Benchmarking Qubit Quality and Critical Subroutines on IBM’s 20 Qubit Device

Daniel Koch\textsuperscript{1}, Brett Martin\textsuperscript{2}, Saahil Patel\textsuperscript{1}, Laura Wessing\textsuperscript{1}, Paul M. Alsing\textsuperscript{1}

\textsuperscript{1}Air Force Research Lab, Information Directorate, Rome, NY and
\textsuperscript{2}Air Force Academy, Colorado Springs, Co

As superconducting qubits continue to advance technologically, so too does the interest level in their viability for tackling quantum algorithms, ultimately reaching for the popularized notion of "Quantum Supremacy". In this study we characterize and identify several experiments which aim to benchmark various qubit qualities and subroutines which we deem critical to the success of larger, more complex quantum algorithms. These include experimentally measuring $T_1$ and $T_2$ coherence times, gate fidelities, sequential CNOT gate success rates, and finally CCNOT and QFT$^\dagger$ circuits implemented on several different qubit geometries. All of these experiments are performed on IBM’s 20-qubit ‘Poughkeepsie’ architecture, demonstrating the capabilities and limitations of state-of-the-art qubits.

\section{I. INTRODUCTION}

For as long as technology remains in the NISQ (Noisy Intermediate Scale Quantum) Era \cite{1} of quantum computers, quantum algorithm design will need to compensate for noisy qubits. While algorithms such as Shor’s \cite{2} and Grover’s \cite{3} have proven mathematical speedups over the best known classical algorithms, they critically rely on demands from quantum computers, such as qubit coherence times and gate fidelities, which are to date unfeasible. For superconducting qubits specifically, these various sources of noise \cite{4,17} inhibit the success of quantum algorithms, which in turn diminish or completely negate their potential for speedups. Even still, technological strides and new techniques for minimizing noise continue to develop \cite{8,10}, with the hope that someday soon we will reach full error-correcting \cite{11,13} quantum computers.

Due to the complex technological nature of quantum computers, the current standard model by which interested users can work with these machines is through remote access with various vendors \cite{14,17}. Analogous to high-level classical programming languages, these vendors offer quantum programming languages which grant the ability for users to execute quantum circuits, without necessarily knowing the full extent of how superconducting qubits work. Consequently, this allows for an important separation of quantum software from hardware, opening up more opportunities for research efforts in the field of quantum algorithms to test new models on real qubits. In the spirit of this new dawn of quantum programming, the findings in this study reflect the capabilities and limitations of this current model for quantum computer access, aiming to test the 20-qubit ‘Poughkeepsie’ architecture through various benchmarking experiments.

Each experiment in this study is motivated by different measures which are critical to the success of larger, more complex quantum algorithms. These include $T_1$ and $T_2$ coherence times \cite{18,20}, single and 2-qubit gate fidelities, and qubit connectivity. After testing these properties individually, we then study their combined effects through implementations of CCNOT and QFT$^\dagger$ circuits \cite{21,22}. Throughout these various experiments, we make a concentrated effort to distinguish between results which are simply technological benchmarks (coherence times, gate fidelities, etc.) and those which are more fundamental to algorithm design. Specifically, results of the latter case include experimental findings whereby trends in the success of the quantum circuit are determined by factors which are independent (or minimally influenced) by the noise of the qubits. We hope that these findings provide insight for future NISQ Era algorithm design, adding to the growing population of studies which aim to benchmark IBM’s qubits \cite{23,25}, as well as test the limits of various algorithm implementations \cite{20,30}.

\subsection{A. Layout}

The layout of this paper is as follows: In section 2 we investigate the $T_1$ and $T_2$ coherence times of various qubits. In section 3 we demonstrate CNOT gate fidelities across all 20 of IBM’s Poughkeepsie qubits, showing the extent to which a single CNOT operation can reliably be performed between distant qubits. Section 4 contains no experimental results, but lays the framework and motivation for the remainder of the study. In sections 5 and 6 we experimentally implement CCNOT and QFT$^\dagger$ circuits on various qubit geometries, and across all possible qubit combinations on the 20-qubit chip. And lastly, section 7 summarizes the main results of the paper and their implications for future research efforts.

\section{II. COHERENCE TIMES}

In quantum computing, a qubit is a two-level system that can simultaneously occupy both the $|0\rangle$ (ground) and $|1\rangle$ (excited) states through superposition, and is also sensitive to the relative phase between the two states. In practice however, one must always be mindful of the potential for noise to cause qubits to deviate from their intended states. Contrary to their classical counterparts,
current qubits have short timescales for which their quantum states are usable in any sort of calculation. These time frames are referred to as coherence times, quantified by the metrics \( T_1 \) and \( T_2 \), and promisingly have shown Moore’s Law like increases as technologies continue to improve [cite?].

The metrics \( T_1 \) and \( T_2 \) are single qubit measures which continue to be standards for comparison with each new generation of quantum computers, representing typical timescales for which a qubit has lost its computational utility. Physically, these metrics correspond to a qubit’s interactions with a noisy environment, tracking the probability that a qubit’s excited state \( (T_1) \) or superposition state \( (T_2) \) is preserved as an exponential decay in time. Equation \( 1 \) below shows the probability of a qubit resisting a decoherence collapse after an interval of time \( \Delta t \).

\[
P_\text{r}(\Delta t) = e^{-\frac{\Delta t}{T_1}}
\]

While working on IBM’s Poughkeepsie architecture, coherence times for both \( T_1 \) and \( T_2 \) ranged from lows of 30 - 40\( \mu \)s, to highs of 100 - 120\( \mu \)s. In this section we present experimental results which aim to verify these coherence times by directly observing decoherence rates. As a disclaimer, the results shown in the coming plots represent the ”best” data obtained while working on Poughkeepsie. When working with these shared devices one must be mindful of other users, which can cause certain times during the day to be more competitive for machine usage than others, leading to increased times between circuit runs. This in turn can be problematic for experiments which require data collecting from numerous trials, such as the coherence experiments to come, as it has been demonstrated that qubit coherence times can fluctuate throughout a 24 hour day [?]. Thus, the results shown in the coming two subsections represent data which was collected over the shortest total time interval, giving rise to plots which demonstrate qubits maintaining consistent coherence times.

