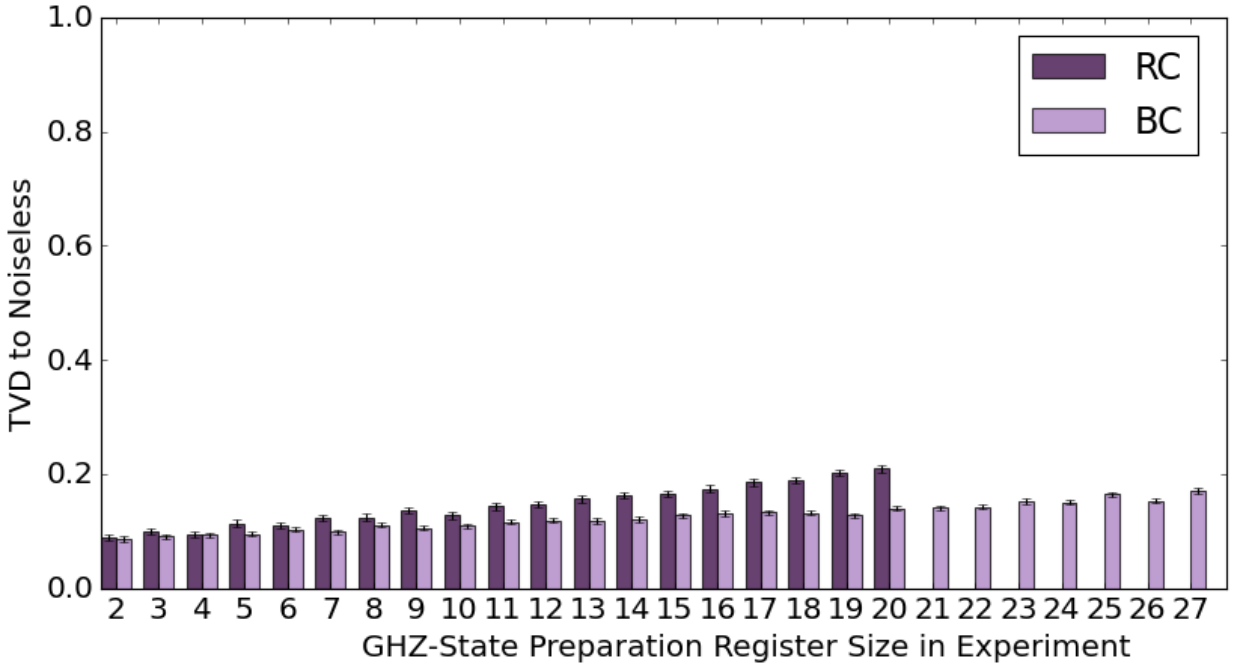

Figure 6.7: TVD between experiment results and noiseless GHZ results. For each size of GHZ-state preparation, the results are trimmed to the first two qubits. The RC TVD increases with register size x as $0.0906e^{0.0445x}$ with $R^2 = 0.999$ and the BC TVD increases as $0.0889e^{0.024x}$ with $R^2 = 0.999$ [7].

In Fig. 6.8 we show the GST estimate of the Pauli Transfer Matrix (PTM) for the CNOT gate. The PTM represents the implementation of the operator in experiment. The ideal PTM consists of a single value of 1 or -1 in each row and column–noise manifests in the PTM in the non-zero terms which are lightly shaded in Fig. 6.8.

Figure 6.9 shows the SPAM estimates from the GST model. We present the matrix of values which represent the probabilities of observing versus preparing each two-qubit state. The highest error rates are observed in the 11 state and the lowest are observed in the 00 state. This readout model inherently accounts for correlations in the two qubits by separately estimating the error on each two-qubit state.

6.3.2 Cycle Benchmarking and Noise Reconstruction

Cycle benchmarking protocols return the process infidelity of a cycle in the context of randomized compiling. The process infidelity e_F is related to the process fidelity F defined in Eq. 3.6 as $1 - e_F = F$ [5]. Table 6.2 presents the results of CB protocols executed for cycles defined per gate and cycles with maximally parallelized gates for the gates used in the GHZ circuits. We also calculate the estimated process fidelity of the larger cycles (A1, A2, A3) based on multiplying together the process fidelities of the process fidelities measured in experiment for the constituent gates. The multiplied process fidelities are generally higher than the process fidelity observed in experiment, which indicates that the larger cycles are capturing additional noise sources that would not be predicted by only considering the process fidelities of the components.

From KNR, we obtain the full stochastic Pauli channel estimated for the specified cycle. In Fig. 6.10 we show the error rates estimated using KNR where we have defined each cycle as a CNOT gate operating on a coupling on `toronto`. Some error types are indistinguishable in the KNR protocol for certain gates because the errors operate in the same way on the cycle of interest. For example, a CNOT gate cycle KNR result conflates IY and ZY Pauli errors. To construct our noise model, we preferentially select weight-one errors (any two-qubit Pauli operator that has an I operator) where possible under the assumption that weight-one errors are more likely and assign the reported error probability to that error type. For indistinguishable weight-two errors there is no guiding principle for which error is more

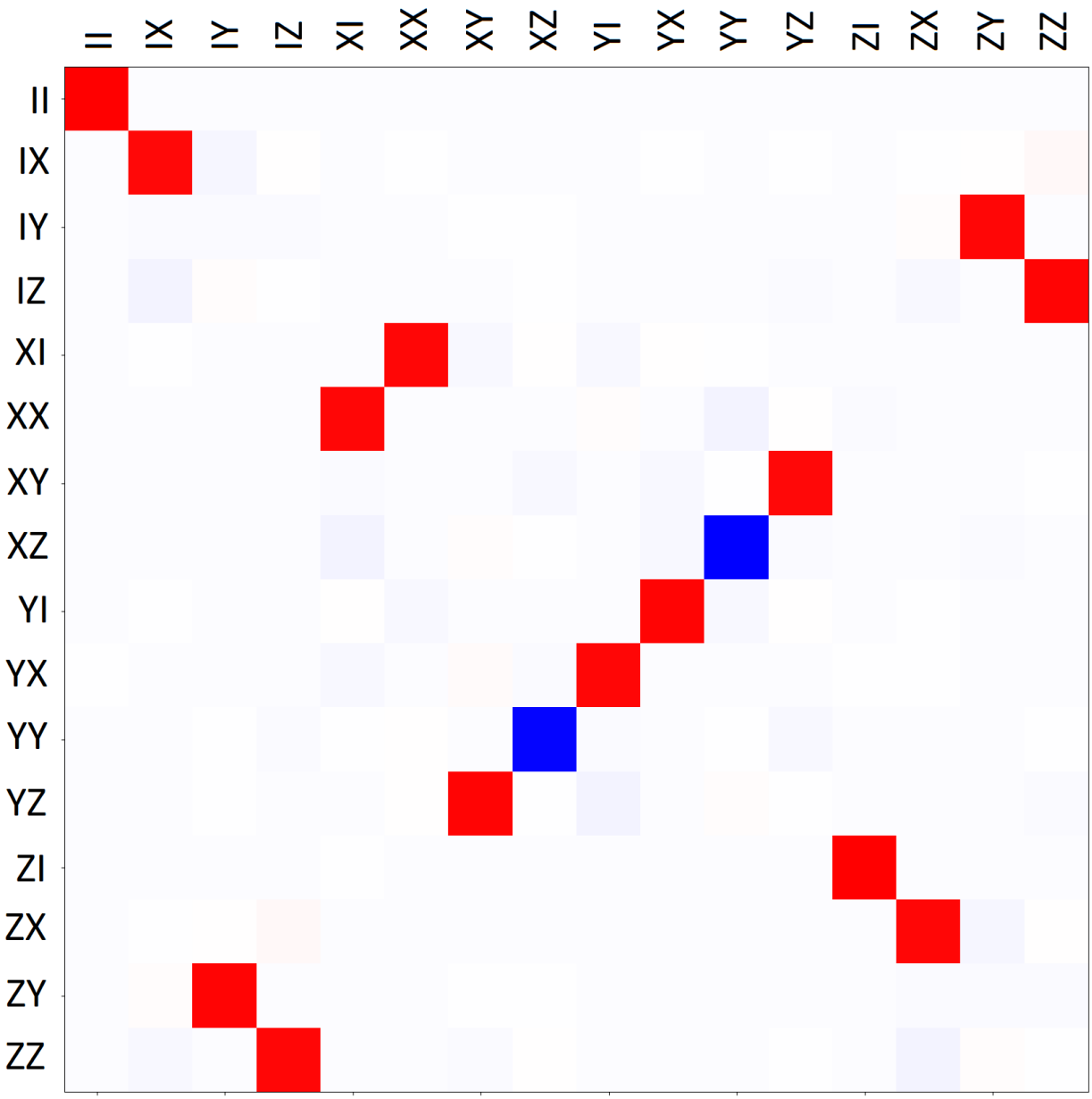

Figure 6.8: The Pauli transfer matrix estimated for CNOT on qubits 0 and 1 from GST [8]. The color scale ranges from red for values close to 1 and blue for values close to -1.

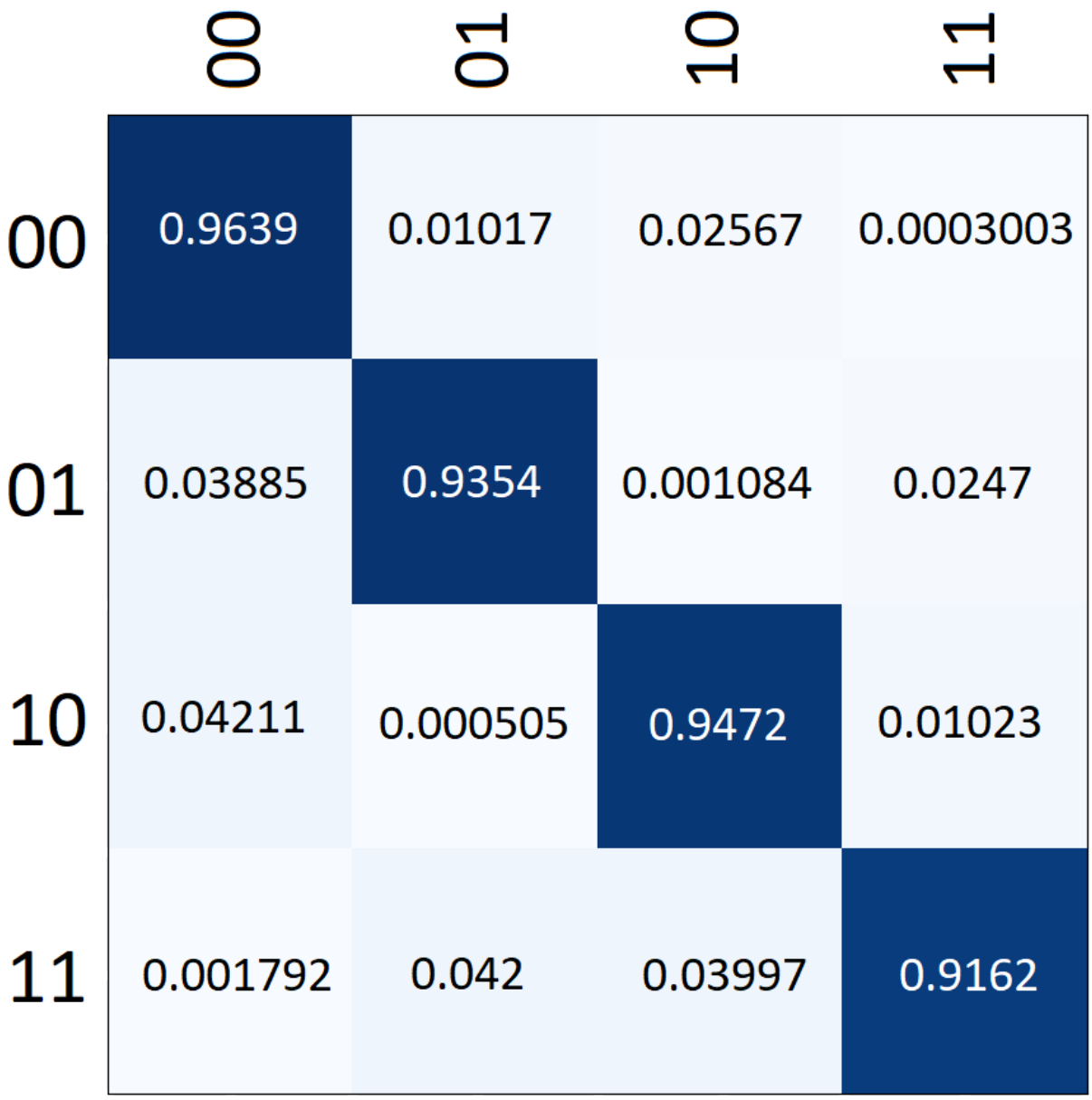

Figure 6.9: Readout matrix representing results from GST SPAM estimates.

