

Asymptotic Improvements to Quantum Circuits via Qutrits

Pranav Gokhale
pranavgokhale@uchicago.edu
University of Chicago

Jonathan M. Baker
jmbaker@uchicago.edu
University of Chicago

Casey Duckering
cduck@uchicago.edu
University of Chicago

Natalie C. Brown
natalie.c.brown@duke.edu
Georgia Institute of Technology

Kenneth R. Brown
kenneth.r.brown@duke.edu
Duke University

Frederic T. Chong
chong@cs.uchicago.edu
University of Chicago

ABSTRACT

Quantum computation is traditionally expressed in terms of quantum bits, or qubits. In this work, we instead consider three-level qutrits. Past work with qutrits has demonstrated only constant factor improvements, owing to the $\log_2(3)$ binary-to-ternary compression factor. We present a novel technique using qutrits to achieve a logarithmic depth (runtime) decomposition of the Generalized Toffoli gate using no ancilla—a significant improvement over linear depth for the best qubit-only equivalent. Our circuit construction also features a 70x improvement in two-qudit gate count over the qubit-only equivalent decomposition. This results in circuit cost reductions for important algorithms like quantum neurons and Grover search. We develop an open-source circuit simulator for qutrits, along with realistic near-term noise models which account for the cost of operating qutrits. Simulation results for these noise models indicate over 90% mean reliability (fidelity) for our circuit construction, versus under 30% for the qubit-only baseline. These results suggest that qutrits offer a promising path towards scaling quantum computation.

CCS CONCEPTS

• Computer systems organization → Quantum computing.

KEYWORDS

quantum computing, quantum information, qutrits

ACM Reference Format:

Pranav Gokhale, Jonathan M. Baker, Casey Duckering, Natalie C. Brown, Kenneth R. Brown, and Frederic T. Chong. 2019. Asymptotic Improvements to Quantum Circuits via Qutrits. In *ISCA '19: 46th International Symposium on Computer Architecture*, June 22–26, 2019, PHOENIX, AZ, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3307650.3322253>

1 INTRODUCTION

Recent advances in both hardware and software for quantum computation have demonstrated significant progress towards practical outcomes. In the coming years, we expect quantum computing

will have important applications in fields ranging from machine learning and optimization [1] to drug discovery [2]. While early research efforts focused on longer-term systems employing full error correction to execute large instances of algorithms like Shor factoring [3] and Grover search [4], recent work has focused on NISQ (Noisy Intermediate Scale Quantum) computation [5]. The NISQ regime considers near-term machines with just tens to hundreds of quantum bits (qubits) and moderate errors.

Given the severe constraints on quantum resources, it is critical to fully optimize the compilation of a quantum algorithm in order to have successful computation. Prior architectural research has explored techniques such as mapping, scheduling, and parallelism [6–8] to extend the amount of useful computation possible. In this work, we consider another technique: quantum trits (qutrits).

While quantum computation is typically expressed as a two-level binary abstraction of qubits, the underlying physics of quantum systems are not intrinsically binary. Whereas classical computers operate in binary states at the physical level (e.g. clipping above and below a threshold voltage), quantum computers have natural access to an infinite spectrum of discrete energy levels. In fact, hardware must actively suppress higher level states in order to achieve the two-level qubit approximation. Hence, using three-level qutrits is simply a choice of including an additional discrete energy level, albeit at the cost of more opportunities for error.

Prior work on qutrits (or more generally, d-level *qudits*) identified only constant factor gains from extending beyond qubits. In general, this prior work [9] has emphasized the information compression advantages of qutrits. For example, N qubits can be expressed as $\frac{N}{\log_2(3)}$ qutrits, which leads to $\log_2(3) \approx 1.6$ -constant factor improvements in runtimes.

Our approach utilizes qutrits in a novel fashion, essentially using the third state as temporary storage, but at the cost of higher per-operation error rates. Under this treatment, the runtime (i.e. circuit depth or critical path) is *asymptotically* faster, and the reliability of computations is also improved. Moreover, our approach only applies qutrit operations in an intermediary stage: the input and output are still qubits, which is important for initialization and measurement on real devices [10, 11].

The net result of our work is to extend the frontier of what quantum computers can compute. In particular, the frontier is defined by the zone in which every machine qubit is a data qubit, for example a 100-qubit algorithm running on a 100-qubit machine. This is indicated by the yellow region in Figure 1. In this frontier zone, we do not have room for non-data workspace qubits known as ancilla. The lack of ancilla in the frontier zone is a costly constraint that

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ISCA '19, June 22–26, 2019, PHOENIX, AZ, USA

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-6669-4/19/06...\$15.00

<https://doi.org/10.1145/3307650.3322253>

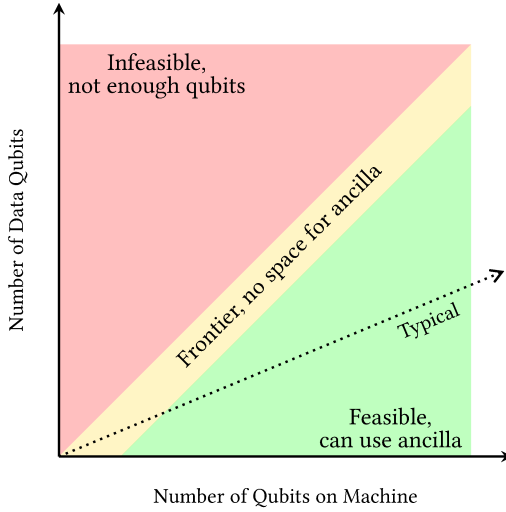


Figure 1: The frontier of what quantum hardware can execute is the yellow region adjacent to the 45° line. In this region, each machine qubit is a data qubit. Typical circuits rely on non-data ancilla qubits for workspace and therefore operate below the frontier.

generally leads to inefficient circuits. For this reason, typical circuits instead operate below the frontier zone, with many machine qubits used as ancilla. Our work demonstrates that ancilla can be substituted with qutrits, enabling us to operate efficiently within the ancilla-free frontier zone.

We highlight the three primary contributions of our work:

- (1) A circuit construction based on qutrits that leads to asymptotically faster circuits ($633N \rightarrow 38 \log_2 N$) than equivalent qubit-only constructions. We also reduce total gate counts from $397N$ to $6N$.
- (2) An open-source simulator, based on Google’s Cirq [12], which supports realistic noise simulation for qutrit (and qudit) circuits.
- (3) Simulation results, under realistic noise models, which demonstrate our circuit construction outperforms equivalent qubit circuits in terms of error. For our benchmarked circuits, our reliability advantage ranges from 2x for trapped ion noise models up to more than 10,000x for superconducting noise models. For completeness, we also benchmark our circuit against a qubit-only construction augmented by an ancilla and find our construction is still more reliable.

The rest of this paper is organized as follows: Section 2 presents relevant background about quantum computation and Section 3 outlines related prior work that we benchmark our work against. Section 4 demonstrates our key circuit construction, and Section 5 surveys applications of this construction toward important quantum algorithms. Section 6 introduces our open-source qudit circuit simulator. Section 7 explains our noise modeling methodology (with full details in Appendix A), and Section 8 presents simulation results for these noise models. Finally, we discuss our results at a higher level in Section 9.