### A. \( T_1 \) Energy Relaxation

The \( T_1 \) metric corresponds to a spontaneous decay from the excited state \( |1\rangle \) to the ground state \( |0\rangle \). Just as classical computing is reliant on the long shelf life of bits, a critical ingredient for quantum circuits is how long a qubit can maintain the \( |1\rangle \) state. 2-qubit gates such as CNOT and control-\( R_\phi \), which make up the backbone of several critical quantum subroutines, are reliant on ‘control’ qubits whereby the action of the 2-qubit gate is only performed if the control qubit is in the \( |1\rangle \) state. The impact of a spontaneous energy relaxation happening in the middle of an algorithm can vary depending on when, and on which qubit the error occurs. If a key qubit in a circuit were to unintentionally undergo a \( T_1 \) event, it could spell the end of the algorithm. Conversely, as demonstrated in some of the later experiments, certain algorithms can still yield successful results despite one or more qubits undergoing spontaneous decays.

Following from the mathematical description of a decoherence collapse given earlier, figure\[ ] below shows the circuit used to verify the underlying \( T_1 \) nature of IBM’s qubits. The circuit is designed such that the qubit is initially brought into the \( |1\rangle \) excited state via an \( X \) gate, followed by a desired amount of time \( \Delta t \) whereby we anticipate an energy relaxation according the probability distribution given in equation \( 1 \).

\[
|0\rangle - X - |\Delta t\rangle
\]

FIG. 1: Quantum circuit for studying \( T_1 \) coherence times. The qubit is excited into the \( |1\rangle \) state via the \( X \) gate, followed by various amounts of time where the qubit may spontaneously undergo a \( T_1 \) collapse.

In performing the experiment in figure\[ ] various \( \Delta t \) times were studied in order to reveal the exponential decaying nature the qubits. Physically, these lengths of time were implemented through the use of Identity gates, which leave the quantum system unaltered after each use. For each value of \( \Delta t \) the circuit was run 8000 times, from which the results were then used to compute an average percentage probability of decay. Once completed, exponential regression fits were then performed to the data, shown below in figure\[2]. The \( T_1 \) values from these best fits are displayed alongside the plots, as well as the reported \( T_1 \) times by IBM for each qubit.

![FIG. 2: (scatter plots) Data collected running the circuit in figure\[1] for various qubits on IBM’s Poughkeepsie architecture. Accompanying each set of data are exponential regression best fits (dashed lines) used to extract experimental \( T_1 \) values (black circles). These \( T_1 \) values are shown in the accompanying table, along with the reported times from IBM.

### B. \( T_2 \) Transverse Relaxation

By comparison to \( T_1 \), which has a single well defined physical description, \( T_2 \) coherence for qubits can take on several potential definitions. In this study, we present results based on two experiments, commonly referred to
as ‘$T_2$ Ramsey’ and ‘$T_2$ Echo’. The quantum circuits for each experiment are shown below in figure 3. In both experiments the qubit is initially brought into a 50-50 superposition state via a Hadamard gate, followed by various amounts of time $\Delta t$, and finally a second Hadamard just before the measurement. During the time between Hadamard gates, the qubit is subject to both spontaneous energy relaxation ($T_1$) as well as dephasing, for a combined referred to as transverse relaxation [6].

**Ramsey:** $|0\rangle \xrightarrow{H} |+\rangle \xrightarrow{\Delta t} |+\rangle \xrightarrow{H} |0\rangle$

**Echo:** $|0\rangle \xrightarrow{H} |+\rangle \xrightarrow{\Delta t/2} |X\rangle \xrightarrow{\Delta t/2} |+\rangle \xrightarrow{H} |0\rangle$

**FIG. 3:** Quantum circuits for demonstrating the $T_2$ nature of IBM’s qubits. The difference between the two experiments can be seen in the extra $X$ gate between the split $\Delta t$.

Beginning with the Ramsey experiment, with a qubit initialized in the $|0\rangle$ state, the theoretical final state after two sequential Hadamard gates ($\Delta t = 0$) should return the qubit back to the ground state. However, when time is introduced in between these two $H$ gates, the qubit becomes susceptible to $T_2$ transverse relaxation. Illustrated in figure 3, the dephasing component of $T_2$ relaxation can be represented as a "drifting" effect around the equatorial plane on the Bloch Sphere, whereby one continually loses knowledge of the exact position of the state over time. Physically, this effect causes the second Hadamard gate to transform the qubit to a new final state based on the elapsed time, one which oscillates between $|0\rangle$ and $|1\rangle$ with the frequency of the drift $^5$.

**FIG. 4:** Bloch Sphere representation of the state of the qubit after being initialized by a Hadamard gate. The orange shaded areas in the equatorial plane represent the growing uncertainty in the state of the qubit, reaching a fully decoherent state after enough time.

$$|\Psi(t)\rangle = \frac{|0\rangle + e^{i\omega t}|1\rangle}{\sqrt{2}} \quad (2)$$

In order to obtain data similar to that of the $T_1$ experiment, to perform an exponential best fit, one can counteract the dephasing process of the qubit using the $T_2$ Echo circuit. By splitting each time evolution ($\Delta t$) into two equal halves and using an additional $X$ gate, one can refocus the state of the qubit back to the state $|+\rangle$ state just prior to the second Hadamard, guaranteeing a theoretical measurement of $|0\rangle$. Graphically, one can picture this process as the state of the qubit drifting along for some time $\Delta t$ (figure 4), becoming $|\Psi(\Delta t)\rangle$ (equation 2), undergoing a reflection as a result of the $X$ gate, and finally drifting once more of equal time back to its starting state.