Table 6.2: Process fidelities measured by CB using the True-Q software framework.

Cycle	**Gates**	**Experiment F**	**Multiplied F**
0	H(0)	0.99903	
1	CNOT (0,1)	0.96539	
2	CNOT (1,2)	0.98663	
3	CNOT (2,3)	0.98550	
4	CNOT (3,5)	0.98404	
5	CNOT (5,8)	0.99011	
6	CNOT (8,9)	0.98861	
7	CNOT (8,11)	0.99088	
8	CNOT (11,14)	0.98688	
9	CNOT (14,13)	0.98395	
10	CNOT (13,12)	0.97753	
11	CNOT (12,10)	0.98632	
12	CNOT (10,7)	0.98912	
13	CNOT (7,6)	0.99267	
14	CNOT (7,4)	0.98442	
15	CNOT (12,15)	0.98174	
16	CNOT (15,18)	0.97980	
17	CNOT (18,17)	0.97658	
18	CNOT (18,21)	0.98206	
19	CNOT (21,23)	0.97807	
A1	CNOT (0,1) CNOT (2,3) CNOT (5,8) CNOT (10,7) CNOT (11,14) CNOT (13,12) CNOT (15,18) CNOT (21,23)	0.77743	0.86137
A2	$H(0)$ CNOT (1,2) CNOT (3,5) CNOT (7,6) CNOT (8,9) CNOT (12,10) CNOT (14,13) CNOT (18,17)	0.62929	0.90214
A3	CNOT (7,4) CNOT (8,11) CNOT (12,15) CNOT (18,21)	0.92761	0.94045

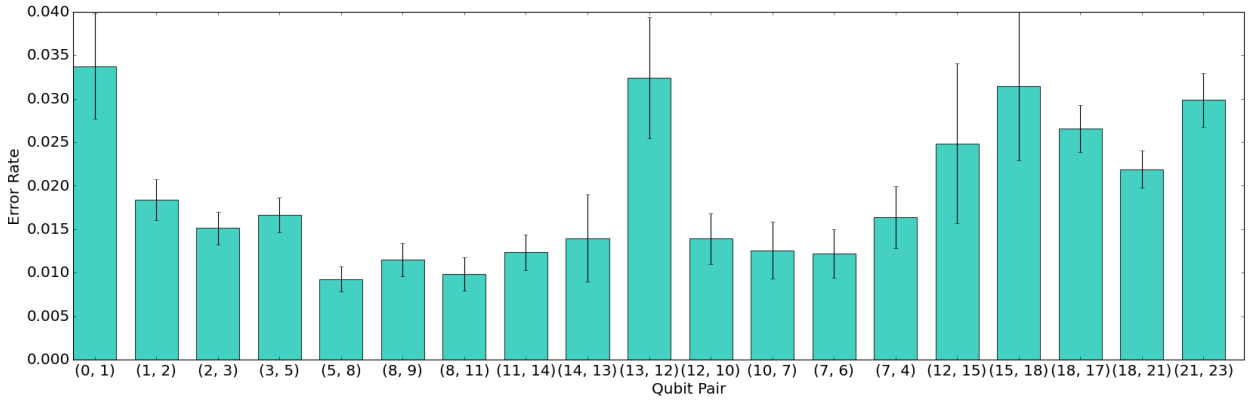

Figure 6.10: Total error rates for each CNOT characterized by KNR. Values represent the sum of all error types measured by the protocol.

likely, so we select the first reported error of the two. Because the errors are lexicographically ordered there is a slight bias towards X-type errors, but we expect this to have little to no effect on the final results of the noise model since these error types are indistinguishable in practice.

In Fig. 6.11 we show the estimated portion of these errors which could be modeled as a depolarizing channel. To estimate this parameter, we consider a single-qubit depolarizing channel like the one defined in Eq. 4.5 except that the parameter p is allowed to vary per error gate operator (X, Y, or Z). We sum together the single-qubit (weight-one) error rates per qubit provided by the KNR estimate and average the two estimates together. This is an approximation of a depolarizing parameter that could describe the noise in a two-qubit gate as the channel $\epsilon_{j,k}^{\mathrm{DP}} = \epsilon_j^{\mathrm{DP}} \otimes \epsilon_k^{\mathrm{DP}}$ in the same way EDC depolarizing estimates are defined. Several degrees of freedom that are estimated by KNR are ignored in this approximation but it is a useful comparison to the EDC-estimated depolarizing rates.

6.3.3 Empirical Direct Characterization

In Fig. 6.12 we show estimated readout error rates for measurements of state zero and state one. The p_0 parameter is the rate of error in readout when state zero was the expected outcome; similarly the p_1 parameter is the rate of error in readout when state one was the expected outcome. These parameters are estimated from results of a blank measurement circuit and a circuit with a single X gate applied to every qubit in parallel. We also test other methods of readout parameter estimation using one X gate operation per qubit per circuit and adding circuits which use two X gates to solve for error rates on X such that the error rate p_1 is corrected for the error of applying X. However, this method provided the best performance in our tests which are shown in detail in Section 6.4. These readout error estimates are also used in the KNR noise model, as this method for estimating readout is the same approach used in True-Q.

The readout error rates indicate spatial variability across the qubit register, as well as a consistent asymmetry between states zero and one. In particular, most qubits have a higher

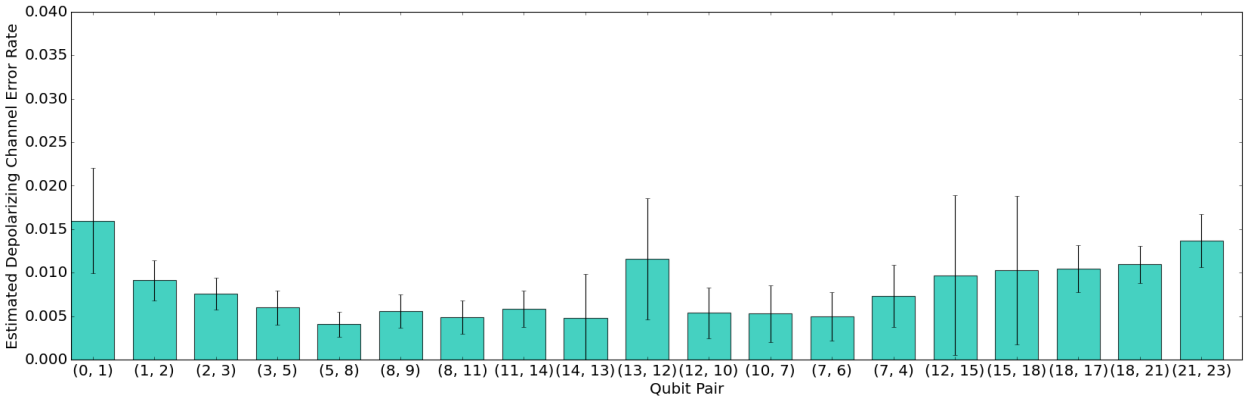

Figure 6.11: Estimate of the depolarizing component of the noise channels estimated by KNR.

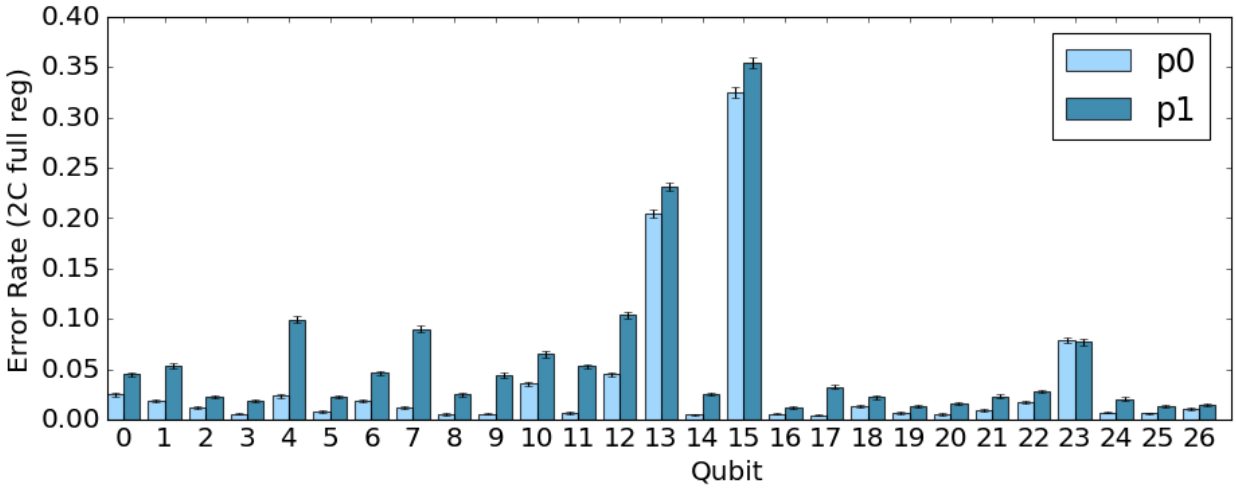

Figure 6.12: Readout error rates for `toronto` estimated by EDC.

error rate in readout of state one. Additionally, most qubits have under 5% error rates, but qubits 13 and 15 are significant outliers with around 30% error.

Using EDC we calculate the depolarizing parameter which best fits Bell-state preparation circuit outcomes for Bell circuits executed on each qubit pair of the layout of `toronto` shown in Fig. 5.1. In Fig. 6.13 we show these parameters which are evaluated for CNOT gates applied with both configurations of control and target qubits. The error bars represent the upper limit of the error from the least squares calculation. These error rates are calculated using the readout error rates from Fig. 6.12.

The results from EDC for depolarizing error rates show lower error rates and more spatial variability than the depolarizing estimates derived from KNR. The estimated error from EDC is frequently lower than the KNR depolarizing estimate, and the relative noise of the qubit couplings among each protocol estimate does not generally agree. The differences between the two estimates may largely be attributed to the use of RC in the KNR estimates, and this comparison provides a numeric estimate of the effect of RC on the observed error rates.

6.4 Comparative Analysis of Characterization

In Table 6.3 we report the process fidelities reported from CB results and the Hellinger fidelities calculated from the GHZ-state preparation results. The Hellinger fidelity is defined as

$$F_H = \sum_i \sqrt{p_i q_i}^2 \tag{6.1}$$

for states i of distributions p and q. The Hellinger fidelity is a metric which provides a basic comparison of the circuit results in experiment to the estimated process fidelities of each cycle in experiment. For each example shown in Table 6.3 the Hellinger fidelity is lower than the process fidelity of the constituent circuit components, and the difference between these metrics may largely be attributed to the effects of readout error. For the largest example of 20-qubit GHZ-state preparation, the process fidelity of the parallelized cycles (A1, A2, and A3) is closer to the Hellinger fidelity of the GHZ circuit in experiment than the constituent cycles. This underlines the value of characterizing cycles rather than components. However,