2 BACKGROUND

A qubit is the fundamental unit of quantum computation. Compared to their classical counterparts which take values of either 0 and 1, qubits may exist in a superposition of the two states. We designate these two basis states as $|0\rangle$ and $|1\rangle$ and can represent any qubit as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $\|\alpha\|^2 + \|\beta\|^2 = 1$. $\|\alpha\|^2$ and $\|\beta\|^2$ correspond to the probabilities of measuring $|0\rangle$ and $|1\rangle$ respectively.

Quantum states can be acted on by quantum gates which (a) preserve valid probability distributions that sum to 1 and (b) guarantee reversibility. For example, the X gate transforms a state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ to $X|\psi\rangle = \beta|0\rangle + \alpha|1\rangle$. The X gate is also an example of a classical reversible operation, equivalent to the NOT operation. In quantum computation, we have a single irreversible operation called measurement that transforms a quantum state into one of the two basis states with a given probability based on α and β .

In order to interact different qubits, two-qubit operations are used. The CNOT gate appears both in classical reversible computation and in quantum computation. It has a control qubit and a target qubit. When the control qubit is in the $|1\rangle$ state, the CNOT performs a NOT operation on the target. The CNOT gate serves a special role in quantum computation, allowing quantum states to become entangled so that a pair of qubits cannot be described as two individual qubit states. Any operation may be conditioned on one or more controls.

Many classical operations, such as AND and OR gates, are irreversible and therefore cannot directly be executed as quantum gates. For example, consider the output of 1 from an OR gate with two inputs. With only this information about the output, the value of the inputs cannot be uniquely determined. These operations can be made reversible by the addition of extra, temporary workspace bits initialized to 0. Using a single additional ancilla, the AND operation can be computed reversibly as in Figure 2.

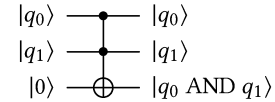


Figure 2: Reversible AND circuit using a single ancilla bit. The inputs are on the left, and time flows rightward to the outputs. This AND gate is implemented using a Toffoli (CC-NOT) gate with inputs q_0 , q_1 and a single ancilla initialized to 0. At the end of the circuit, q_0 and q_1 are preserved, and the ancilla bit is set to 1 if and only if both other inputs are 1.

Physical systems in classical hardware are typically binary. However, in common quantum hardware, such as in superconducting and trapped ion computers, there is an infinite spectrum of discrete energy levels. The qubit abstraction is an artificial approximation achieved by suppressing all but the lowest two energy levels. Instead, the hardware may be configured to manipulate the lowest three energy levels by operating on qutrits. In general, such a computer could be configured to operate on any number of d levels, except as d increases the number of opportunities for error, termed

error channels, increases. Here, we focus on $d = 3$ with which we achieve the desired improvements to the Generalized Toffoli gate.

In a three level system, we consider the computational basis states $|0\rangle$, $|1\rangle$, and $|2\rangle$ for qutrits. A qutrit state $|\psi\rangle$ may be represented analogously to a qubit as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle$, where $\|\alpha\|^2 + \|\beta\|^2 + \|\gamma\|^2 = 1$. Qutrits are manipulated in a similar manner to qubits; however, there are additional gates which may be performed on qutrits.

For instance, in quantum binary logic, there is only a single X gate. In ternary, there are three X gates denoted X_{01} , X_{02} , and X_{12} . Each of these X_{ij} for $i \neq j$ can be viewed as swapping $|i\rangle$ with $|j\rangle$ and leaving the third basis element unchanged. For example, for a qutrit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle$, applying X_{02} produces $X_{02}|\psi\rangle = \gamma|0\rangle + \beta|1\rangle + \alpha|2\rangle$. Each of these operations' actions can be found in the left state diagram in Figure 3.

There are two additional non-trivial operations on a single trit. They are the $+1$ and -1 (sometimes referred to as a $+2$) operations (with $+$ meaning addition modulo 3). These operations can be written as $X_{01}X_{12}$ and $X_{12}X_{01}$, respectively; however, for simplicity, we will refer to them as X_{+1} and X_{-1} operations. A summary of these gates' actions can be found in the right state diagram in Figure 3.

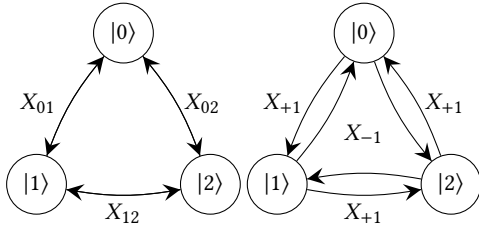


Figure 3: The five nontrivial permutations on the basis elements for a qutrit. (Left) Each operation here switches two basis elements while leaving the third unchanged. These operations are self-inverses. (Right) These two operations permute the three basis elements by performing a $+1 \bmod 3$ and $-1 \bmod 3$ operation. They are each other's inverses.

Other, non-classical, operations may be performed on a single qutrit. For example, the Hadamard gate [13] can be extended to work on qutrits in a similar fashion as the X gate was extended. In fact, all single qubit gates, like rotations, may be extended to operate on qutrits. In order to distinguish qubit and qutrit gates, all qutrit gates will appear with an appropriate subscript.

Just as single qubit gates have qutrit analogs, the same holds for two qutrit gates. For example, consider the CNOT operation, where an X gate is performed conditioned on the control being in the $|1\rangle$ state. For qutrits, any of the X gates presented above may be performed, conditioned on the control being in any of the three possible basis states. Just as qubit gates are extended to take multiple controls, qutrit gates are extended similarly. The set of single qutrit gates, augmented by any entangling two-qutrit gate, is sufficient for universality in ternary quantum computation [14].

One question concerning the feasibility of using higher states beyond the standard two is whether these gates can be implemented

and perform the desired manipulations. Qutrit gates have been successfully implemented [15–17] indicating it is possible to consider higher level systems apart from qubit only systems.

In order to evaluate a decomposition of a quantum circuit, we consider quantum circuit costs. The space cost of a circuit, i.e. the number of qubits (or qutrits), is referred to as circuit *width*. Requiring ancilla increases the circuit width and therefore the space cost of a circuit. The time cost for a circuit is the *depth* of a circuit. The depth is given as the length of the critical path (in terms of gates) from input to output.

3 PRIOR WORK

3.1 Qudits

Qutrits, and more generally qudits, have been studied in past work both experimentally and theoretically. Experimentally, d as large as 10 has been achieved (including with two-qudit operations) [18], and $d = 3$ qutrits are commonly used internally in many quantum systems [19, 20].

However, in past work, qudits have conferred only an information compression advantage. For example, N qubits can be compressed to $\frac{N}{\log_2(d)}$ qudits, giving only a constant-factor advantage [9] at the cost of greater errors from operating qudits instead of qubits. Under the assumption of linear cost scaling with respect to d , it has been demonstrated that $d = 3$ is optimal [21, 22], although as we show in Section 7 the cost is generally superlinear in d .