Plotted below are results obtained running the Ramsey and Echo experiments on various qubits on the Poughkeepsie architecture. Figure 5 shows a typical Ramsey experiment, whereby the probability of the final state oscillates between $|0\rangle$ and $|1\rangle$ as a function of time, while simultaneously damping into a fully decohered state. Figure 6 illustrates the Echo technique described earlier, showing the effect of using an $X$ gate to let the quantum system naturally refocus the state of the qubit. Additionally, exponential best fits for the $T_2$ values are given, along with reported values from IBM.

**FIG. 5:** Data collected running the $T_2$ Ramsey circuit [3] for two different time scales, both on the same qubit.

**FIG. 6:** (green circles) Experimental results running the $T_2$ Echo circuit for two different qubits. (dashed line) Exponential best fit, used to extract experimental values for $T_2$ (black circle) to compare with those reported by IBM.

## III. CHAINING CNOT GATES

When comparing quantum algorithms to classical competitors, claims of speedups often assume full connectivity between all qubits in the quantum system. By connectivity, we refer to the ability for two qubits to perform a 2-qubit operation. If one looks to the foreseeable future of
qubit technologies however, it is possible that a fully connected superconducting 20 or 50+ qubit device may be upwards of a decade away. Thus, in order to compensate for lacking connectivity, quantum algorithms will need to be adapted to fit the various existing architectures.

In this section we investigate the effectiveness of implementing CNOT gates as a means of compensating for limited qubit connectivity. We study the reliability with which one can use a series of CNOT gates to invoke a control operation between distant qubits, which do not directly share a connection. Figure 7 shows an example of a length-3 chain (two intermediate qubits separating the control and target), achieving a CNOT operation between qubits A and B.

![FIG. 7: A CNOT gate implemented between qubits A and B. Qubits 1 and 2 serve as ancilla, acting as temporary control qubits in order to pass along the desired effect.](image)

In figure 7, qubits such as 1 and 2, which only serve an intermediate means for connecting A and B, are often referred to as ancilla qubits. Such qubits play a pivotal role in delivering the control operation between the distant computational qubits (A and B), and in principle are meant to have no direct impact on the success of the algorithm. In practice however, this last point can be difficult to control, as merely their incorporated presence in the quantum system can lead to new sources of error. Additionally, this problem can become compounding as certain algorithms may require the use of the same ancilla qubits several times, requiring proper handling of these qubits after each use.

A. Delivering Desired States

In testing the effectiveness of IBM’s 20-qubit device for implementing CNOT chains, the overall goal of each experiment is to measure both a starting and final (control and target) qubit in the |1⟩ state. This is done by exciting the initial control qubit into the |1⟩ state by an X gate, followed by a series of CNOT gates along some path of ancilla qubits. For each path studied we examine three circuits (figure 8), two such that the desired final state of the system contains all of the ancilla qubits in the |0⟩ state, and one for |1⟩. The motivation for studying multiple variations of each chain circuit stems from whether or not a certain algorithm requires the ancilla qubits to be reset for future use, often times determined by whether or not the control qubit contains superposition.

![FIG. 8: (left) CNOT chain circuit where the desired final state of the system leaves all of the ancilla qubits in the |1⟩ state. (center) Modified version of the left circuit, where the presence of additional X gates now results in the desired final state of each ancilla qubit to be |0⟩. (right) The most general form for a CNOT chain, using additional CNOT gates to account for the potential of superposition present on the starting qubit.](image)

The left circuit in figure 8 is the simplest form of a CNOT chain, where the ideal final state of the system leaves each ancilla qubit in the |1⟩ state. More generally, one can imagine that this circuit represents the case where the ancilla qubit states are inconsequential. That is to say, the only desired effect is such that the distant control and target qubits are both in the |1⟩ state, completing the effect of the computational CNOT gate between them regardless of the intermediate ancilla qubits. Conversely, circuits of the center and right type are designed such that the desired final state of the system leaves each ancilla qubit back in the |0⟩ state, representing the case where one anticipates future use from these ancilla. For instances where one knows that the initial control qubit is purely in the |1⟩ state, the central circuit would be optimal due to the use of only single qubit gates for resetting the ancilla. However, for more general cases in which the control qubit may contain superposition, the additional CNOT gates in reverse order are necessary.

B. Experimental Design

In designing CNOT chain paths for IBM’s 20-qubit Poughkeepsie architecture, figure 9 below illustrates the general layout for each experiment, showcasing the longest path of ancilla qubits tested for a single CNOT chain (touching all 20 qubits on the device). In addition to the maximum, all intermediate lengths were tested as well, keeping the control qubit fixed and moving the final target qubit along the path shown.

The path shown in figure 9 is one of four orientations tested experimentally. In order to achieve the best average result for CNOT chain success, as well as potentially identify any trends for certain qubits, three additional orientations of the 9 were also tested (figure 10). The success rates of the various experiments and circuit types, as well as implications for algorithm design, are discussed in the next section.
CNOT Chain Path on the Poughkeepsie Architecture. One experimental run studies the success rates for implementing a CNOT operation between the starting control qubit (C) and the final target qubit (T), tested for all chain lengths from 1 to 19.