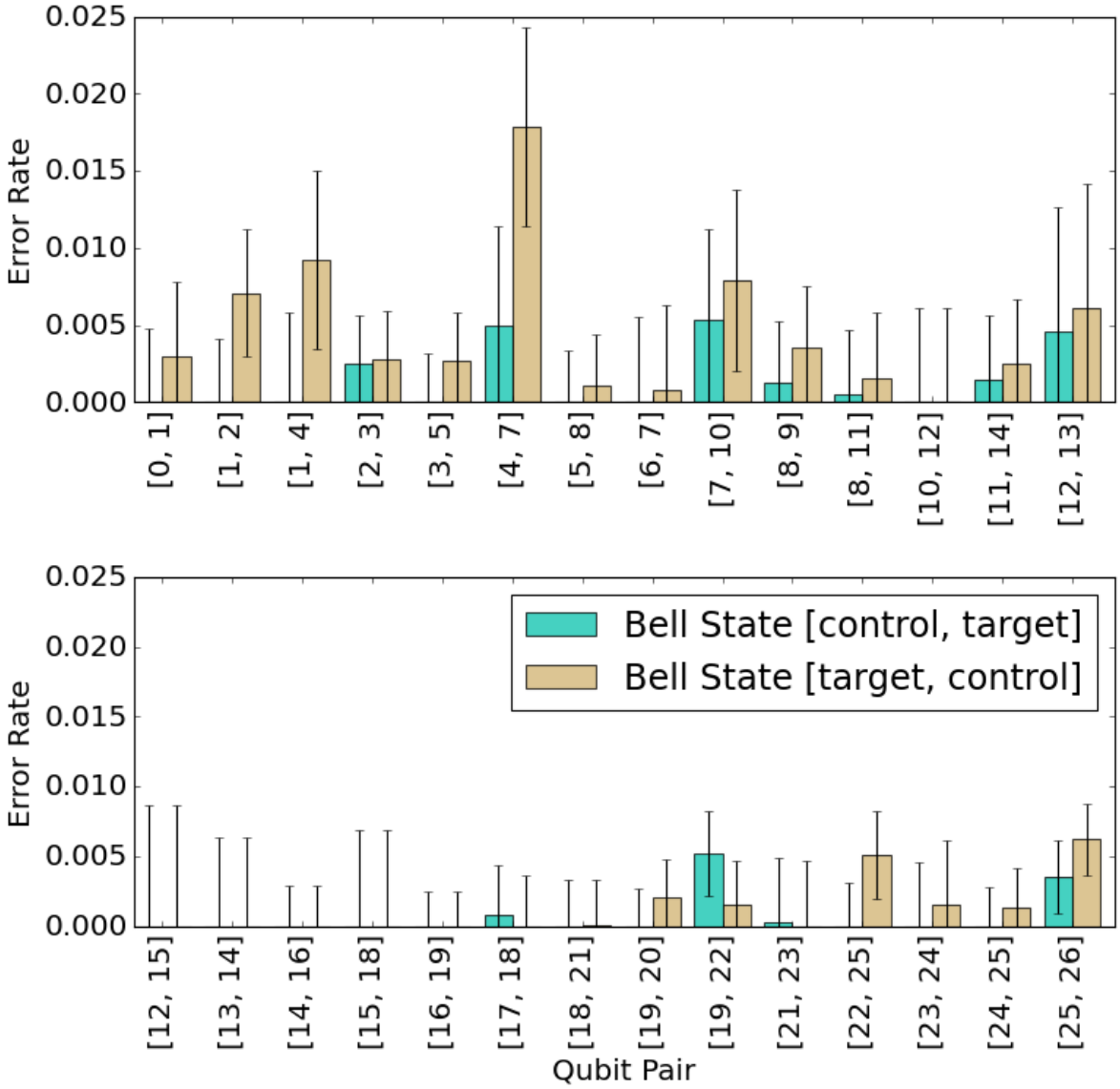

Figure 6.13: Error rates of CNOT for `toronto` estimated by EDC.

Table 6.3: Comparison of estimated process fidelities using GST and CB. Square brackets indicate cycle fidelity estimates that are multiplied together.

Method	**Fidelity**
BC Bell state circuit qubits (0,1)	0.81767 (Hellinger)
GST CNOT qubits (0,1)	0.97692 (process)
RC Bell state qubits (0,1)	0.78493 (Hellinger)
CB H qubit (0)	0.99903 (process)
CB CNOT qubits (0,1)	0.96539 (process)
CB [H qubit (0)] [CNOT qubits (0,1)]	0.96445 (process)
RC GHZ 20 qubits	0.11528 (Hellinger)
CB cycles [A1][A2][A3]	0.45381 (process)
CB cycles [0][1][2]...[18][19]	0.73081 (process)

it also highlights the difficulty in defining cycles for circuits that do not have an inherent cyclical structure. For GHZ circuits, the natural cycle definition is one cycle per gate. Our parallelized cycles ignore the actual structure of GHZ circuits in favor of characterizing CNOT gates in the context of a large set of CNOT operations at once. While this provides a closer estimate of fidelity in experiment than the natural cycle definition of GHZ, it still does not come close to the final circuit fidelity in experiment. This could be because the cycles A1, A2, and A3 are not representative of operation sequences used in GHZ circuit experiments.

In Fig. 6.14 we show the results of simulating a Bell-state preparation circuit using noise models derived from KNR results. We use the gate noise estimates from KNR shown in Fig. 6.10. The "Gate Only" noise model consists of just these error rates. We then add to this gate model four different methods of readout error. The readout error estimates are derived from using a single blank measurement circuit, a circuit with a single X gate applied per qubit, and a circuit with two X gates applied per qubit. The "2C full register" readout model uses the first two of these circuits to estimate readout. The "3C full register" readout model uses all three of these circuits to estimate readout. We can apply these X and XX gates once per qubit per circuit such that we have as many circuits as qubits. The motivation of this approach is to take any correlations between simultaneous operators into account. This approach is used in the "2C per qubit" and "3C per qubit" models.

In Fig. 6.15 we show these results for EDC models with the same set of four readout models. We use the gate noise estimates from Fig. 6.13. For both the KNR and EDC results, we compare the performance of these noise models to the TVD between the experiment results and a noiseless Bell-state preparation, which is an exactly equal split between the 00 and 11 states, just as we defined the noiseless GHZ state.

In Fig. 6.16 we show the results of simulating a Bell-state preparation circuit using the best noise model from EDC, KNR, and GST. We again compare to the TVD of the noiseless Bell-state preparation of equal counts of state 00 and state 11. We also compare to "self-simulated" cases, which is the TVD between the targeted Bell circuit results from experiment and another data set of Bell circuit results executed on `toronto`. The self-simulation examples indicate a potential best-case simulation of `toronto` simulating itself by generating additional data sets.

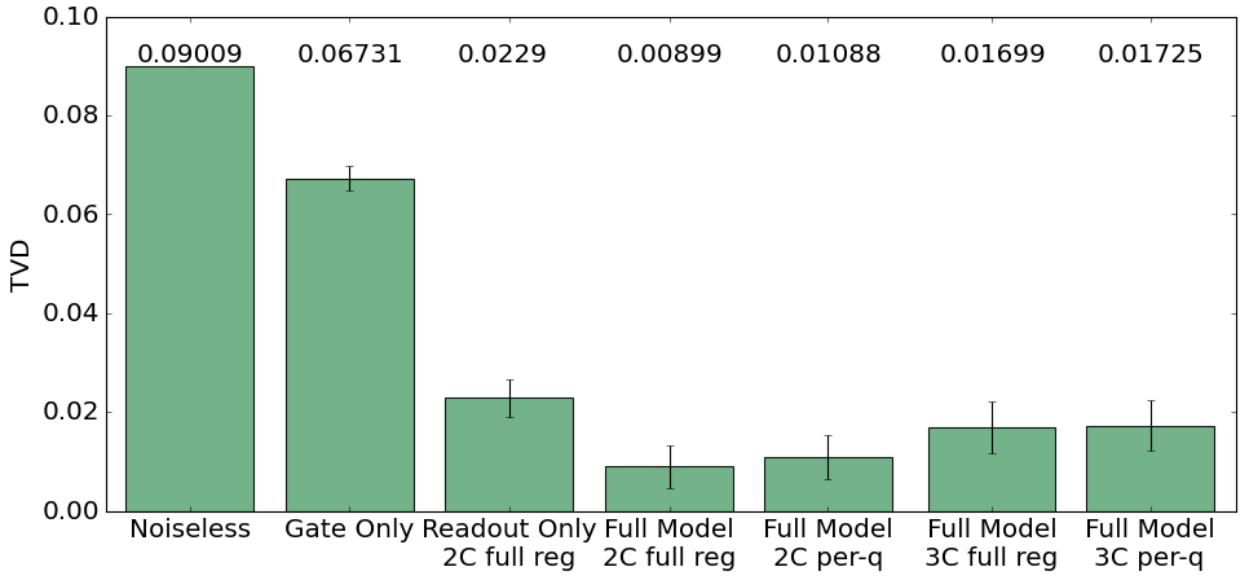

Figure 6.14: TVD to experiment of composite noise models constructed from error rates estimated using KNR. Error bars are calculated as the standard deviation across 100 trials of the Bell-state preparation circuit distributions.

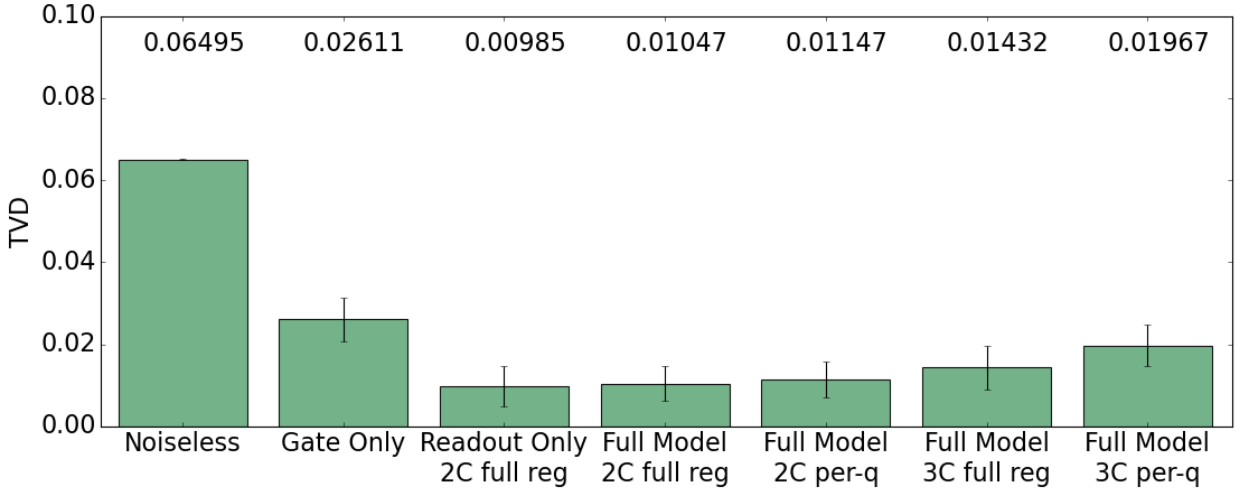

Figure 6.15: TVD to experiment of composite noise models constructed from error rates estimated using EDC. Error bars are calculated as the standard deviation across 100 trials of the Bell-state preparation circuit distributions.

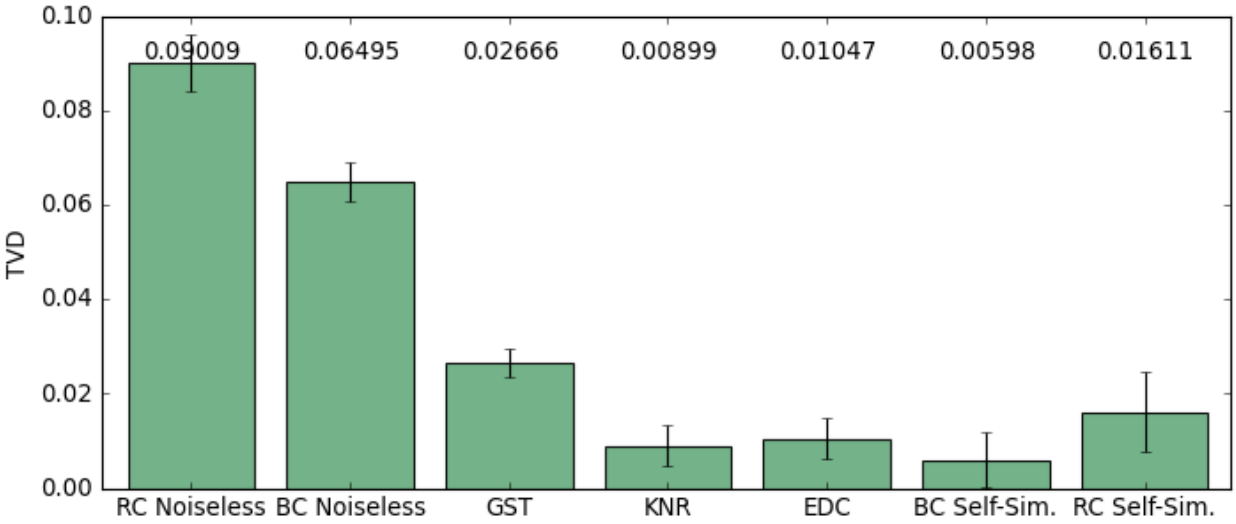

Figure 6.16: TVD between experiment and noisy simulation of the Bell state on qubits 0 and 1 using noise models constructed from GST, KNR, and EDC protocols. Error bars represent the standard deviation across 100 trials of Bell state simulation distributions. The error for the noiseless and self-simulation cases is calculated as the error propagation in TVD from the distributions.