The information compression advantage of qudits has been applied specifically to Grover's search algorithm [23–26] and to Shor's factoring algorithm [27]. Ultimately, the tradeoff between information compression and higher per-qudit errors has not been favorable in past work. As such, the past research towards building practical quantum computers has focused on qubits.

Our work introduces qutrit-based circuits which are *asymptotically* better than equivalent qubit-only circuits. Unlike prior work, we demonstrate a compelling advantage in both runtime and reliability, thus justifying the use of qutrits.

3.2 Generalized Toffoli Gate

We focus on the Generalized Toffoli gate, which simply adds more controls to the Toffoli circuit in Figure 2. The Generalized Toffoli gate is an important primitive used across a wide range of quantum algorithms, and it has been the focus of extensive past optimization work. Table 1 compares past circuit constructions for the Generalized Toffoli gate to our construction, which is presented in full in Section 4.2.

Among prior work, the Gidney [28], He [29], and Barenco [30] designs are all qubit-only. The three circuits have varying tradeoffs. While Gidney and Barenco operate at the ancilla-free frontier, they have large circuit depths: linear with a large constant for Gidney and quadratic for Barenco. The Gidney design also requires rotation gates for very small angles, which poses an experimental challenge. While the He circuit achieves logarithmic depth, it requires an ancilla for each data qubit, effectively halving the effective potential of any given quantum hardware. Nonetheless, in practice, most circuit implementations use these linear-ancilla constructions due to their small depths and gate counts.

	This Work	Gidney [28]	He [29]	Barenco [30]	Wang [25]	Lanyon [31], Ralph [32]
Depth	$\log N$	N	$\log N$	N^2	N	N
Ancilla	0	0	N	0	0	0
Qudit Types	Controls are qutrits	Qubits	Qubits	Qubits	Controls are qutrits	Target is $d = N$ -level qudit
Constants	Small	Large	Small	Small	Small	Small

Table 1: Asymptotic comparison of N -controlled gate decompositions. The total gate count for all circuits scales linearly (except for Barenco [30], which scales quadratically). Our construction uses qutrits to achieve logarithmic depth without ancilla. We benchmark our circuit construction against Gidney [28], which is the asymptotically best ancilla-free qubit circuit.

As in our approach, circuit constructions from Lanyon [31], Ralph [32], and Wang [25] have attempted to improve the ancilla-free Generalized Toffoli gate by using qudits. Both the Lanyon [31] and Ralph [32] constructions, which have been demonstrated experimentally, achieve linear circuit depths by operating the target as a $d = N$ -level qudit. Wang [25] also achieves a linear circuit depth but by operating each control as a qutrit.

Our circuit construction, presented in Section 4.2, has similar structure to the He design, which can be represented as a binary tree of gates. However, instead of storing temporary results with a linear number of ancilla qubits, our circuit temporarily stores information directly in the qutrit $|2\rangle$ state of the controls. Thus, no ancilla are needed.

In our simulations, we benchmark our circuit construction against the Gidney construction [28] because it is the asymptotically best qubit circuit in the ancilla-free frontier zone. We label these two benchmarks as **QUTRIT** and **QUBIT**. The **QUBIT** circuit handles the lack of ancilla by using *dirty* ancilla, which unlike *clean* (initialized to $|0\rangle$) ancilla, can have an unknown initial state. Dirty ancilla can therefore be bootstrapped internally from a quantum circuit. However, this technique requires a large number of Toffoli gates which makes the decomposition particularly expensive in gate count.

Augmenting the base Gidney construction with a single ancilla¹ does reduce the constants for the decomposition significantly, although the asymptotic depth and gate counts are maintained. For completeness, we also benchmark our circuit against this augmented construction, **QUBIT+ANCILLA**. However, the augmented circuit does not operate at the ancilla-free frontier, and it conflicts with parallelism, as discussed in Section 9.

4 CIRCUIT CONSTRUCTION

In order for quantum circuits to be executable on hardware, they are typically decomposed into single- and two- qudit gates. Performing efficient low depth and low gate count decompositions is important in both the NISQ regime and beyond. Our circuits assume all-to-all connectivity—we discuss this assumption in Section 9.

4.1 Key Intuition

To develop intuition for our technique, we first present a Toffoli gate decomposition which lays the foundation for our generalization to multiple controls. In each of the following constructions, all inputs and outputs are qubits, but we may occupy the $|2\rangle$ state temporarily during computation. Maintaining binary input and

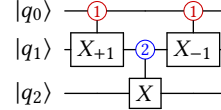


Figure 4: A Toffoli decomposition via qutrits. Each input and output is a qubit. The red controls activate on $|1\rangle$ and the blue controls activate on $|2\rangle$. The first gate temporarily elevates q_1 to $|2\rangle$ if both q_0 and q_1 were $|1\rangle$. We then perform the X operation only if q_1 is $|2\rangle$. The final gate restores q_0 and q_1 to their original state.

output allows these circuit constructions to be inserted into any preexisting qubit-only circuits.

In Figure 4, a Toffoli decomposition using qutrits is given. A similar construction for the Toffoli gate is known from past work [31, 32]. The goal is to perform an X operation on the last (target) input qubit q_2 if and only if the two control qubits, q_0 and q_1 , are both $|1\rangle$. First a $|1\rangle$ -controlled X_{+1} is performed on q_0 and q_1 . This elevates q_1 to $|2\rangle$ iff q_0 and q_1 were both $|1\rangle$. Then a $|2\rangle$ -controlled X gate is applied to q_2 . Therefore, X is performed only when both q_0 and q_1 were $|1\rangle$, as desired. The controls are restored to their original states by a $|1\rangle$ -controlled X_{-1} gate, which undoes the effect of the first gate. The key intuition in this decomposition is that the qutrit $|2\rangle$ state can be used instead of ancilla to store temporary information.

4.2 Generalized Toffoli Gate

We now present our circuit decomposition for the Generalized Toffoli gate in Figure 5. The decomposition is expressed in terms of three-qutrit gates (two controls, one target) instead of single- and two- qutrit gates, because the circuit can be understood purely classically at this granularity. In actual implementation and in our simulation, we used a decomposition [15] that requires 6 two-qutrit and 7 single-qutrit physically implementable quantum gates.

Our circuit decomposition is most intuitively understood by treating the left half of the the circuit as a tree. The desired property is that the root of the tree, q_7 , is $|2\rangle$ if and only if each of the 15 controls was originally in the $|1\rangle$ state. To verify this property, we observe the root q_7 can only become $|2\rangle$ iff q_7 was originally $|1\rangle$ and q_3 and q_{11} were both previously $|2\rangle$. At the next level of the tree, we see q_3 could have only been $|2\rangle$ if q_3 was originally $|1\rangle$ and both q_1 and q_5 were previously $|2\rangle$, and similarly for the other triplets. At the bottom level of the tree, the triplets are controlled on the

¹This ancilla can also be dirty.