Additional orientations for the full 20-qubit CNOT chain experiment. Individual results for each orientation can be seen in figure 11 as well as the combined average success rates across all four in figure 12.

C. Experimental Results

Presented here are the results experimentally gathered on the IBM 20-qubit Poughkeepsie architecture for the three CNOT experiments laid out previously. In all of the plots, we distinguish the measurement results from each experiment into two categories: 'Success %' or 'Perfect %', which we refer to as the 'success rates' for the remainder of this paper. The more lenient of the two metrics, Success %, refers to measured final states whereby the distant control and target qubits are both in the $|1\rangle$ state, regardless of the ancilla qubits. Conversely, Perfect % tracks the percentage of trials where all qubits in the system are found to be in their theoretically desired state of either $|0\rangle$ or $|1\rangle$.

When comparing the plots in figure 11 it is clear that there are some noticeable differences in success rates between the various paths, some as large as 10 - 20% after length-10 chains. Interestingly, a finer look at the data reveals distinct drops at certain chain lengths for the various paths. The most noticeable of these dips being at length-7 for diamonds, length-12 for circles, and length-14 for squares. Upon reviewing each path orientation, we found that all of these drops in success occur at the same qubit, the central qubit in the third row from the top in figure 9. In all of the plots, it is worth noting the apparent low success rates for length-1 chains. Due to the consistency of this trend, it is possible that this result is simply the readout error for Poughkeepsie, which can be thought of as a flat percentage of failure applied to each qubit in all experiments.

Focusing now on the trends between the three circuit types laid out in figure 8 plotted below in figure 12 are the success rates for each circuit, averaged across the four path orientations. Beginning with the Success % results, the data shows that the presence of the additional X gates for resetting ancilla qubits has little to no impact when compared to using no gates. This agrees with what one would expect theoretically, given that chronologically the chain of CNOTs in both circuits are the same, therefore uninfluenced by any later gate operations on each qubit. Conversely, because of the way in which the additional CNOT gates are staggered in the third circuit type, we see a decrease in Success % rates resulting from the fact that these extra gates ultimately delay the final measurement, effectively creating more time for decoherence errors.

Now turning to the results for each circuit’s Perfect % rates, the bottom plot in figure 12 reveals the impact of using additional gates on the ancilla qubits. Using the case of no gates as a baseline, the data shows that the usage of X gates improves the reliability of having predictable ancilla states, while the use of CNOT gates has the opposite effect. For the circuit containing no gates, the data shows a widening gap between the Perfect % and Success % rates as a function of chain length. This trend is likely attributable to $T_1$ energy decays on the ancilla, which become more problematic as the number of ancilla increases (more opportunities for collapse) as well as circuit length (more time for each qubit to decay). By comparison, the usage of additional X gates immediately after each CNOT remedies this second problem, effectively minimizing the time an ancilla may experience a $T_1$ relaxation regardless of chain length.

Finally, the results for the circuit type utilizing CNOT gates shows the worst Perfect % rates between the three different circuits. Conceptually, resetting the ancilla
qubits in this way is plagued with several issues, the worst of which being increased circuit length. By requiring an entire second CNOT chain for resetting, qubits closer to the control must maintain their excited state for almost double that of the other two circuits. Additionally, unlike with the usage of X gates, resetting the ancilla qubits in this way means that the success of each ancilla is conditioned on the one prior. Thus, a single intermediate $T_1$ relaxation is enough to disrupt the entire resetting process.

IV. QUBIT GEOMETRY & ALGORITHM DESIGN

Having now seen the extent to which a single CNOT operation can be reliably transmitted across ancilla qubits, the next question is how useful could such chains be for constructing larger circuits? With limited connectivity on future NISQ devices being the expected standard, near term quantum circuits will need to critically rely on ancilla qubits and various techniques for algorithm implementation. In this section we present several qubit geometries and circuit implementations for which we later present experimental results on the Poughkeepsie architecture (see sections V and VI).

A. 3 Qubit Geometry

Despite lacking any set of three directly interconnected qubits, the Poughkeepsie architecture possesses numerous combinations of three linearly connected qubits, as shown in figure 13. Using qubits in this way to implement 3-qubit algorithms has the advantage of avoiding the need for any additional ancilla qubits, but becomes problematic when 2-qubit gate operations are required between the outer two qubits, who lack a direction connection. Compensating for this lacking connection requires additional 2-qubit gates through the central qubit, acting essentially as a temporary ancilla.

FIG. 12: Success rates for the three circuit types laid out in figure [5] averaged across all four path orientations. The top plots demonstrate each circuit type’s ability to successfully deliver the CNOT operation, while the bottom plots show how reliably each circuit handles the ancilla qubits.

When implementing 2-qubit gate operations between the two unconnected qubits in figure 13, the challenge lies in protecting the central qubit from additional noise while simultaneously ensuring that its quantum state is unaltered in the end. Depending on the algorithm, there are circuit techniques which can be used to optimally perform such 2-qubit operations, but are not always generally implementable. In this study our goal is not to test any single optimal technique, but rather more standard approaches which in turn will serve as indicators into their effectiveness for implementation on a larger scale.

FIG. 13: Three linearly connected qubits.

FIG. 14: (top) Circuit implementation for a CNOT gate between qubits 1 and 3, which are both connected to qubit 2, but do not share a direct connection between themselves (see figure 13). (bottom) Circuit implementation for a control-$R_\phi$ gate between qubits 1 and 3 for the same geometry.

The circuits shown in figure 14 are those tested in the coming experiments. While not always optimal in circuit depth and gate count, each circuit is guaranteed to accomplish two things: 1) the successful implementation of the 2-qubit operation between the distant qubits, and 2) the central qubit is always returned back to its initial quantum state. This second point comes at the cost of the final SWAP gates, where notably this is where one typically looks to optimize when possible.