The difference in performance of RC circuits and BC circuits is highlighted in the noiseless results. The lower TVD between the noiseless Bell results and the BC Bell results indicates that the BC Bell results are closer to ideal than the RC circuits. The KNR TVD is calculated to the RC Bell data, and the GST and EDC TVD is calculated to the BC Bell data. The noise model with the closest fit to experiment was KNR, although EDC is within error of KNR. In the RC self-simulation case, the TVD result likely indicates the effects of drift, since the additional data set used for comparison was taken several hours later. For BC Bell circuits, the additional data set is taken from the same job.

6.5 GHZ Benchmark Results

We evaluate the performance of noise models built using KNR and EDC methods in simulating GHZ-state preparation circuits. We calculate the TVD between our noisy circuit simulation outcomes and the circuit outcomes in experiment from `toronto` and show these results in Fig. 6.17. We compare these results to the TVD calculated between our selected RC and BC GHZ-state preparation circuit results and noiseless GHZ-state preparation results, for which we use an equal split between the all-zero and all-one states. We also compare to the TVD calculated between the selected RC and BC GHZ-state preparation circuit results and an additional data set of the same circuits run on `toronto` during the same time frame.

We find that the EDC noise model comes closest to accurately simulating the results of the GHZ circuits in experiment. The EDC noisy simulations are both closer to the self-simulated (best case) results and farther from the noiseless (worst case) results than the KNR noisy simulation results are to the respective RC GHZ results. At a size of 20 qubits, the EDC noise model simulation is about 0.4 lower TVD than noiseless, whereas the KNR noisy simulation is about 0.2 lower TVD than noiseless. For GHZ circuits of size 2 and 3 qubits, KNR simulation TVD is lower than the self-simulated TVD, but reaches a maximum distance away from self-simulated of 0.26 at 10 qubits, whereas EDC simulation TVD reaches a maximum distance from self-simulated of only 0.16 at 17 qubits.

These results indicate that the EDC noise model provides a closer description of the noise present in `toronto` than the KNR noise model.

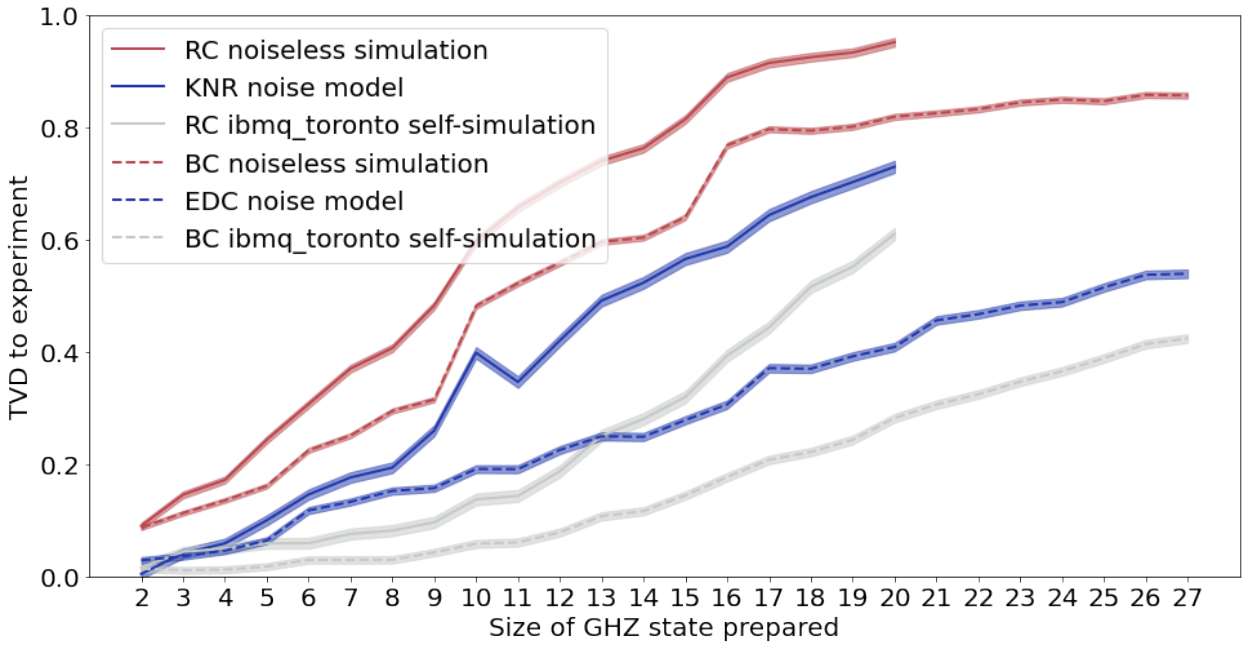

Figure 6.17: Total variation distance between experiment results and simulation results. Solid lines indicate TVDs calculated with randomly compiled results; dashed lines indicate TVDs with bare circuits. TVD for the noiseless case is calculated between experiment and results exactly split between an all-zero state and an all-one state. TVD for the "self-simulation" case is calculated between experiment and GHZ results executed on `toronto` during the same 12-hour period.

6.6 Bernstein-Vazirani Benchmark Results

In Fig. 6.18 we show the performance of our EDC and KNR noise models on the Bernstein-Vazirani algorithm. We compare these to the performance of the targeted experiment. In the case of EDC, this is the set of BV circuits which were executed in the same job as the characterization experiments and GHZ circuits. In the case of KNR, this is the set of BV circuits which were executed closest in time following the KNR characterization experiments. The EDC noise model used to model GHZ circuits is sufficient to model BV circuits because it consists of the same components. We use the KNR protocol to characterize cycles which define BV circuits and construct a noise model from these results as detailed in Section 5.3.2.

We compare the results of noisy BV circuit simulations to a "self-simulating" experiment, which for EDC is a set of BC BV circuits from a job about 15 minutes later and for KNR is a set of RC BV circuits executed a few hours later. The time differences are due to the amount of experiments which happened in between. In particular the short time between EDC circuit trials is a result of running the EDC circuit set multiple times to track error over time as in Figs. 6.1 and 6.2. BV secret strings should be returned by 100% of the results in the absence of noise, and therefore the noiseless case of BV circuit implementation returns an accuracy of 1.

The randomly compiled BV circuits performed better than the uncompiled BV circuits for every secret string encoding. The KNR noise model also performed better than the EDC noise model for every secret string, coming closer to the BV circuit results from `toronto` though both noise models were outside of error bars in every bitstring example. For encoded strings 101 and 110, the KNR noise model performed about as well as the self-simulated results.

Additionally, for both noise models the difference in accuracy between the noisy simulation and experiment increases with the number of CNOT gates. This is a correlation we observed in the data from `poughkeepsie` in Chapter 4. It might indicate that there are additional noise sources present in CNOT that are not accounted for in either the KNR model or the EDC model.

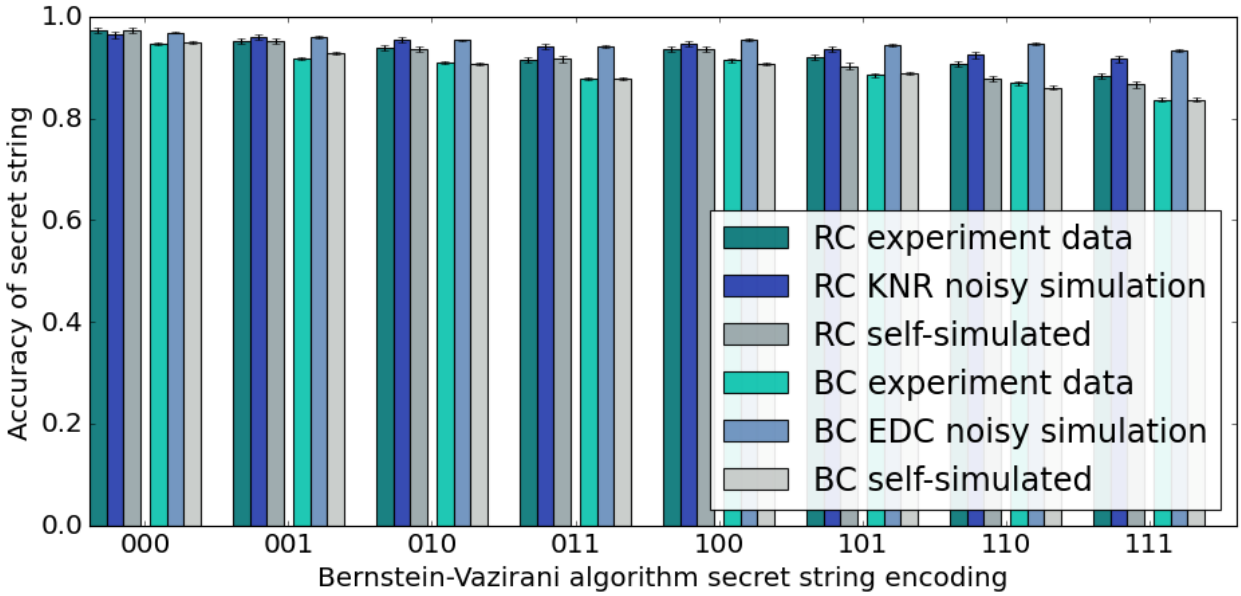

Figure 6.18: Bernstein-Vazirani results from experiment and noisy simulation.

6.7 Classical Resources Usage

We evaluate classical computation expense for these methods in creating experiments, analyzing the data, and performing classical simulations of quantum circuits. We report these in Table 6.4 measured by the amount of time taken to complete the computation. In general, the computational intensity of creating experiments is negligible. Although creating a GST experiment set can be computationally intensive, we are using a pre-built experiment set in pyGSTi. Analyzing the data of CB, KNR, and EDC requires a trivial amount of time to calculate on a basic laptop, but calculating the results of GST is computationally intensive as the algorithm for the GST analysis optimizes a model to best fit the experiment data [26].

Classical simulations of noisy quantum circuits are notoriously intensive, and we present a detailed report of their performance in Fig. 6.19. The classical computational expense of simulating quantum circuits grows exponentially in the size of the qubit count, and this trend is demonstrated in all of our noisy GHZ-state preparation circuit simulations.

Table 6.4: Estimates of classical resources used in our selected protocols. The "local" simulations were computed on a laptop with 16 GB RAM and Intel Core i7 processor. The "backend" simulations were sent as jobs to the IBM Q `ibmq_qasm_simulator`, a dedicated quantum circuit simulator backend which is available through the IBM Q suite [3]. While the GST classical calculation is computationally expensive, it may be parallelized on multiple processors to achieve speedup.

Method	**Details**	**Time Taken**
GST	Calculate results	70.13 hours
KNR	Simulate noisy RC GHZ circuits (2-20 qubits, local)	1.95 hours
EDC	Simulate noisy BC GHZ circuits (2-18 qubits, local)	33.53 minutes
EDC	Simulate noisy BC GHZ circuits (19-27 qubits, backend)	21.02 hours
KNR	Simulate noisy RC BV circuits (8 four-qubit circuits compiled into 32 circuits each)	1 minute
EDC	Simulate noisy BC BV circuits (8 four-qubit circuits)	1 minute

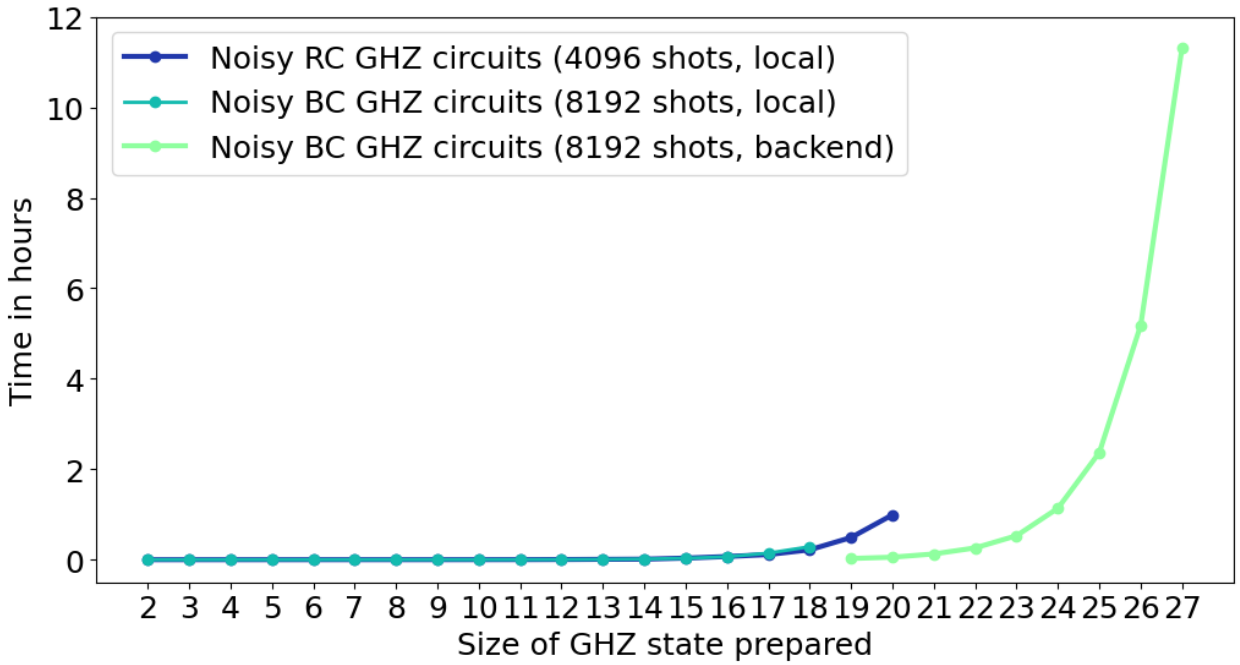

Figure 6.19: Time taken for noisy GHZ simulation as a function of qubit count. Noisy BC GHZ circuit simulations were switched from the local laptop to the IBM Q simulator backend for 19+ qubits due to the computational intensity of the largest GHZ circuit simulations. The IBM Q simulation backend has a limit of 32 qubits [3].