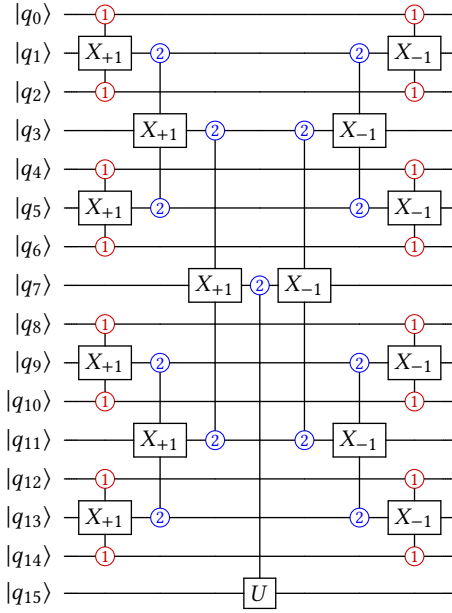


Figure 5: Our circuit decomposition for the Generalized Toffoli gate is shown for 15 controls and 1 target. The inputs and outputs are both qubits, but we allow occupation of the $|2\rangle$ qutrit state in between. The circuit has a tree structure and maintains the property that the root of each subtree can only be elevated to $|2\rangle$ if all of its control leaves were $|1\rangle$. Thus, the U gate is only executed if all controls are $|1\rangle$. The right half of the circuit performs uncomputation to restore the controls to their original state. This construction applies more generally to any multiply-controlled U gate. Note that the three-input gates are decomposed into 6 two-input and 7 single-input gates in our actual simulation, as based on the decomposition in [15].

$|1\rangle$ state, which are only activated when the even-index controls are all $|1\rangle$. Thus, if any of the controls were not $|1\rangle$, the $|2\rangle$ states would fail to propagate to the root of the tree. The right half of the circuit performs *uncomputation* to restore the controls to their original state.

After each subsequent level of the tree structure, the number of qubits under consideration is reduced by a factor of ~ 2 . Thus, the circuit depth is logarithmic in N . Moreover, each qutrit is operated on by a constant number of gates, so the total number of gates is linear in N .

Our circuit decomposition still works in a straightforward fashion when the control type of the top qubit, q_0 , activates on $|2\rangle$ or $|0\rangle$ instead of activating on $|1\rangle$. These two constructions are necessary for the Incrementer circuit in 5.3.

We verified our circuits, both formally and via simulation. Our verification scripts are available on our GitHub [33].

5 APPLICATION TO ALGORITHMS

The Generalized Toffoli gate is an important primitive in a broad range of quantum algorithms. In this section, we survey some of the applications of our circuit decomposition.

5.1 Artificial Quantum Neuron

The artificial quantum neuron [34] is a promising target application for our circuit construction, because the algorithm's circuit implementation is dominated by large Generalized Toffoli gates. The algorithm may exhibit an exponential advantage over classical perceptron encoding and it has already been executed on current quantum hardware. Moreover, the threshold behavior of perceptrons has inherent noise resilience, which makes the artificial quantum neuron particularly promising as a near-term application on noisy systems. The current implementation of the neuron on IBM quantum computers relies on ancilla qubits [35] which constrains the circuit width to $N = 4$ data qubits. Our circuit construction offers a path to larger circuit sizes without waiting for larger hardware.

5.2 Grover's Algorithm

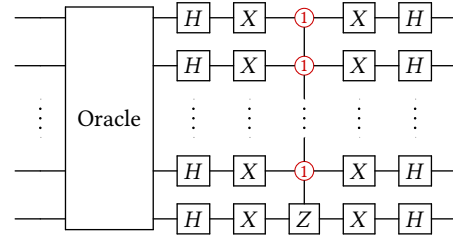


Figure 6: Each iteration of Grover Search has a multiply-controlled Z gate. Our logarithmic depth decomposition, reduces a $\log M$ factor in Grover's algorithm to $\log \log M$.

Grover's Algorithm for search over M unordered items requires just $O(\sqrt{M})$ oracle queries. However, each oracle query is followed by a post-processing step which requires a multiply-controlled gate with $N = \lceil \log_2 M \rceil$ controls [13]. The explicit circuit diagram is shown in Figure 6.

Our log-depth circuit construction directly applies to the multiply-controlled gate. Thus, we reduce a $\log M$ factor in Grover search's time complexity to $\log \log M$ via our ancilla-free qutrit decomposition.

5.3 Incrementer

The Incrementer circuit performs the $+1 \bmod 2^N$ operation to a register of N qubits. While logarithmic circuit depth can be achieved with linear ancilla qubits [36], the best ancilla-free incrementers require either linear depth with large linearity constants [37] or quadratic depth [30]. Using alternate control activations for our Generalized Toffoli gate decomposition, the incrementer circuit is reduced to $O(\log^2 N)$ depth with no ancilla, a significant improvement over past work.

Our incrementer circuit construction is shown in Figure 7 for an $N = 8$ wide register. The multiple-controlled X_{+1} gates perform the

job of computing carries: a carry is performed iff the least significant bit generates (represented by the $|2\rangle$ control) and all subsequent bits propagate (represented by the consecutive $|1\rangle$ controls). We present an $N = 8$ incrementer here and have verified the general construction, both by formal proof and by explicit circuit simulation for larger N .

The critical path of this circuit is the chain of $\log N$ multiply-controlled gates (of width $\frac{N}{2}, \frac{N}{4}, \frac{N}{8}, \dots$) which act on $|a_0\rangle$. Since our multiply-controlled gate decomposition has log-depth, we arrive at a total circuit depth circuit scaling of $\log^2 N$.

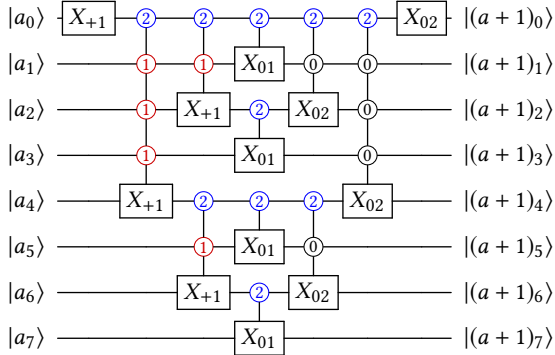


Figure 7: Our circuit decomposition for the Incrementer. At each subcircuit in the recursive design, multiply-controlled gates are used to efficiently propagate carries over half of the subcircuit. The $|2\rangle$ control checks for carry generation and the chain of $|1\rangle$ controls checks for carry propagation. The circuit depth is $\log^2 N$, which is only possible because of our log depth multiply-controlled gate primitive.

5.4 Arithmetic Circuits and Shor’s Algorithm

The Incrementer circuit is a key subcircuit in many other arithmetic circuits such as constant addition, modular multiplication, and modular exponentiation. Further, the modular exponentiation circuit is the bottleneck in the runtime for executing Shor’s algorithm for factorization [37, 38]. While a shallower Incrementer circuit alone is not sufficient to reduce the asymptotic cost of modular exponentiation (and therefore Shor’s algorithm), it does reduce constants relative to qubit-only circuits.