B. 4 Qubit Geometry

If one wishes to avoid using any computational qubits as temporary ancilla, such as with the 3-qubit geometry,
then one is forced to increase the size of the quantum system to supplement missing connections. Figure 15 below illustrates one such solution, using a single central ancilla qubit to supply all 2-qubit gate operations between the surrounding three computational qubits.

FIG. 15: Four qubit geometry used for implementing 3-qubit algorithms throughout this study. The central ancilla qubit allows for the implementation of 2-qubit gates between any two computational qubits, without needing to interfere with the state of the third qubit.

The 4-qubit geometry shown above represents an alternative to the problem of 3-qubit geometry discussed earlier, using a single ancilla in order to avoid any additional 2-qubit gates on the computational qubits. However, using a single ancilla for all 2-qubit gates requires that the state of the ancilla be properly reset after each usage in order to ensure the success of future gate operations. Figure 16 below shows two examples of how to implement a CNOT gate between computational qubits. Analogous to the different circuit types in the CNOT chain experiments, each way of resetting the ancilla qubit depends on the state of the control qubit.

FIG. 16: Example implementations of a CNOT gate between two computational qubits for the 4-qubit geometry (see 15). Depending on whether or not the control computational qubit possess superposition determines whether an X or additional CNOT gate is necessary for resetting the state of the ancilla.

In comparing the 3 and 4-qubit geometries, the question becomes whether or not the additional ancilla qubit aids or hinders algorithmic success. By requiring all 2-qubit gate operations traverse through a single qubit, a heavy burden is placed on the ancilla in terms of proper state manipulation and minimizing noise. Comparable to the central qubit in the 3-qubit geometry, one would expect that a majority of algorithmic success is tied to the ancilla qubit’s coherence properties and gate fidelities.

C. 6 Qubit Geometries

As the final qubit geometry size tested in this study, we present here two 6-qubit geometries which are motivated by the respective strengths and weaknesses anticipated of the 3 and 4-qubit geometries already mentioned. These geometries can be seen in figure 17 below, as well as their location on the Poughkeepsie architecture in 18.

FIG. 17: Six qubit geometries used for implementing 3-qubit algorithms throughout this study. (left) Qubit geometry referred to as ”3 Chain”, which uses three ancilla qubits to supplement the lacking connection between qubits Q1 and Q3. (right) Qubit geometry referred to as ”1 Chains”, whereby each computational qubit is separated by a unique ancilla.

FIG. 18: The two 6-qubit geometries experimentally tested on the Poughkeepsie architecture.

While not expected to be optimal in success rates, the two geometries put forth in figure 17 are designed to provide additional insight into the role of using ancilla in implementing algorithms over more sparse architectures. Additionally, each geometry possess a key ingredient found in the circuit designs for the 3 and 4-qubit geometries. Specifically, the left circuit in 17 (‘3 Chain’) contains the same connectivity between the three computational qubits as the 3-qubit geometry, sharing the advantage of having two-out-of-three direct connections. Conversely, the right geometry (‘1 Chains’) uses one ancilla for all 2-qubit gate operations just as with the 4-qubit geometry, but avoids the issue of repeated single ancilla use by using a different ancilla between each computational qubit.

V. CCNOT EXPERIMENTAL RESULTS

For the progression of quantum computing, the importance of the CCNOT (Control-Control X gate) operation, also known as a Toffoli gate, cannot be understated as so many quantum algorithms critically rely on its usage. Unlike previous operations studied up to this point however, the CCNOT gate is better thought of as a quantum circuit, consisting of several quantum gates in order to construct, as shown below in figure 19.

In order to implement the quantum circuit outlined in 19, one necessary condition is that the three qubits be interconnected, as required by the CNOT gates between not just the target and control qubits, but the two controls as well. As already mentioned however, such connectivity does not exist on IBM’s 20-qubit Poughkeepsie
FIG. 19: Quantum circuit for implementing a CCNOT gate operation. $Q_1$ and $Q_2$ serve as the control qubits, while $Q_3$ is the target, receiving the equivalent of an $X$ gate if both $Q_1$ and $Q_2$ are in the $|1\rangle$ state.

architecture, requiring the circuit be adapted to the fit various qubit geometries laid out in the previous section.

A. 3 Qubit Results

In testing the CCNOT circuit using three linearly connected qubits, there are two unique configurations for which one can implement the control and target qubits. Specifically, one can have the target qubit be either an outer (referred to as ‘CCT’) or central (‘CTC’) qubit. However, due to the design of the CCNOT circuit, which requires exactly two CNOT gates between all three qubits, both configurations result in the same circuit depth and gate count when using the techniques shown in 14 to supplement the missing outer connection. In total, the Poughkeepsie architecture possess 32 possible 3-qubit combinations, for which both orientations were tested, and the results are shown in figure 20.

FIG. 20: Success rates for the 32 possible 3-qubit combinations on the Poughkeepsie architecture, implementing the CCNOT circuit [19]. The higher success rate between the two orientations for control & target qubits are indicated color, blue for the case where the central qubit was the target (CTC) and red when it was an outer qubit (CCT).

Figure 20 shows the Success % rates found across the 96 tested CCNOT circuit implementations (three unique locations for the target qubit per each of the 32 total combinations). As illustrated by the varying colors and numerical values, it is clear that no single combination of qubits or orientation is dominant in producing the highest success rates. While certain combinations produced notably worse results, the data suggests that on average one can expect a successful CCNOT gate implementation on the order of 50-60%, with only a select few qubits reducing these values to around 25-40%.