Chapter 7

Conclusion

In Chapter 1 we presented the background on quantum computers and the need to characterize and benchmark them. We introduced fundamental concepts in Chapter 2 and in Chapter 3 we presented a set of methods to characterize and benchmark quantum computers. In Chapter 4 we presented an approach to characterization focused on developing a coarse-grained model based on data from a small set of targeted subcircuit experiments. In Chapter 5 we present an experiment design to test a set of methods for characterizing and benchmarking quantum computers. In Chapter 6 we report our results from using these methods and discuss our analysis. Finally, we present our conclusions.

7.1 Characterization and Benchmarking Conclusions

We have implemented several different characterization methods in experiment on quantum computers. We demonstrate the effectiveness of each method in estimating device parameters and the accuracy of the resulting noise model in describing the QPU outcomes. We show that EDC, a highly efficient approach which yields a coarse-grained noise model, offers competitive accuracy with other state-of-the-art methods in tests on standard quantum algorithms.

Our results demonstrate that the best characterization method depends on the application. The structure, components and size of the circuit to be characterized all play a role in choosing characterization approaches. Furthermore, our results show that an increase in

experiment count and consequently information gain does not correlate with higher accuracy in noise descriptions used to simulate a QPU.

Our BV circuit results can be further applied as a benchmark of QPU performance. While BV algorithm circuits are unlikely to represent a broad indication of QPU performance, they share some hallmarks of benchmark tests with other commonly used benchmarks. For instance, BV circuits represent a practical calculation that a QPU might be used to solve. They are easily extensible and include several features of a typical quantum computation, such as single- and two-qubit gates, entanglement, superposition, and measurement. Furthermore, the expected outcome of BV is a single bitstring, which means that any results that are not the expected bitstring are the result of errors.

In this way, KNR may provide an estimate of performance of a QPU on BV algorithm benchmarks. KNR had close agreement to experiment in the reported accuracy of the encoded secret string. Using noise models constructed by KNR in simulation of BV benchmark experiments could provide valuable insight into expected performance of a QPU on a relevant test.

None of the models derived from the three methods were able to describe all noise present in the device. We tested all three on the Bell state and none of the results achieved zero TVD, although the TVD was low for KNR and EDC. In addition, the TVD results for GHZ circuits grew with the size of the circuit qubit register, indicating not only that there are additional soures of noise present in the system but that they might correlate with qubit count. In the GHZ example, some of this additional noise may come from decoherence of idling qubits as shown in Fig. 6.7. However, TVD sharply increases for larger circuits, approaching the maximum of one for the maximum register sizes (20-27 qubits), indicating that these experiments reach the limit of the capabilities of the QPU's ability to produce correct or predictable results.

The best characterization method varied by test. EDC performed better in GHZ tests, and KNR performed better on BV benchmarks. The structure of the quantum circuit might be a factor in this difference. To improve EDC performance on BV it may be beneficial to re-characterize the components in the BV circuit context and define a set of tests to characterize more components of the circuit. To improve KNR performance on GHZ, we

may be able to change the experiment setup in a way that is more optimized for the circuit and set of cycles, such as testing different combinations of CNOT parallelizations or sequence lengths.

If the performance of RC circuits and KNR noise models are strongly correlated and the performance of BC circuits and EDC noise models are similarly correlated, it may be possible to test what characterization method is most useful for a particular circuit. We could test the circuit of interest or a subcircuit of it for which we know the expected outcome both with and without randomized compiling. If the RC circuits perform better than the BC circuits, that might indicate that KNR is the best noise model choice. Likewise, if the BC circuits perform better than RC circuits, EDC may be the better noise model.

The main findings are generally corroborated among the protocols. For instance, the relative noise levels between asymmetric readout states and single- and two-qubit gates are similar among the protocols. However, the fidelity metrics and gate error levels are not always in agreement.

Non-Markovian noise is present in the system and particularly in the CNOT gates. GST indicates this, as does the performance of RC circuits to a lesser extent. RC circuits should perform best at tailoring noise into stochastic Pauli channels in the presence of arbitrarily non-Markovian noise [4]. Because the performance of KNR noise models steadily degrades for larger counts of CNOT gates, these stochastic channels evidently do not predict the QPU results and therefore the noise has not been tailored well.

7.2 GST

The expense of GST is not prohibitive for a two-qubit example. However, it is large enough that over the time period that GST experiments are run it is possible that parameter drift comes into play which affects the accuracy of the best-fit model. Drift is more likely to impact GST results than KNR, CB, or EDC primarily because GST requires the most experiments. To characterize a similar two-qubit example using KNR or EDC requires about 1/40th or less of the circuit count of GST.

However, the results of GST have more information to offer. In our experiments, GST confirms the presence of non-Markovian noise particularly prevalent in CNOT gates, which are also the noisiest gates in our tests. Non-Markovian noise impacts the accuracy of noise models built using GST, KNR, and EDC, and is likely to be a source of the additional error not accounted for in our best-fit models.

7.3 KNR and CB

In our GHZ circuit tests, the KNR noise model did not account for all the noise in the system. The KNR model achieved a poorer fit to RC GHZ experiment results than the EDC model fit to the BC GHZ experiment results as measured by TVD. Furthermore, the RC GHZ circuit results themselves have generally lower performance than uncompiled GHZ circuit results as measured by the rate of observation of the expected outcomes. As the circuit qubit register size increases, this performance worsens more quickly in the RC GHZ circuits than in the uncompiled GHZ circuits.

On the other hand, the KNR noise model performs better than the EDC noise model on the BV circuit benchmark as measured by the agreement between simulation and experiment in accuracy reported by the algorithm. This was true for every encoded secret string. Additionally, the RC BV circuit results had better performance than the uncompiled BV circuits by this same measure for every encoded secret string.

These two aspects of the KNR protocol results–performance of the KNR model fit and performance of the RC application circuit–are likely correlated. In the GHZ example, the performance of the KNR model and the RC circuits was poor yet in the BV example, the performance of the KNR model and the RC circuits was good. One likely reason for this is that while the gate set of GHZ and BV is the same, the structure of GHZ circuits is very different from the structure of BV circuits. GHZ circuits are a chain of CNOT gates with no cyclical structure, no parallelized gates, one easy gate and virtually all hard gates. In contrast, BV circuits have a cyclical structure, many parallelized gates, many more easy gates than hard gates, and only a few hard gates that are applied right before or after easy gates which allows twirling gates to be compiled together with the easy gates. The way that hard

gates are used in the circuit is likely a primary factor in the performance of RC and KNR noise models. The chain of unparallelized hard gates of GHZ circuits means that each hard gate becomes its own cycle and randomly compiling GHZ circuits results in a potentially large number of twirling gates inserted around each cycle. For the largest GHZ circuit examples we implemented, this can be up to an additional 300 single qubit easy gates inserted into the GHZ circuit. For the BV circuits, the additional gate count is no more than 8. This means that the potential for additional noise is much higher for the GHZ example than the BV example, leading to a commensurate degradation in performance of the application circuit. Likewise, this would have the effect of altering the noise channels measured by the KNR protocol such that the results of KNR may not be sufficiently descriptive of the application circuit to yield an accurate noise model.

7.4 EDC

The model of EDC is the simplest of all the methods and therefore provides the least detail of the underlying device or circuit characteristics. However, the EDC model provides a description of noise present on every tested component of the QPU and yields a noise model which performs best in simulating GHZ circuits as measured by TVD to experiment. It also requires the fewest experiments and scales only linearly in the characterized components, making it the most efficient approach to characterization.

While the EDC model did not perform as well as the KNR model in simulating BV circuit results, it does not provide a noise model for the single-qubit gates present in the BV circuits. Developing a noise model for single-qubit gates using the EDC method may improve the accuracy of the EDC model in the BV example.

In the BC GHZ circuit results, there are sharp increases in TVD between experiment and a noiseless GHZ when unusually noisy qubits are included in the circuit. These results demonstrate that it is worthwhile to avoid low-quality qubits. These results from `toronto` would likely be improved using routing techniques to better handle highly noisy qubits [74].

7.5 Computational Resources

A central focus of our tests is scalability, namely how accuracy of characterization correlates with experiment count. Our results suggest that this is not a strong correlation. EDC has a low resource count but high accuracy in some of our tests. KNR had high accuracy in other tests but requires significantly more circuits than EDC. GST has the highest experiment count which yielded a suite of information about the 2-qubit system but did not perform well in the Bell state test.

Classical computation resources needed to calculate GST are considerably higher than those needed for other protocols. These are not prohibitive in our example and can be reduced over our reported classical performance using parallelization and enhanced classical hardware. However, it is noteworthy that the classical computation expense of GST is a consideration in the overall experiment design, whereas for EDC, KNR, and CB the classical portion of the methodology is negligible.

While the experiment count of KNR and CB does not scale with qubit count, the total number of experiments needed to characterize all necessary cycles of the circuit of interest might still be high. Efficiency in experiments can be tuned in selecting the sequence lengths, number of sequence lengths, shot count, and circuit decompositions into cycles.

Bibliography

[1] M. L. Dahlhauser and T. S. Humble, "Modeling noisy quantum circuits using experimental characterization," *Phys. Rev. A*, vol. 103, p. 042603, Apr 2021. iv, 29

[2] "Creative Commons Attribution 4.0 International." https://creativecommons.org/licenses/by/4.0/, 2021. v, xiii, 27, 29

[3] "IBM Quantum." https://quantum-computing.ibm.com/, 2021. xi, xvi, 63, 65, 102, 103

[4] J. J. Wallman and J. Emerson, "Noise tailoring for scalable quantum computation via randomized compiling," *Phys. Rev. A*, vol. 94, p. 052325, Nov 2016. xiii, 22, 23, 24, 106

[5] S. J. Beale, A. Carignan-Dugas, D. Dahlen, J. Emerson, I. Hincks, A. Jain, D. Hufnagel, E. Ospadov, J. Saunders, J. J. Wallman, and A. Winick, "True-q," June 2020. xiii, 24, 67, 76, 83

[6] A. Erhard, J. J. Wallman, L. Postler, M. Meth, R. Stricker, E. A. Martinez, P. Schindler, T. Monz, J. Emerson, and R. Blatt, "Characterizing large-scale quantum computers via cycle benchmarking," *Nature Communications*, vol. 10, Nov 2019. xiii, 15, 23, 25, 26, 27

[7] Wolfram Alpha, "Wolfram Alpha." www.wolframalpha.com, March 2021. xv, 79, 80, 82

[8] E. Nielsen, L. Saldyt, Rob, J. Gross, T. Proctor, K. Rudinger, T. L. Scholten, msarovar, K. Young, D. Nadlinger, pyIonControl, and R. Blume-Kohout, "pygstio/pygsti: Version 0.9.9.2," June 2020. xv, 69, 70, 81, 84

[9] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, p. 1484–1509, Oct 1997. 1

[10] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, "Quantum computational chemistry," *Reviews of Modern Physics*, vol. 92, Mar 2020. 2

[11] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, pp. 467–488, Jun 1982. 2

[12] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, pp. 195–202, Sep 2017. 2

[13] S. Jordan, "Quantum algorithm zoo." quantumalgorithmzoo.org, 2020. Accessed: 2020-04-28. 2