5.5 Error Correction and Fault Tolerance

The Generalized Toffoli gate has applications to circuits for both error correction [39] and fault tolerance [40]. We foresee two paths of applying these circuits. First, our circuit construction can be used to construct error-resilient *logical qubits* more efficiently. This is critical for quantum algorithms like Grover’s and Shor’s which are expected to require such logical qubits. In the nearer-term, NISQ algorithms are likely to make use of limited error correction. For instance, recent results have demonstrated that error correcting a single qubit at a time for the Variational Quantum Eigensolver algorithm can significantly reduce total error [41]. Thus, our circuit construction is also relevant for NISQ-era error correction.

6 SIMULATOR

To simulate our circuit constructions, we developed a qudit simulation library, built on Google’s Cirq Python library [12]. Cirq is a qubit-based quantum circuit library and includes a number of useful abstractions for quantum states, gates, circuits, and scheduling.

Our work extends Cirq by discarding the assumption of two-level qubit states. Instead, all state vectors and gate matrices are expanded to apply to d -level qudits, where d is a circuit parameter. We include a library of common gates for $d = 3$ qutrits. Our software adds a comprehensive noise simulator, detailed below in Section 6.1.

In order to verify our circuits are logically correct, we first simulated them with noise disabled. We extended Cirq to allow gates to specify their action on classical non-superposition input states without considering full state vectors. Therefore, each classical input state can be verified in space and time proportional to the circuit width. By contrast, Cirq’s default simulation procedure relies on a dense state vector representation requiring space and time exponential in the circuit width. Reducing this scaling from exponential to linear dramatically improved our verification procedure, allowing us to verify circuit constructions for all possible classical inputs across circuit sizes up to widths of 14.

Our software is fully open source [33].

6.1 Noise Simulation

Figure 8 depicts a schematic view of our noise simulation procedure which accounts for both gate errors and idle errors, described below. To determine when to apply each gate and idle error, we use Cirq’s scheduler which schedules each gate as early as possible, creating a sequence of Moments of simultaneous gates. During each Moment, our noise simulator applies a gate error to every qudit acted on. Finally, the simulator applies an idle error to every qudit. This noise simulation methodology is consistent with previous simulation techniques which have accounted for either gate errors [42] or idle errors [43].

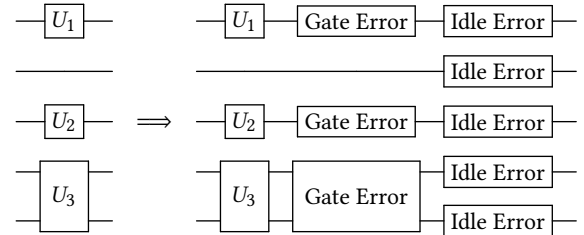


Figure 8: This Moment comprises three gates executed in parallel. To simulate with noise, we first apply the ideal gates, followed by a gate error noise channel on each affected qudit. This gate error noise channel depends on whether the corresponding gate was single- or two- qudit. Finally, we apply an idle error to every qudit. The idle error noise channel depends on the duration of the Moment.

Gate errors arise from the imperfect application of quantum gates. Two-qudit gates are noisier than single-qudit gates [44], so we apply different noise channels for the two. Our specific gate error probabilities are given in Section 7.

Idle errors arise from the continuous decoherence of a quantum system due to energy relaxation and interaction with the environment. The idle errors differ from gate errors in two ways which require special treatment:

- (1) Idle errors depend on duration, which in turn depend on the schedule of simultaneous gates (Moments). In particular, two-qutrit gates take longer to apply than single-qutrit gates. Thus, if a Moment contains a two-qutrit gate, the idling errors must be scaled appropriately. Our specific scaling factors are given in Section 7.
- (2) For the generic model of gate errors, the error channel is applied with probability independent of the quantum state. This is not true for idle errors such as T_1 amplitude damping, which only applies when the qutrit is in an excited state. This is treated in the simulator by computing idle error probabilities during each Moment, for each qutrit.

Gate errors are reduced by performing fewer *total gates*, and idle errors are reduced by decreasing the circuit *depth*. Since our circuit constructions asymptotically decrease the depth, this means our circuit constructions scale favorably in terms of asymptotically fewer idle errors.

Our full noise simulation procedure is summarized in Algorithm 1. The ultimate metric of interest is the mean *fidelity*, which is defined as the squared overlap between the ideal (noise-free) and actual output state vectors. Fidelity expresses the probability of overall successful execution. We do not consider initialization errors and readout errors, because our circuit constructions maintain binary input and output, only occupying the qutrit $|2\rangle$ states during intermediate computation. Therefore, the initialization and readout errors for our circuits are identical to those for conventional qubit circuits.

We also do not consider crosstalk errors, which occur when gates are executed in parallel. The effect of crosstalk is very device-dependent and difficult to generalize. Moreover, crosstalk can be mitigated by breaking each Moment into a small number of sub-moments and then scheduling two-qutrit operations to reduce crosstalk, as demonstrated in prior work [45, 46].

6.2 Simulator Efficiency

Simulating a quantum circuit with a classical computer is, in general, exponentially difficult in the size of the input because the state of N qutrits is represented by a state vector of d^N complex numbers. For 14 qutrits, with complex numbers stored as two 8-byte floats (complex128 in NumPy), a state vector occupies 77 megabytes.

A naive circuit simulation implementation would treat every quantum gate or Moment as a $d^N \times d^N$ matrix. For 14 qutrits, a single such matrix would occupy 366 terabytes—out of range of simulability. While the exponential nature of simulating our circuits is unavoidable, we mitigate the cost by using a variety of techniques which rely only on state vectors, rather than full square matrices. For example, we maintain Cirq’s approach of applying gates by Einstein Summation [47], which obviates computation of the $d^N \times d^N$ matrix corresponding to every gate or Moment.

Our noise simulator only relies on state vectors, by adopting the quantum trajectory methodology [48, 49], which is also used by the Rigetti PyQuil noise simulator [50]. At a high level, the effect of

```

 $|\Psi\rangle \leftarrow$  random initial state vector
 $|\Psi\rangle_{\text{ideal}} =$  circuit applied to  $|\Psi\rangle$  without noise

foreach Moment do
    foreach Gate  $\in$  Moment do
         $|\psi\rangle \leftarrow$  Gate applied to  $|\psi\rangle$ 
        GateError  $\leftarrow$  DrawRand(GateError Prob.)
         $|\psi\rangle \leftarrow$  GateError applied to  $|\psi\rangle$ 
    end
    foreach Qutrit do
        if Moment has 2-qutrit gate then
            IdleErrors  $\leftarrow$  long-duration idle errors
        else
            IdleErrors  $\leftarrow$  short-duration idle errors
        end
        Prob.  $\leftarrow$  [ $\|M|\Psi\rangle\|^2$  for  $M \in$  IdleErrors]
        IdleError  $\leftarrow$  DrawRand(Prob.)
         $|\psi\rangle \leftarrow$  IdleError applied to  $|\psi\rangle$ 
        Renormalize( $|\psi\rangle$ )
    end
end
return  $\langle \Psi_{\text{ideal}} | \Psi \rangle^2$ , fidelity between ideal & actual output;

```

Algorithm 1: Pseudocode for each simulation trial, given a particular circuit and noise model.

noise channels like gate and idle errors is to turn a coherent quantum state into an incoherent mix of classical probability-weighted quantum states (for example, $|0\rangle$ and $|1\rangle$ with 50% probability each). The most complete description of such an incoherent quantum state is called the density matrix and has dimension $d^N \times d^N$. The quantum trajectory methodology is a stochastic approach—instead of maintaining a density matrix, only a single state is propagated and the error term is drawn randomly at each timestep. Over repeated trials, the quantum trajectory methodology converges to the same results as from full density matrix simulation [50]. Our simulator employs this technique—each simulation in Algorithm 1 constitutes a single quantum trajectory trial. At every step, a specific GateError or IdleError term is picked, based on a weighted random draw.