B. 4 & 6 Qubit Results

When analyzing the results for the 4 and 6-qubit geometry implementations of the CCNOT circuit, the addition of ancilla qubits requires the tracking of the Perfect % metric as well. Beginning with the 4-qubit geometry, two cases for handling the CNOT gates between computational qubits were tested, corresponding to the two implementations shown in figure 16. For each implementation, experimental trials were performed on all six possible 4-qubit combinations available on the Poughkeepsie architecture. Additionally, for each combination of qubits, each of the three outer qubits were tested as the target. The average results for each qubit combination are shown below in figure 21.

FIG. 21: Success (green) and Perfect % (red) rates for the 4-qubit implementations of the CCNOT gate. The two plots correspond to using $X$ gates (left) for resetting the ancilla qubit, versus using additional CNOT gates (right).

When looking at the results in figure 21, it is clear that the use of $X$ gates for resetting the central ancilla qubit produce noticeably higher success rates, both for the Perfect and Success %’s. This is in agreement with the CNOT chain results from earlier (figure 12), once again highlighting the sensitivity of these real quantum devices when accounting for superposition states. In both experiments it is clear that certain qubit combinations boost higher fidelities than others, and upon further investigation, we were able to confirm that indeed the same problematic qubits resulted in the lowest success rates for both experiments.

If we now compare the results of the left plot in figure 21 (use of $X$ gates) to those of 20, we see that the success rates between the 3 and 4-qubit geometries are very close, with a slight edge going to the 3-qubit geometry. The closeness in the two results suggests that the use of an ancilla qubit, versus SWAP gates through a computational qubit, is a viable approach for algorithm design. However, this viability is lost when superpositions must be accounted for, which are are handled automatically by SWAP gates for the 3-qubit geometry but come with a...
heavy price when using CNOTs for the 4-qubit geometry. Proceeding now to the 6-qubit geometries, figure 22 below shows the full results for the two implementations illustrated in [17]. Once again the results are separated into the Success and Perfect % metrics, tracking the ability for the system to handle now three ancilla qubits. Unlike the 4-qubit geometry however, only the case using X gates for resetting ancilla were tested. In comparing the results from 22 with the 3 and 4-qubit geometries, the data shows that the "3 Chain" circuit implementation yields comparable success rates. Conversely however, the use of single ancilla qubits separating each computational qubit shows a dramatic decrease in CCNOT fidelities.

In order to understand the source of the pronounced difference between the two circuit designs' success rates, let us start by noting the consistencies between the two. First, both circuit implementations use the same total number of CNOT and X gates, producing near identical total gate counts and circuit depth. Additionally, in both cases each ancilla qubit is called upon exactly twice, followed immediately by an X gate for resetting back to the |0⟩ state. These consistencies suggest that the results shown in 22 are then attributable to the way in which the CNOT gates are distributed throughout the circuits. Specifically, it appears that there is a significant advantage to having direct connections for two out of the three required CNOTs. This in turn suggests that perhaps the preference for larger algorithm designs will be to group computational qubits as near together as permitted, supplementing missing connections using outer chains of ancilla.

VI. QUANTUM FOURIER TRANSFORMATION

Having just seen the varying degrees to which the Poughkeepsie architecture can handle CCNOT circuits, we now turn to another critical subroutine for quantum computing, the Quantum Fourier Transformation (QFT). Many promising quantum algorithms, including the famous Shor’s Algorithm, require the use of the QFT as an essential ingredient for extracting out a desired final quantum state. In this section, we present experimental results which demonstrate the reliability with which one can successfully perform a 3-qubit QFT using the various qubit geometries outlined earlier.

At its core, the QFT is the quantum equivalent to the Discrete Fourier Transformation, applied to a quantum state. The power of the QFT lies in its ability to apply up to $2^N$ unique phases across the various components of a quantum state, where $N$ is the number of qubits. The core element necessary to any successful QFT is the control-$R_\phi$ gate, which applies an arbitrary phase to a target qubit’s $|1⟩$ state, conditional on a control qubit $|0⟩$.

$$R_\phi|1⟩ \otimes (\alpha|0⟩ + \beta|1⟩) = |1⟩ \otimes (\alpha|0⟩ + e^{i\phi}\beta|1⟩) \quad (3)$$

Just like the CCNOT circuit, the QFT requires full connectivity between all of the qubits. The standard quantum circuit for a 3-qubit QFT is shown below in figure 23 (technically a QFT† circuit, which we discuss in coming section), which we adapt accordingly for the various qubit geometries. When physically implementing these 3-qubit QFT’s, note that the true gate count for each control-$R_\phi$ gate includes two CNOTs and three $R_\phi$ gates, plus any additional SWAP gates for qubits lacking a direct connection (see [14]). As a result, the circuit depth and total gate count for the 3-qubit QFT turns out to be comparable to that of the CCNOT circuit, which in turn will provide some interesting insight when comparing the success rates between the two.

FIG. 22: Success rates found for the "3 Chain" (left) and "1 Chains" (right) circuits, using X gates to reset the ancilla qubits. Each bar represents a unique combination of target and control computational qubits across the two 6-qubit loops shown in figure 18.

FIG. 23: QFT† circuit for three qubits. The QFT† shown above is the circuit tested on the IBM 20-qubit architecture, identical to the QFT circuit in both total gate count and circuit depth, differing only in gate order and phase values.