[14] N. Jones, "How to stop data centres from gobbling up the world's electricity," 2018. 3

[15] Z. Li, N. S. Dattani, X. Chen, X. Liu, H. Wang, R. Tanburn, H. Chen, X. Peng, and J. Du, "High-fidelity adiabatic quantum computation using the intrinsic hamiltonian of a spin system: Application to the experimental factorization of 291311," 2017. 3

[16] Y. Nam, J.-S. Chen, N. C. Pisenti, K. Wright, C. Delaney, D. Maslov, K. R. Brown, S. Allen, J. M. Amini, J. Apisdorf, K. M. Beck, A. Blinov, V. Chaplin, M. Chmielewski, C. Collins, S. Debnath, K. M. Hudek, A. M. Ducore, M. Keesan, S. M. Kreikemeier, J. Mizrahi, P. Solomon, M. Williams, J. D. Wong-Campos, D. Moehring, C. Monroe, and J. Kim, "Ground-state energy estimation of the water molecule on a trapped-ion quantum computer," *npj Quantum Information*, vol. 6, p. 33, Apr 2020. 3

[17] Z. Li, X. Liu, N. Xu, and J. Du, "Experimental realization of a quantum support vector machine," *Phys. Rev. Lett.*, vol. 114, p. 140504, Apr 2015. 3

[18] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018. 3, 29

[19] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," *Nature*, vol. 549, pp. 242–246, Sep 2017. 3

[20] A. Nersisyan, S. Poletto, N. Alidoust, R. Manenti, R. Renzas, C. Bui, K. Vu, T. Whyland, Y. Mohan, E. A. Sete, S. Stanwyck, A. Bestwick, and M. Reagor, "Manufacturing low dissipation superconducting quantum processors," in *2019 IEEE International Electron Devices Meeting (IEDM)*, pp. 31.1.1–31.1.4, 2019. 3

[21] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, pp. 505–510, Oct 2019. 3, 4, 29

[22] N. Friis, O. Marty, C. Maier, C. Hempel, M. Holzäpfel, P. Jurcevic, M. B. Plenio, M. Huber, C. Roos, R. Blatt, and B. Lanyon, "Observation of entangled states of a fully controlled 20-qubit system," *Phys. Rev. X*, vol. 8, p. 021012, Apr 2018. 3

[23] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. 3, 8, 11, 14, 40

[24] R. J. Sadlier and T. S. Humble, "Near-optimal routing of noisy quantum states," in *Quantum Communications and Quantum Imaging XVII* (K. S. Deacon, ed.), vol. 11134, pp. 28 – 39, International Society for Optics and Photonics, SPIE, 2019. 4, 60

[25] G. A. L. White, C. D. Hill, F. A. Pollock, L. C. L. Hollenberg, and K. Modi, "Demonstration of non-markovian process characterisation and control on a quantum processor," *Nature Communications*, vol. 11, p. 6301, Dec 2020. 13

[26] E. Nielsen, K. Rudinger, T. Proctor, A. Russo, K. Young, and R. Blume-Kohout, "Probing quantum processor performance with pygsti," *Quantum Science and Technology*, vol. 5, p. 044002, Jul 2020. 15, 81, 101

[27] Z. Hradil, "Quantum-state estimation," *Phys. Rev. A*, vol. 55, pp. R1561–R1564, Mar 1997. 15

[28] G. C. Knee, E. Bolduc, J. Leach, and E. M. Gauger, "Quantum process tomography via completely positive and trace-preserving projection," *Phys. Rev. A*, vol. 98, p. 062336, Dec 2018. 15

[29] D. Greenbaum, "Introduction to quantum gate set tomography," 2015. 15, 20, 21, 69

[30] S. Kimmel, M. P. da Silva, C. A. Ryan, B. R. Johnson, and T. Ohki, "Robust extraction of tomographic information via randomized benchmarking," *Phys. Rev. X*, vol. 4, p. 011050, Mar 2014. 15

[31] S. Kimmel, G. H. Low, and T. J. Yoder, "Robust calibration of a universal single-qubit gate set via robust phase estimation," *Phys. Rev. A*, vol. 92, p. 062315, Dec 2015. 15

[32] S. G. Schirmer, A. Kolli, and D. K. L. Oi, "Experimental hamiltonian identification for controlled two-level systems," *Phys. Rev. A*, vol. 69, p. 050306, May 2004. 15

[33] S. T. Flammia and Y.-K. Liu, "Direct fidelity estimation from few pauli measurements," *Phys. Rev. Lett.*, vol. 106, p. 230501, Jun 2011. 15

[34] E. Magesan, J. M. Gambetta, and J. Emerson, "Scalable and robust randomized benchmarking of quantum processes," *Phys. Rev. Lett.*, vol. 106, p. 180504, May 2011. 15

[35] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, "Validating quantum computers using randomized model circuits," *Physical Review A*, vol. 100, Sep 2019. 15

[36] R. Blume-Kohout and K. C. Young, "A volumetric framework for quantum computer benchmarks," 2019. 15

[37] A. J. McCaskey, Z. P. Parks, J. Jakowski, S. V. Moore, T. D. Morris, T. S. Humble, and R. C. Pooser, "Quantum chemistry as a benchmark for near-term quantum computers," *npj Quantum Information*, vol. 5, p. 99, Nov 2019. 15

[38] S. T. Flammia and J. J. Wallman, "Efficient estimation of pauli channels," *ACM Transactions on Quantum Computing*, vol. 1, no. 1, pp. 1–32, 2020. 16, 28

[39] R. Blume-Kohout, J. K. Gamble, E. Nielsen, K. Rudinger, J. Mizrahi, K. Fortier, and P. Maunz, "Demonstration of qubit operations below a rigorous fault tolerance threshold with gate set tomography," *Nature Communications*, vol. 8, p. 14485, Feb 2017. 20, 69

[40] S. T. Merkel, J. M. Gambetta, J. A. Smolin, S. Poletto, A. D. Córcoles, B. R. Johnson, C. A. Ryan, and M. Steffen, "Self-consistent quantum process tomography," *Phys. Rev. A*, vol. 87, p. 062119, Jun 2013. 20

[41] R. Blume-Kohout, J. K. Gamble, E. Nielsen, J. Mizrahi, J. D. Sterk, and P. Maunz, "Robust, self-consistent, closed-form tomography of quantum logic gates on a trapped ion qubit," 2013. 20, 30

[42] Z. Cai and S. C. Benjamin, "Constructing smaller pauli twirling sets for arbitrary error channels," *Scientific Reports*, vol. 9, no. 1, p. 11281, 2019. 23

[43] R. Harper, S. T. Flammia, and J. J. Wallman, "Efficient learning of quantum noise," *Nature Physics*, Aug 2020. 28, 30

[44] K. M. Svore and M. Troyer, "The quantum future of computation," *Computer*, vol. 49, no. 9, pp. 21–30, 2016. 29

[45] K. A. Britt and T. S. Humble, "Instruction set architectures for quantum processing units," in *High Performance Computing* (J. M. Kunkel, R. Yokota, M. Taufer, and J. Shalf, eds.), (Cham), pp. 98–105, Springer International Publishing, 2017. 29, 31

[46] A. J. McCaskey, E. F. Dumitrescu, D. Liakh, M. Chen, W.-c. Feng, and T. S. Humble, "A language and hardware independent approach to quantum–classical computing," *SoftwareX*, vol. 7, pp. 245–254, 2018. 29

[47] L. Riesebos, X. Fu, A. A. Moueddenne, L. Lao, S. Varsamopoulos, I. Ashraf, H. van Someren, N. Khammassi, C. G. Almudever, and K. Bertels, "Quantum accelerated

computer architectures," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, IEEE, 2019. 29

[48] J. M. Gambetta, J. M. Chow, and M. Steffen, "Building logical qubits in a superconducting quantum computing system," *npj Quantum Information*, vol. 3, no. 1, p. 2, 2017. 29, 45

[49] M. Reagor, C. B. Osborn, N. Tezak, A. Staley, G. Prawiroatmodjo, M. Scheer, N. Alidoust, E. A. Sete, N. Didier, M. P. da Silva, E. Acala, J. Angeles, A. Bestwick, M. Block, B. Bloom, A. Bradley, C. Bui, S. Caldwell, L. Capelluto, R. Chilcott, J. Cordova, G. Crossman, M. Curtis, S. Deshpande, T. El Bouayadi, D. Girshovich, S. Hong, A. Hudson, P. Karalekas, K. Kuang, M. Lenihan, R. Manenti, T. Manning, J. Marshall, Y. Mohan, W. O'Brien, J. Otterbach, A. Papageorge, J.-P. Paquette, M. Pelstring, A. Polloreno, V. Rawat, C. A. Ryan, R. Renzas, N. Rubin, D. Russel, M. Rust, D. Scarabelli, M. Selvanayagam, R. Sinclair, R. Smith, M. Suska, T.-W. To, M. Vahidpour, N. Vodrahalli, T. Whyland, K. Yadav, W. Zeng, and C. T. Rigetti, "Demonstration of universal parametric entangling gates on a multi-qubit lattice," *Science Advances*, vol. 4, no. 2, 2018. 29

[50] M. Gong, M.-C. Chen, Y. Zheng, S. Wang, C. Zha, H. Deng, Z. Yan, H. Rong, Y. Wu, S. Li, F. Chen, Y. Zhao, F. Liang, J. Lin, Y. Xu, C. Guo, L. Sun, A. D. Castellano, H. Wang, C. Peng, C.-Y. Lu, X. Zhu, and J.-W. Pan, "Genuine 12-qubit entanglement on a superconducting quantum processor," *Phys. Rev. Lett.*, vol. 122, p. 110501, Mar 2019. 29

[51] C. Figgatt, A. Ostrander, N. M. Linke, K. A. Landsman, D. Zhu, D. Maslov, and C. Monroe, "Parallel entangling operations on a universal ion-trap quantum computer," *Nature*, vol. 572, no. 7769, pp. 368–372, 2019. 29

[52] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, "Realization of a scalable shor algorithm," *Science*, vol. 351, no. 6277, pp. 1068–1070, 2016. 29

[53] M. A. Sepiol, A. C. Hughes, J. E. Tarlton, D. P. Nadlinger, T. G. Ballance, C. J. Ballance, T. P. Harty, A. M. Steane, J. F. Goodwin, and D. M. Lucas, "Probing qubit memory errors at the part-per-million level," *Phys. Rev. Lett.*, vol. 123, p. 110503, Sep 2019. 29

[54] D. Kielpinski, C. Monroe, and D. J. Wineland, "Architecture for a large-scale ion-trap quantum computer," *Nature*, vol. 417, no. 6890, pp. 709–711, 2002. 29

[55] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. Landsman, K. Wright, and C. Monroe, "Experimental comparison of two quantum computing architectures," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3305–3310, 2017. 29

[56] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. Chow, and J. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," *Nature*, vol. 549, no. 7671, p. 242, 2017. 29

[57] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski, "Cloud quantum computing of an atomic nucleus," *Physical Review Letters*, vol. 120, no. 21, p. 210501, 2018. 29

[58] N. Klco, E. Dumitrescu, A. McCaskey, T. Morris, R. Pooser, M. Sanz, E. Solano, P. Lougovski, and M. Savage, "Quantum-classical computation of schwinger model dynamics using quantum computers," *Physical Review A*, vol. 98, no. 3, p. 032331, 2018. 29

[59] A. J. McCaskey, Z. P. Parks, J. Jakowski, S. V. Moore, T. D. Morris, T. S. Humble, and R. C. Pooser, "Quantum chemistry as a benchmark for near-term quantum computers," *npj Quantum Information*, vol. 5, no. 1, pp. 1–8, 2019. 29

[60] A. Veitia, M. da Silva, R. Blume-Kohout, and S. van Enk, "Macroscopic instructions vs microscopic operations in quantum circuits," *Physics Letters A*, p. 126131, 2019. 30

[61] K. Rudinger, T. Proctor, D. Langharst, M. Sarovar, K. Young, and R. Blume-Kohout, "Probing context-dependent errors in quantum processors," *Physical Review X*, vol. 9, Jun 2019. 30

[62] D. Leibfried, D. M. Meekhof, B. E. King, C. Monroe, W. M. Itano, and D. J. Wineland, "Experimental determination of the motional quantum state of a trapped atom," *Phys. Rev. Lett.*, vol. 77, pp. 4281–4285, Nov 1996. 30

[63] J. Poyatos, J. I. Cirac, and P. Zoller, "Complete characterization of a quantum process: the two-bit quantum gate," *Physical Review Letters*, vol. 78, no. 2, p. 390, 1997. 30