Finally, our random state vector generation function was also implemented in $O(d^N)$ space and time. This is an improvement over other open source libraries [51, 52], which perform random state vector generation by generating full $d^N \times d^N$ unitary matrices from a Haar-random distribution and then truncating to a single column. Our simulator directly computes the first column and circumvents the full matrix computation.

With optimizations, our simulator is able to simulate circuits up to 14 qutrits in width. This is in the range as other state-of-the-art noisy quantum circuit simulations [53] (since 14 qutrits $\approx$ 22 qubits). While each simulation trial took several minutes (depending on the particular circuit and noise model), we were able to run trials in parallel over multiple processes and multiple machines, as described in Section 8.

7 NOISE MODELS

In this section, we describe our noise models at a high level, with mathematical details described in Appendix A. We chose noise models which represent realistic near-term machines. We first present a generic, parametrized noise model roughly applicable to all quantum systems. We then present specific parameters, under the generic noise model, which apply to near-term superconducting quantum computers. Finally, we present a specific noise model for trapped ion quantum computers.

7.1 Generic Noise Model

7.1.1 Gate Errors. The scaling of gate errors for a d -level qudit can be roughly summarized as increasing as d^4 for two-qudit gates and d^2 for single-qudit gates. For $d = 2$, there are 4 single-qubit gate error channels and 16 two-qubit gate error channels. For $d = 3$ there are 9 and 81 single- and two- qutrit gate error channels respectively. Consistent with other simulators [43, 50], we use the symmetric depolarizing gate error model, which assumes equal probabilities between each error channel. Under these noise models, two-qutrit gates are $(1 - 80p_2)/(1 - 15p_2)$ times less reliable than two-qubit gates, where p_2 is the probability of each two-qubit gate error channel. Similarly, single-qutrit gates are $(1 - 8p_1)/(1 - 3p_1)$ times less reliable than single-qubit gates, where p_1 is the probability of each single-qubit gate error channel.

7.1.2 Idle Errors. Our treatment of idle errors focuses on the relaxation from higher to lower energy states in quantum devices. This is called amplitude damping or T_1 relaxation. This noise channel irreversibly takes qudits to lower states. For qubits, the only amplitude damping channel is from $|1\rangle$ to $|0\rangle$, and we denote this damping probability as λ_1 . For qutrits, we also model damping from $|2\rangle$ to $|0\rangle$, which occurs with probability λ_2 .

7.2 Superconducting QC

We chose four noise models based on superconducting quantum computers expected in the next few years. These noise models comply with the generic noise model above and are thus parametrized by p_1, p_2, λ_1 and λ_2 . The λ_i probabilities are derived from two other experimental parameters: the gate time Δt and T_1 , a timescale that captures how long a qudit persists coherently.

As a starting point for representative near-term noise models, we consider parameters for *current* superconducting quantum computers. For IBM’s public cloud-accessible superconducting quantum computers, we have $3p_1 \approx 10^{-3}$ and $15p_2 \approx 10^{-2}$. The duration of single- and two- qubit gates is $\Delta t \approx 100\text{ns}$ and $\Delta t \approx 300\text{ns}$ respectively, and the IBM devices have $T_1 \approx 100\mu\text{s}$ [44, 54].

However, simulation for these current parameters indicates an error is almost certain to occur during execution of a modest size 14-input Generalized Toffoli circuit. This motivates us to instead consider noise models for better devices which are a few years away. Accordingly, we adopt a baseline superconducting noise model, labeled as **SC**, corresponding to a superconducting device which has 10x lower gate errors and 10x longer T_1 duration than the current IBM hardware. This range of parameters has already been achieved experimentally in superconducting devices for gate errors [55, 56] and for T_1 duration [57, 58] independently. Faster gates

Noise Model	$3p_1$	$15p_2$	T_1
SC	10^{-4}	10^{-3}	1 ms
SC+T1	10^{-4}	10^{-3}	10 ms
SC+GATES	10^{-5}	10^{-4}	1 ms
SC+T1+GATES	10^{-5}	10^{-4}	10 ms

Table 2: Noise models simulated for superconducting devices. Current publicly accessible IBM superconducting quantum computers have single- and two- qubit gate errors of $3p_1 \approx 10^{-3}$ and $15p_2 \approx 10^{-2}$, as well as T_1 lifetimes of 0.1 ms [44, 54]. Our baseline benchmark, **SC, assumes 10x better gate errors and T_1 . The other three benchmarks add a further 10x improvement to T_1 , gate errors, or both.**

(shorter Δt) are yet another path towards greater noise resilience. We do not vary gate speeds, because errors only depend on the $\Delta t/T_1$ ratio, and we already vary T_1 . In practice however, faster gates could also improve noise-resilience.

We also consider three additional near-term device noise models, indexed to the **SC** noise model. These three models further improve gate errors, T_1 , or both, by a 10x factor. The specific parameters are given in Table 2. Our 10x improvement projections are realistic extrapolations of progress in hardware. In particular, Schoelkopf’s Law—the quantum analogue of Moore’s Law—has observed that T_1 durations have increased by 10x every 3 years for the past 20 years [59]. Hence, 100x longer T_1 is a reasonable projection for devices that are ~ 6 years away.

7.3 Trapped Ion $^{171}\text{Yb}^+$ QC

We also simulated noise models for trapped ion quantum computing devices. Trapped ion devices are well matched to our qutrit-based circuit constructions because they feature all-to-all connectivity [60], and many ions that are ideal candidates for QC devices are naturally multi-level systems.

We focus on the $^{171}\text{Yb}^+$ ion, which has been experimentally demonstrated as both a qubit and qutrit [10, 11]. Trapped ions are often favored in QC schemes due to their long T_1 times. One of the main advantages of using a trapped ion is the ability to take advantage of magnetically insensitive states known as “clock states.” By defining the computational subspace on these clock states, idle errors caused from fluctuations in the magnetic field are minimized—this is termed a **DRESSED_QUTRIT**, in contrast with a **BARE_QUTRIT**. However, compared to superconducting devices, gates are much slower. Thus, gate errors are the dominant error source for ion trap devices. We modelled a fundamental source of these errors: the spontaneous scattering of photons originating from the lasers used to drive the gates. The duration of single- and two- qubit gates used in this calculation was $\Delta t \approx 1\mu\text{s}$ and $\Delta t \approx 200\mu\text{s}$ respectively [61]. The single- and two- qudit gate error probabilities are given in Table 3.