A. Testing QFT† With Phase Estimation

In order to isolate and benchmark the success of the QFT† circuit in figure 23 in a way similar to the previous sections, one ideally needs an experiment whereby the effect of the QFT† produces a single desirable final state. However, unlike the CCNOT operation whose effect is directly observable by means of the target qubit, the QFT is a more versatile quantum operation whose effect ranges widely based on the state of the qubits it is applied to. To this end, we quantify the fidelity of our QFT† implementations in a manner analogous to the Quantum Phase Estimation Algorithm (QPE) [31], whereby the effect of the final QFT† leaves all of the qubits in a final state containing no superposition.
Quantum Phase Estimation is a quantum algorithm which uses a control-U operation, along with one of its eigenstates |µ⟩, in order to detect some unknown eigenphase e^{iθ}. An example QPE is shown below in the top plot of figure 24. Creating such a circuit is typically very challenging, as both the implementation of arbitrary control-U operators and their eigenstates require clever circuit design. For the purpose of our QFT benchmarking however, we apply the core idea of the QPE in a much simpler form, effectively achieving the states resulting from the control-U operations acting on |µ⟩ with only Rφ gates, illustrated by the bottom plot in figure 24.

FIG. 24: (top) Quantum circuit for a 3-qubit Quantum Phase Estimation Algorithm. (bottom) An equivalent circuit which mimics the effect of the control-U operations through the use of single qubit rotation gates.

By using single qubit rotation gates to initialize the computational qubits, we are able to prepare quantum states just prior to the QFT† with high fidelities, minimizing any additional noise not caused by the QFT† circuit. Additionally, the use of Rφ gates allows for the creation of a wider range of states than typically achievable through the use of control-U gates, which we in turn use for deeper insight in the viability of the QFT in the coming experiments.

B. Perfect Phase Detection

For the case of a 3-qubit QFT†, there are exactly eight choices for φ such that the bottom circuit in figure 24 will result in a single final state for each qubit (no superpositions leftover). These eight values of φ span an even distribution from 0 to π/4, corresponding to the eight unique quantum states from |000⟩ to |111⟩. Shown below in figure 25 are the various success rates for each qubit geometry’s ability to detect these eight perfect phases. In the QFT† fidelity results below, the success rates from 20 - 22 were used to determine which qubits were experimentally tested. Specifically, the top three qubit combinations for each geometry which yielded the highest fidelities were tested here, and then averaged together. For the 3-qubit geometries, the top three qubit combinations for both control-target orientations were tested. Conversely, only the ”3 Chains” orientation for the 6-qubit geometry was tested (preliminary results showed once again a significant decrease in success rates for the ”1 Chains” orientation). For both the 4 and 6-qubit geometries, CNOT gates were used for resetting ancilla qubits.

Beginning with the success rates between the various qubit geometries, the results shown in figure 25 reveal that the 4-qubit geometry lead to the overall highest QFT† fidelities across all eight phases. In addition to the high Success %’s, the accompanying high Perfect % rates suggests that one could also reliably perform further gate operations after the QFT†. Behind the 4-qubit geometry in success rates, the 3 and 6-qubit geometries produced fidelities of the order 50-60 and 30-40% respectively.

Based on the results from the CCNOT experiments, the higher success rates of the 4-qubit geometry may come as a surprise at first glance. However, in analyzing the two circuits, there is a key difference in the ordering of control gates which could be the contributing factor leading to the 4-qubit geometry’s edge. Specifically, the −π/4 and −π/2 control-Rφ gates happen in succession, originating from the same computational qubit. This succession of control gates allows for optimization in both the 3 and 4-qubit circuit designs, whereby both control operations can be performed between a single set of SWAP and CNOT gates respectively. As indicated in the data, these optimizations appear to favor the 4-qubit geometry, allowing for both control operations to steam from a
single excitation of the central ancilla qubit.

In addition to the success rate discrepancies, a second interesting result emerging from the data reveals an alternating pattern in fidelities between phases. One possible explanation for this unexpected trend could be in the complexity of the quantum state just prior to the QFT\(^\dagger\) circuit. Specifically, the superposition states created from the even integer cases of \(\frac{\pi}{4}\) contain at most four unique phases: \(\frac{1}{8}, -\frac{1}{8}, i\frac{1}{8}, -i\frac{1}{8}\). The odd integer cases however produce states which also contain four additional phases: \(\pm\frac{1}{4}, \pm i\frac{1}{4}\). Since these quantum states have a higher demand on precision in superposition, it is possible that they are more sensitive to noise and errors, leading to lower success rates in detection from the QFT\(^\dagger\) circuit.

C. Continuous Phase Detection

Following from the data trends revealed in the previous section, we now present experimental results which are motivated by a more realistic usage of the QPE Algorithm. Specifically, we present results which extend the data shown in figure 25, testing for intermediate values of \(\phi\), ultimately revealing the ability to detect phases which do not match up perfectly with the number of qubits being used for detection. Detecting these ‘non-perfect’ phases comes with an inherent probability of failure, even for a noiseless quantum computer, as the resulting final states from the QFT\(^\dagger\) now contain superposition. Consequently, one experimentally expects lower success rates between the \(2^N\) perfect phases, with the lowest points being exactly halfway between each perfect phase (approximately 40% for a 3-qubit QPE). Figure 26 indeed reveals this trend, illustrating Success % swings of nearly 50% as \(\phi\) changes by as little as \(\frac{\pi}{16}\).

![FIG. 26: QPE Success % rates for the 3 (green circles) and 4-qubit (blue triangles) geometries as a function of phase \(\phi\), each data point represents the measured percentage of states corresponding to that value.](image)

The success rates illustrated in figure 26 are in general agreement with those of figure 25, with fidelities on the order of 60 and 70% between the two geometries. Here however we can see the degree to which these fidelities diminish for the intermediate phases as well. Additionally, we once again see the success rate discrepancies between even and odd integer cases of \(\frac{\pi}{4}\), present in both the 3 and 4-qubit geometries.