[64] I. L. Chuang and M. A. Nielsen, "Prescription for experimental determination of the dynamics of a quantum black box," *Journal of Modern Optics*, vol. 44, p. 2455–2467, Nov 1997. 30

[65] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland, "Randomized benchmarking of quantum gates," *Physical Review A*, vol. 77, p. 012307, Jan 2008. 30

[66] E. Magesan, J. M. Gambetta, and J. Emerson, "Scalable and robust randomized benchmarking of quantum processes," *Phys. Rev. Lett.*, vol. 106, p. 180504, May 2011. 30

[67] J. Helsen, J. J. Wallman, S. T. Flammia, and S. Wehner, "Multiqubit randomized benchmarking using few samples," *Phys. Rev. A*, vol. 100, p. 032304, Sep 2019. 30

[68] J. Eisert, D. Hangleiter, N. Walk, I. Roth, D. Markham, R. Parekh, U. Chabaud, and E. Kashefi, "Quantum certification and benchmarking," *Nature Reviews Physics*, vol. 2, p. 382–390, Jun 2020. 30

[69] M. J. Dousti and M. Pedram, "Minimizing the latency of quantum circuits during mapping to the ion-trap circuit fabric," *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Mar 2012. 30

[70] F. T. Chong, D. Franklin, and M. Martonosi, "Programming languages and compiler design for realistic quantum hardware," *Nature*, vol. 549, no. 7671, p. 180, 2017. 30

[71] Y. Shi, N. Leung, P. Gokhale, Z. Rossi, D. I. Schuster, H. Hoffmann, and F. T. Chong, "Optimized compilation of aggregated instructions for realistic quantum computers," *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS '19*, 2019. 30

[72] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, "Validating quantum computers using randomized model circuits," *Phys. Rev. A*, vol. 100, p. 032328, Sep 2019. 30

[73] F. B. Maciejewski, Z. Zimborás, and M. Oszmaniec, "Mitigation of readout noise in near-term quantum devices by classical post-processing based on detector tomography," *Quantum*, vol. 4, p. 257, Apr 2020. 30, 31, 35

[74] S. S. Tannu and M. K. Qureshi, "Not all qubits are created equal," *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS '19*, 2019. 30, 108

[75] P. Murali, J. M. Baker, A. Javadi-Abhari, F. T. Chong, and M. Martonosi, "Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '19, (New York, NY, USA), pp. 1015–1029, ACM, 2019. 30, 60

[76] A. Bassi and D.-A. Deckert, "Noise gates for decoherent quantum circuits," *Physical Review A*, vol. 77, Mar 2008. 34

[77] M. N. Lilly and T. S. Humble, "Quantum Noise Modeling," 2019. https://code.ornl.gov/qci/characterization. 37

[78] J. M. Chow, A. D. Córcoles, J. M. Gambetta, C. Rigetti, B. R. Johnson, J. A. Smolin, J. R. Rozen, G. A. Keefe, M. B. Rothwell, M. B. Ketchen, and et al., "Simple all-microwave entangling gate for fixed-frequency superconducting qubits," *Physical Review Letters*, vol. 107, Aug 2011. 37, 45

[79] S. Sheldon, E. Magesan, J. M. Chow, and J. M. Gambetta, "Procedure for systematically tuning up cross-talk in the cross-resonance gate," *Phys. Rev. A*, vol. 93, p. 060302, Jun 2016. 37

[80] D. Cruz, R. Fournier, F. Gremion, A. Jeannerot, K. Komagata, T. Tosic, J. Thiesbrummel, C. L. Chan, N. Macris, M.-A. Dupertuis, and C. Javerzac-Galy, "Efficient quantum algorithms for ghz and w states, and implementation on the ibm quantum computer," *Advanced Quantum Technologies*, vol. 2, no. 5-6, p. 1900015, 2019. 37

[81] K. X. Wei, I. Lauer, S. Srinivasan, N. Sundaresan, D. T. McClure, D. Toyli, D. C. McKay, J. M. Gambetta, and S. Sheldon, "Verifying multipartite entangled ghz states via multiple quantum coherences," 2019. 37

[82] E. Magesan, J. M. Gambetta, A. D. Córcoles, and J. Chow, "Machine learning for discriminating quantum measurement trajectories and improving readout," *Phys. Rev. Lett.*, vol. 114, p. 200501, May 2015. 40

[83] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, *et al.*, "Scipy 1.0–fundamental algorithms for scientific computing in python," *arXiv preprint arXiv:1907.10121*, 2019. 41, 44

[84] J. Koch, T. M. Yu, J. M. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, "Charge-insensitive qubit design derived from the cooper pair box," *Phys. Rev. A*, vol. 76, p. 042319, Oct 2007. 45

[85] A. D. Córcoles, E. Magesan, S. J. Srinivasan, A. W. Cross, M. Steffen, J. M. Gambetta, and J. M. Chow, "Demonstration of a quantum error detection code using a square

lattice of four superconducting qubits," *Nature communications*, vol. 6, p. 6979, 2015. 45

[86] A. W. Cross, L. S. Bishop, J. A. Smolin, and J. M. Gambetta, "Open quantum assembly language," 2017. 45

[87] A. Asfaw, L. Bello, Y. Ben-Haim, S. Bravyi, N. Bronn, L. Capelluto, A. C. Vazquez, J. Ceroni, R. Chen, A. Frisch, J. Gambetta, S. Garion, L. Gil, S. D. L. P. Gonzalez, F. Harkins, T. Imamichi, D. McKay, A. Mezzacapo, Z. Minev, R. Movassagh, G. Nannicni, P. Nation, A. Phan, M. Pistoia, A. Rattew, J. Schaefer, J. Shabani, J. Smolin, K. Temme, M. Tod, S. Wood, and J. Wootton, "Learn quantum computation using qiskit," 2020. 45

[88] M. Takita, A. W. Cross, A. D. Córcoles, J. M. Chow, and J. M. Gambetta, "Experimental demonstration of fault-tolerant state preparation with superconducting qubits," *Phys. Rev. Lett.*, vol. 119, p. 180501, Oct 2017. 45

[89] C. Rigetti and M. Devoret, "Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies," *Phys. Rev. B*, vol. 81, p. 134507, Apr 2010. 45

[90] IBM, "Qiskit Terra Documentation." https://qiskit.org/documentation/, (19 December 2018). 45, 71

[91] E. Bernstein and U. Vazirani, "Quantum complexity theory," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1411–1473, 1997. 56

[92] S. S. Tannu and M. K. Qureshi, "A case for variability-aware policies for nisq-era quantum computers," *arXiv preprint arXiv:1805.10224*, 2018. 60

[93] E. Wilson, S. Singh, and F. Mueller, "Just-in-time quantum circuit transpilation reduces noise," 2020. 65

[94] H. Abraham, AduOffei, R. Agarwal, I. Y. Akhalwaya, G. Aleksandrowicz, T. Alexander, E. Arbel, A. Asfaw, C. Azaustre, AzizNgoueya, A. Bansal, P. Barkoutsos, G. Barron,

L. Bello, Y. Ben-Haim, D. Bevenius, L. S. Bishop, S. Bolos, S. Bosch, S. Bravyi, D. Bucher, A. Burov, F. Cabrera, P. Calpin, L. Capelluto, J. Carballo, G. Carrascal, A. Chen, C.-F. Chen, E. Chen, J. C. Chen, R. Chen, J. M. Chow, S. Churchill, C. Claus, C. Clauss, R. Cocking, A. J. Cross, A. W. Cross, S. Cross, J. Cruz-Benito, C. Culver, A. D. Córcoles-Gonzales, S. Dague, T. E. Dandachi, M. Daniels, M. Dartiailh, DavideFrr, A. R. Davila, A. Dekusar, D. Ding, J. Doi, E. Drechsler, Drew, E. Dumitrescu, K. Dumon, I. Duran, K. EL-Safty, E. Eastman, P. Eendebak, D. Egger, M. Everitt, P. M. Fernández, A. H. Ferrera, R. Fouilland, FranckChevallier, A. Frisch, A. Fuhrer, M. GEORGE, J. Gacon, B. G. Gago, C. Gambella, J. M. Gambetta, A. Gammanpila, L. Garcia, S. Garion, A. Gilliam, A. Giridharan, J. Gomez-Mosquera, S. de la Puente González, J. Gorzinski, I. Gould, D. Greenberg, D. Grinko, W. Guan, J. A. Gunnels, M. Haglund, I. Haide, I. Hamamura, O. C. Hamido, V. Havlicek, J. Hellmers, Ł. Herok, S. Hillmich, H. Horii, C. Howington, S. Hu, W. Hu, R. Huisman, H. Imai, T. Imamichi, K. Ishizaki, R. Iten, T. Itoko, JamesSeaward, A. Javadi, A. Javadi-Abhari, Jessica, M. Jivrajani, K. Johns, Jonathan-Shoemaker, T. Kachmann, N. Kanazawa, Kang-Bae, A. Karazeev, P. Kassebaum, S. King, Knabberjoe, Y. Kobayashi, A. Kovyrshin, R. Krishnakumar, V. Krishnan, K. Krsulich, G. Kus, R. LaRose, E. Lacal, R. Lambert, J. Lapeyre, J. Latone, S. Lawrence, C. Lee, G. Li, D. Liu, P. Liu, Y. Maeng, A. Malyshev, J. Manela, J. Marecek, M. Marques, D. Maslov, D. Mathews, A. Matsuo, D. T. McClure, C. McGarry, D. McKay, D. McPherson, S. Meesala, T. Metcalfe, M. Mevissen, A. Mezzacapo, R. Midha, Z. Minev, A. Mitchell, N. Moll, M. D. Mooring, R. Morales, N. Moran, MrF, P. Murali, J. Müggenburg, D. Nadlinger, K. Nakanishi, G. Nannicini, P. Nation, E. Navarro, Y. Naveh, S. W. Neagle, P. Neuweiler, P. Niroula, H. Norlen, L. J. O'Riordan, O. Ogunbayo, P. Ollitrault, S. Oud, D. Padilha, H. Paik, Y. Pang, S. Perriello, A. Phan, F. Piro, M. Pistoia, C. Piveteau, A. Pozas-iKerstjens, V. Prutyanov, D. Puzzuoli, J. Pérez, Quintiii, R. I. Rahman, A. Raja, N. Ramagiri, A. Rao, R. Raymond, R. M.-C. Redondo, M. Reuter, J. Rice, D. M. Rodríguez, RohithKarur, M. Rossmannek, M. Ryu, T. SAPV, SamFerracin, M. Sandberg, R. Sapra, H. Sargsyan, A. Sarkar, N. Sathaye, B. Schmitt, C. Schnabel, Z. Schoenfeld, T. L. Scholten, E. Schoute, J. Schwarm, I. F.

Sertage, K. Setia, N. Shammah, Y. Shi, A. Silva, A. Simonetto, N. Singstock, Y. Siraichi, I. Sitdikov, S. Sivarajah, M. B. Sletfjerding, J. A. Smolin, M. Soeken, I. O. Sokolov, SooluThomas, Starfish, D. Steenken, M. Stypulkoski, S. Sun, K. J. Sung, H. Takahashi, I. Tavernelli, C. Taylor, P. Taylour, S. Thomas, M. Tillet, M. Tod, M. Tomasik, E. de la Torre, K. Trabing, M. Treinish, TrishaPe, W. Turner, Y. Vaknin, C. R. Valcarce, F. Varchon, A. C. Vazquez, V. Villar, D. Vogt-Lee, C. Vuillot, J. Weaver, R. Wieczorek, J. A. Wildstrom, E. Winston, J. J. Woehr, S. Woerner, R. Woo, C. J. Wood, R. Wood, S. Wood, S. Wood, J. Wootton, D. Yeralin, D. Yonge-Mallo, R. Young, J. Yu, C. Zachow, L. Zdanski, H. Zhang, C. Zoufal, Zoufalc, a kapila, a matsuo, bcamorrison, brandhsn, chlorophyll zz, dekel.meirom, dekool, dime10, drholmie, dtrenev, ehchen, elfrocampeador, faisaldebouni, fanizzamarco, gadial, gruu, hhorii, hykavitha, jagunther, jliu45, kanejess, klinvill, kurarrr, lerongil, ma5x, merav aharoni, michelle4654, ordmoj, rmoyard, saswati qiskit, sethmerkel, strickroman, sumitpuri, tigerjack, toural, vvilpas, welien, willhbang, yang.luh, yotamvakninibm, and M. Čepulkovskis, "Qiskit: An open-source framework for quantum computing," 2019. 71

[95] M. L. Dahlhauser and T. S. Humble, "Quantum Computer Characterization and Benchmarking," 2021. https://code.ornl.gov/qci/qccb. 72

[96] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, "Efficient Z gates for quantum computing," *Physical Review A*, vol. 96, Aug 2017. 77

Appendices

A Sampling Effects in Experiment and Simulation

Because of the statistical nature of quantum computers, the effects of sampling on final results are an important factor in analyzing output from experiments as well as from simulations when performed in a finite sampling mode as ours are. In particular, for binomial distributions defined by a probability parameter p, e.g. an error rate, we verify that our sampling statistics are maintaining reasonably small error bars on the measurement of p.