8 RESULTS

Figure 9 plots the exact circuit depths for all three benchmarked circuits. The qubit-based circuit constructions from past work are linear in depth and have a high linearity constant. Augmenting

Noise Model	p_1	p_2
TI_QUBIT	6.4×10^{-4}	1.3×10^{-4}
BARE_QUTRIT	2.2×10^{-4}	4.3×10^{-4}
DRESSED_QUTRIT	1.5×10^{-4}	3.1×10^{-4}

Table 3: Noise models simulated for trapped ion devices. The single- and two- qutrit gate error channel probabilities are based on calculations from experimental parameters. For all three models, we use single- and two- qutrit gate times of $\Delta t \approx 1 \mu s$ and $\Delta t \approx 200 \mu s$ respectively.

with a single borrowed ancilla reduces the circuit depth by a factor of 8. However, both circuit constructions are surpassed significantly by our qutrit construction, which scales logarithmically in N and has a relatively small leading coefficient.

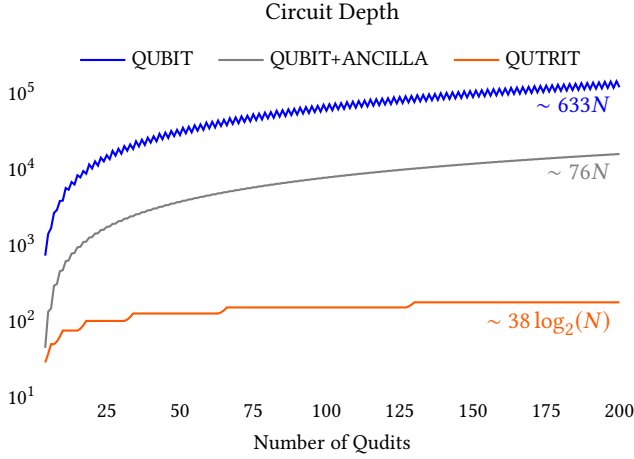


Figure 9: Exact circuit depths for all three benchmarked circuit constructions for the N -controlled Generalized Toffoli up to $N = 200$. Both QUBIT and QUBIT+ANCILLA scale linearly in depth and both are bested by QUTRIT’s logarithmic depth.

Figure 10 plots the total number of two-qutrit gates for all three circuit constructions. As noted in Section 4, our circuit construction is not asymptotically better in total gate count—all three plots have linear scaling. However, as emphasized by the logarithmic vertical axis, the linearity constant for our qutrit circuit is 70x smaller than for the equivalent ancilla-free qubit circuit and 8x smaller than for the borrowed-ancilla qubit circuit.

Our simulations under realistic noise models were run in parallel on over 100 n1-standard-4 Google Cloud instances. These simulations represent over 20,000 CPU hours, which was sufficient to estimate mean fidelity to an error of $2\sigma < 0.1\%$ for each circuit-noise model pair.

The full results of our circuit simulations are shown in Figure 11. All simulations are for the 14-input (13 controls, 1 target) Generalized Toffoli gate. We simulated each of the three circuit benchmarks against each of our noise models (when applicable), yielding the 16 bars in the figure.

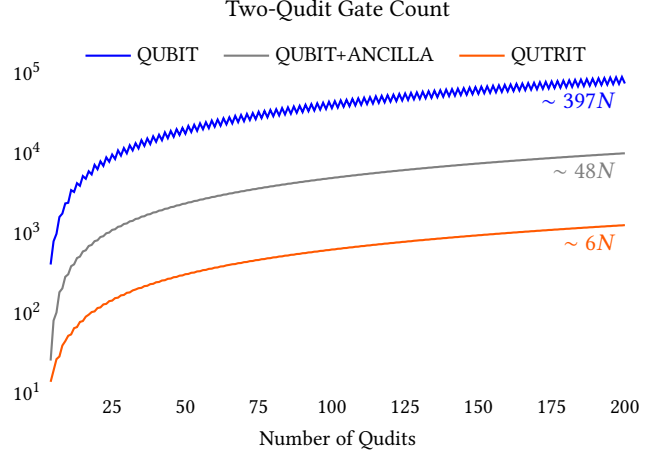


Figure 10: Exact two-qutrit gate counts for the three benchmarked circuit constructions for the N -controlled Generalized Toffoli. All three plots scale linearly; however the QUTRIT construction has a substantially lower linearity constant.

9 DISCUSSION

Figure 11 demonstrates that our QUTRIT construction (orange bars) significantly outperforms the ancilla-free QUBIT benchmark (blue bars) in fidelity (success probability) by more than 10,000x.

For the SC, SC+T1, and SC+GATES noise models, our qutrit constructions achieve between 57-83% mean fidelity, whereas the ancilla-free qubit constructions all have almost 0% fidelity. Only the lowest-error model, SC+T1+GATES achieves modest fidelity of 26% for the QUBIT circuit, but in this regime, the qutrit circuit is close to 100% fidelity.

The trapped ion noise models achieve similar results—the DRESSED_QUTRIT and the BARE_QUTRIT achieve approximately 95% fidelity via the QUTRIT circuit, whereas the TI_QUBIT noise model has only 45% fidelity. Between the dressed and bare qutrits, the dressed qutrit exhibits higher fidelity than the bare qutrit, as expected. Moreover, as discussed in Appendix A.3, the dressed qutrit is resilient to leakage errors, so the simulation results should be viewed as a lower bound on its advantage over the qubit and bare qutrit.

Based on these results, trapped ion qutrits are a particularly strong match to our qutrit circuits. In addition to attaining the highest fidelities, trapped ions generally have all-to-all connectivity [60] within each ion chain, which is critical as our circuit construction requires operations between distant qutrits.

The superconducting noise models also achieve good fidelities. They exhibit a particularly large advantage over ancilla-free qubit constructions because idle errors are significant for superconducting systems, and our qutrit construction significantly reduces idling (circuit depth). However, most superconducting quantum systems only feature nearest-neighbor or short-range connectivity. Accounting for data movement on a nearest-neighbor-connectivity 2D architecture would expand the qutrit circuit depth from $\log N$ to $\sqrt{N}$ (since the distance between any two qutrits would scale as