As a final metric of interest for the QPE circuit, figure 27 presents the Success % rates as a function of \(\phi\), for the 4-qubit geometry, whereby the criteria for success is now measuring either of the two nearest perfect phase states. Interestingly, the data reveals a more intricate pattern in the QPE circuit’s ability to detect different phase regions. The pattern seemingly shows a repeating trend each \(\pi\) cycle, with the lowest success rates being between \([\frac{3\pi}{4} - \pi]\) and \([\frac{7\pi}{4} - 2\pi]\).

![FIG. 27: QPE Success % rates for the 4-qubit (blue triangles) geometries as a function of phase \(\phi\), tracking the percentage of trials whereby either of nearest two perfect phase states were measured.](image)

The data shown in figure 27 presents an interesting question as to the source of the apparent success rate discrepancies as a function of phase. As shown by the dashed line in the figure, the theoretically output from the QPE circuit should produce a smooth repeating trend between each \(\frac{\pi}{4}\) cycle. However, since the experimental results show that this is seemingly not the case, we leave its explanation as an open question for future research efforts.

VII. CONCLUSIONS

The experimental results in this paper have showcased various qualities of IBM’s 20-qubit chip Poughkeepsie. In analyzing these results, it is important to keep in context the steadily improving technology of quantum computers, specifically superconducting qubits in this case. In the coming years, it is reasonable to expect quantities such as \(T_1\) & \(T_2\) coherence times and gate fidelities to continually improve, eventually pushing their numerical quantities in this study to become outdated.

In anticipation of better qubits however, we believe there are still notable results from this study which go beyond qubit quality. For example, in testing the CNOT
chains across all 20 qubits, the difference in success rates between using X gates versus CNOT gates for resetting ancilla qubits was very pronounced. When combined with other factors such as qubit connectivity or the various qubit geometry results from sections 5 and 6, one could in principle use this knowledge for optimizing future quantum circuits. Ultimately, as more and more is asked of these superconducting quantum computers, we hope that the experimental demonstrations of this study will help influence and improve future research efforts.

Acknowledgments

We gratefully acknowledge support from the National Research Council Associateship Programs and funding from OSD ARAP QSEP program. Special thanks to Dan Campbell for his numerous insightful talks throughout this project. We would also like to thank the IBM Quantum Experience team and all of their support. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of AFRL.

[1] J. Preskill, Quantum, volume 2, page 79 (2006)
[2] P. Shor, SIAM J. Sci. Statist. Comput. 26 1484 (1997)
[3] L. K. Grover, Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (1996)
[4] A. A. Clerk, M. H. Devoret, S. M. Girvin, F. Marquardt and R. J. Schoelkopf, Rev. Mod. Phys. 82, 1155 (2010).
[5] P. V. Klimov et. all, Phys. Rev. Lett. 121, 090502 (2018)
[6] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson and W. D. Oliver, Applied Phys. Rev. 6, 021318 (2019)
[7] M. Kjaergaard, M. E. Schwartz, J. Braumiller, P. Krantz, J. I.-J. Wang, S. Gustavsson and William D. Oliver, arXiv: 1905.13641 (2019)
[8] K. Temme, S. Bravyi, and J. M. Gambetta Phys. Rev. Lett. 119, 180509
[9] R. Harper and S. T. Flammia, Phys. Rev. Lett. 122, 080504 (2019)
[10] P. Murali, D. C. McKay, M. Martonosi and A. Javadi-Abhari, arXiv: 2001.02826 (2020)
[11] P. Shor, Phys. Rev. A 52, R2493 (1995).
[12] A. Calderbank and P. Shor, Phys. Rev. A 54, 1098 (1996).
[13] A. Steane, Phys. Rev. Lett. 77, 793 (1996).
[14] IBM 20-Qubit Poughkeepsie Architecture, https://quantum-computing.ibm.com. Accessed Sep. - Dec. 2019
[15] Google’s Cirq Quantum Computing, https://github.com/quantumlib/Cirq (2019)
[16] Rigetti’s Pyquil Quantum Computing, https://www.rigetti.com/ (2019)
[17] Microsoft’s Azure Quantum Computing, https://azure.microsoft.com/en-us/services/quantum/ (2019)
[18] E. L. Hahn Phys. Rev. 80 (4), 580594 (1950)
[19] N. F. Ramsey, Phys. Rev. 78 (6), 695699. (1950)
[20] X. R. Wang, Y. S. Zheng, and S. Yin, Phys. Rev. B 72, 121303(R) (2005)
[21] D. Coppersmith, arXiv: 0201067 (1994)
[22] T. Toffoli (1980), J. W. de Bakker and J. van Leeuwen (eds.) Automata, Languages and Programming, pg.632 Springer: New York
[23] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, Proceedings of the National Academy of Sciences 114, 13 (2017)
[24] S. S. Tannu and M. K. Qureshi arXiv:quant-ph/1805.10224 (2018)
[25] P. J. Coles, S. Eidenbenz et al., arXiv:1804.03719 (2018)
[26] R. Balu, D. Castillo, and G. Siopsis, Quantum Science and Technology, IOP Publishing (2018)
[27] A. May, L. Schlieper and J. Schwinger, arXiv: 1910.00802 (2019)
[28] H. Mohammadbagherpoor, Y-H. Oh, P. Dreher, A. Singh, X. Yu and A. J. Rindos, arXiv: 1910.11696 (2019)
[29] E. Buksman, A. L. Fonseca de Oliveira and C. Allende, arXiv: 1912.07486 (2019)
[30] P. Q. Cruz, G. Catarina, R. Gautier and J. Fernández-Rossier, arXiv: 1910.06265 (2019)
[31] A. Y. Kitaev, arXiv: 9511026 (1995)