In Fig. 1 we plot the standard deviation for a binomial distribution as a function of the value p. The standard deviation is given by Eq. 1 where N is the number of samples, e.g. shots per circuit, and p is the binomial probability parameter, e.g. error rate. We plot this function for $N = 8192$, the maximum shot count per quantum circuit sent to the IBM quantum and simulation backends as an example.

$$E_{st.dev.} = \sqrt{\frac{p(1-p)}{N}} \tag{1}$$

The peak standard deviation occurs when $p = 0.5$. We next plot the standard deviation as a function of sample number, again up to a maximum of $N = 8192$, for $p = 0.5$ in Fig. 2. Figure 3 plots this same relationship but limits the window to small sample sizes for $N < 200$.

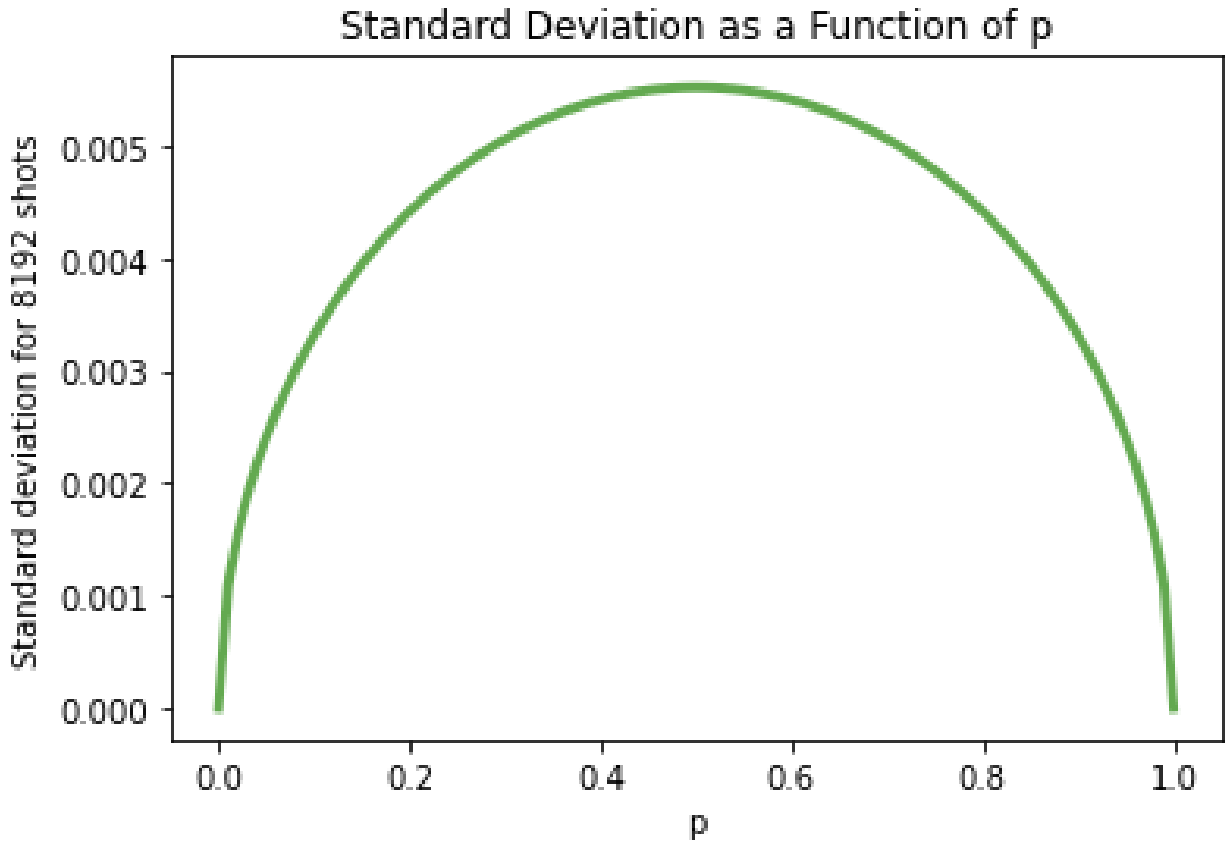

Figure 1: Standard deviation of a binomial distribution for $N = 8192$. The maximum standard deviation is when $p = 0.5$.

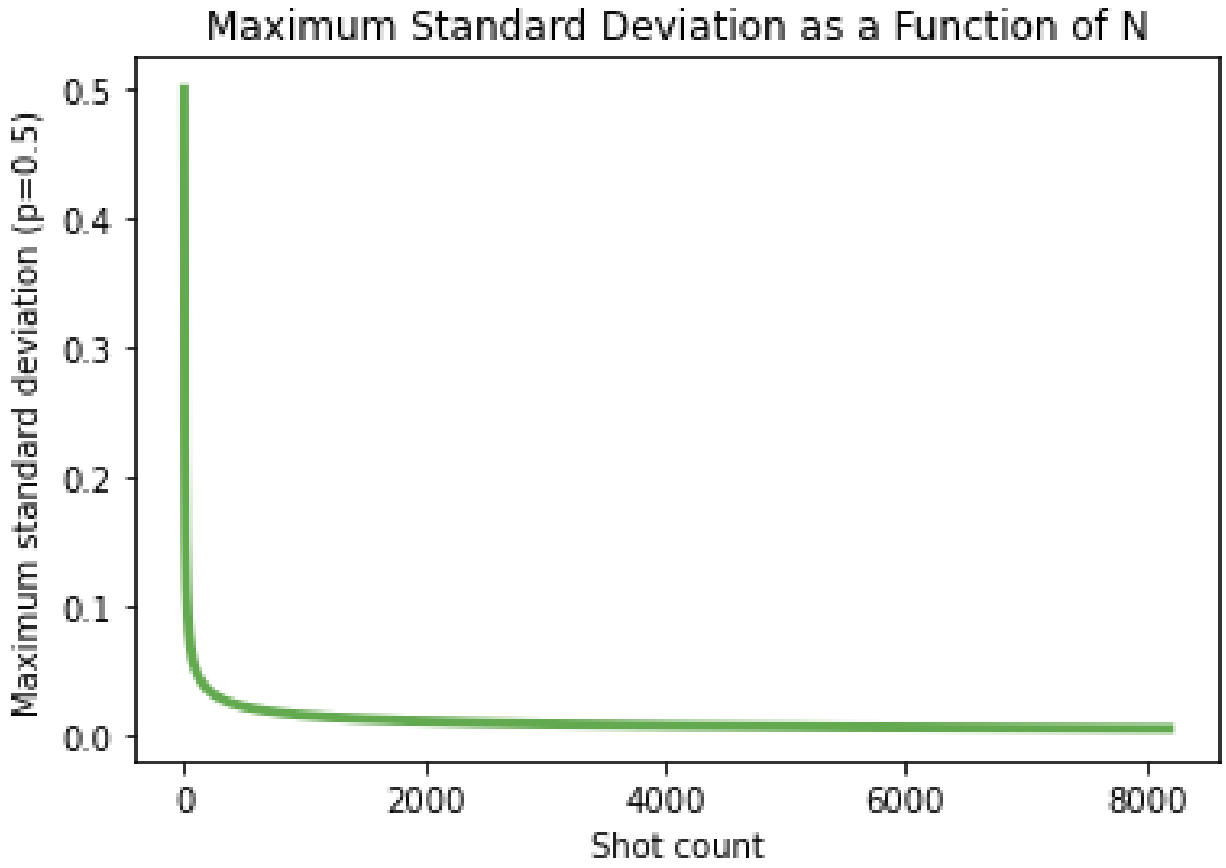

Figure 2: Standard deviation of a binomial distribution for $p = 0.5$.

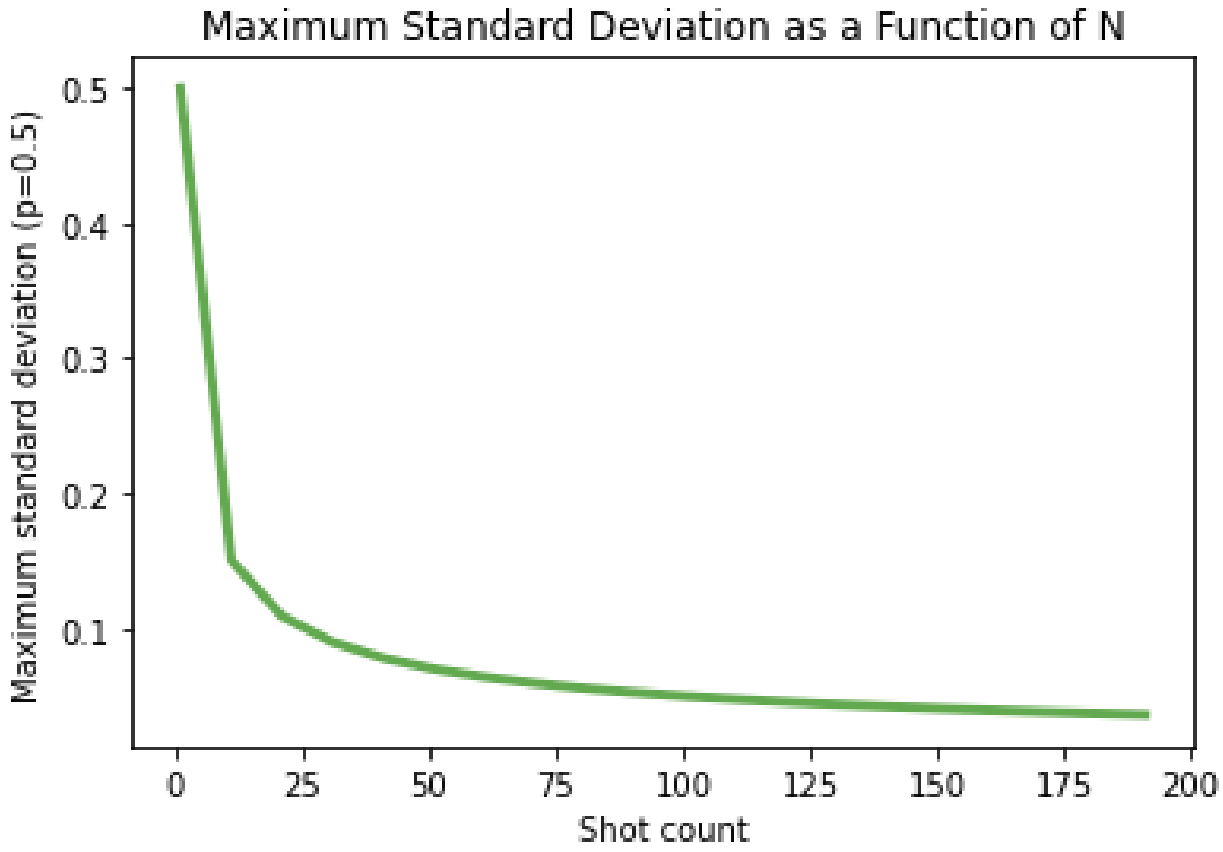

Figure 3: A closer view of Fig. 2 to show standard deviation at small sample sizes.

Vita

Megan Lilly Dahlhauser graduated from William Monroe High School and Blue Ridge Virtual Governor's School as salutatorian before attending Virginia Tech in Blacksburg, Virginia. She graduated with a Bachelor of Science degree in Physics, magna cum laude, with an Honors Scholar diploma from the Honors College and minors in Astronomy, Mathematics, and Music. After two terms as an intern at Oak Ridge National Laboratory, she chose to pursue doctoral studies in quantum computing in the Bredesen Center for Interdisciplinary Research and Graduate Education. As a graduate student with the Bredesen Center, she studied quantum computing through research conducted at Oak Ridge National Laboratory and coursework at the University of Tennessee, which included a concentration in science communication and outreach. She will continue working on quantum computing characterization and benchmarking at Sandia National Laboratories in the Computer Science Research Institute.