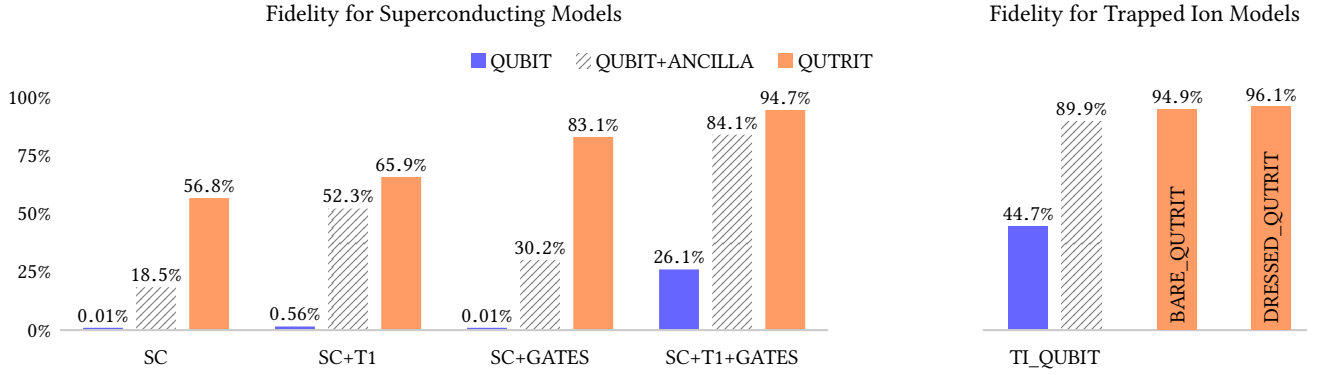


Figure 11: Circuit simulation results for all possible pairs of circuit constructions and noise models. Each bar represents 1000+ trials, so the error bars are all $2\sigma < 0.1\%$. Our QUTRIT construction significantly outperforms the QUBIT construction. The QUBIT+ANCILLA bars are drawn with dashed lines to emphasize that it has access to an extra ancilla bit, unlike our construction.

$\sqrt{N}$). However, recent work has experimentally demonstrated fully-connected superconducting quantum systems via random access memory [62]. Such systems would also be well matched to our circuit construction.

For completeness, Figure 11 also shows fidelities for the QUBIT+ANCILLA circuit benchmark, which augments the ancilla-free QUBIT circuit with a single dirty ancilla. Since QUBIT+ANCILLA has linearity constants $\sim 10\times$ better than the ancilla-free qubit circuit, it exhibits significantly better fidelities. While our QUTRIT circuit still outperforms the QUBIT+ANCILLA circuit, we expect a crossing point where augmenting a qubit-only Generalized Toffoli with enough ancilla would eventually outperform QUTRIT. However, we emphasize that the gap between an ancilla-free and constant-ancilla construction for the Generalized Toffoli is actually a fundamental rather than an incremental gap, because:

- Constant-ancilla constructions prevent circuit parallelization. For example, consider the parallel execution of N/k disjoint Generalized Toffoli gates, each of width k for some constant k . An ancilla-free Generalized Toffoli would pose no issues, but an ancilla-augmented Generalized Toffoli would require $\Theta(N/k)$ ancilla. Thus, constant-ancilla constructions can impose a choice between serializing to linear depth or regressing to linear ancilla count. The Incrementer circuit in Figure 7 is a concrete example of this scenario—any multiply-controlled gate decomposition requiring a single clean ancilla or more than 1 dirty ancilla would contradict the parallelism and reduce runtime.
- Even if we only consider serial circuits, given the exponential advantage of certain quantum algorithms, there is a significant practical difference between operating at the ancilla-free frontier and operating just a few data qubits below the frontier.

While we only performed simulations up to 14 inputs in width, we would see an even bigger advantage in larger circuits because our construction has asymptotically lower depth and therefore asymptotically lower idle errors. We also expect to see an advantage

for the circuits in Section 5 that rely on the Generalized Toffoli, although we did not explicitly simulate these circuits.

Our circuit construction and simulation results point towards promising directions of future work that we highlight below:

- A number of useful quantum circuits, especially arithmetic circuits, make extensive use of multiply-controlled gates. However, these circuits are typically pre-compiled into single- and two- qubit gates using one of the decompositions from prior work, usually one that involves ancilla qubits. Revisiting these arithmetic circuits from first principles, with our qutrit circuit as a new tool, could yield novel and improved circuits like our Incrementer circuit in Section 5.3.
- Relatedly, we see value in a logic synthesis tool that injects qutrit optimizations into qubit circuits, automated in fashion inspired by classical reversible logical synthesis tools [63, 64].
- While $d = 3$ qutrits were sufficient to achieve the desired asymptotic speedups for our circuits of interest, there may be other circuits that are optimized by qudit information carriers for larger d . In particular, we note that increasing d and thereby increasing information compression may be advantageous for hardware with limited connectivity.

Independent of these future directions, the results presented in this work are applicable to quantum computing in the near term, on machines that are expected within the next five years. The net result of this work is to extend the frontier of what is computable by quantum hardware, and hence to accelerate the timeline for practical quantum computing, rather than waiting for better hardware. Emphatically, our results are driven by the use of qutrits for *asymptotically* faster ancilla-free circuits. Moreover, we also improve linearity constants by two orders of magnitudes. Finally, as verified by our open-source circuit simulator coupled with realistic noise models, our circuits are more reliable than qubit-only equivalents. Our results justify the use of qutrits as a path towards scaling quantum computers.

ACKNOWLEDGEMENTS

We would like to thank Michel Devoret and Steven Girvin for suggesting that we investigate qutrits. We also acknowledge David Schuster for helpful discussion on superconducting qutrits. This work is funded in part by EPiQC, an NSF Expedition in Computing, under grants CCF-1730449/1832377, and in part by STAQ, under grant NSF Phy-1818914.

A DETAILED NOISE MODEL

We chose noise models that represent realistic near-term machines. We first present a generic, parametrized noise model in that is roughly applicable to all quantum systems. Next, we present specific parameters, under the generic noise model, that apply to near-term superconducting quantum computers. Finally, we present a specific noise model for $^{171}\text{Yb}^+$ trapped ions.

A.1 Generic Noise Model

The general form of a quantum noise model is expressed by the Kraus Operator formalism which specifies a set of matrices, $\{K_i\}$, each capturing an error channel. Under this formalism, the evolution of a system with initial state $\sigma = |\Psi\rangle\langle\Psi|$ is expressed as a function $\mathcal{E}(\sigma)$, where:

$$\mathcal{E}(\sigma) = \mathcal{E}(|\Psi\rangle\langle\Psi|) = \sum_i K_i \sigma K_i^\dagger \quad (1)$$

where $\dagger$ denotes the matrix conjugate-transpose.

A.1.1 Gate Errors. For a single qubit, there are four possible error channels: no-error, bit flip, phase flip, and phase+bit flip. These channels can be expressed as products of the Pauli matrices:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

which correspond to bit and phase flips respectively. The no-error channel is $X^0 Z^0 = I$ and the phase+bit flip channel is the product $X^1 Z^1$.

In the Kraus operator formalism, we express this single-qubit gate error model as

$$\mathcal{E}(\sigma) = \sum_{j=0}^1 \sum_{k=0}^1 p_{jk} (X^j Z^k) \sigma (X^j Z^k)^\dagger \quad (2)$$

where p_{jk} denotes the probability of the corresponding Kraus operator.

This gate error model is called the Pauli or depolarizing channel. We assume all error terms have equal probabilities, i.e. $p_{jk} = p_1$ for $j, k \neq 0$. This assumption of symmetric depolarizing is standard and is used by most noise simulators [43]. Under this model, the error channel simplifies to:

$$\mathcal{E}(\sigma) = (1 - 3p_1)\sigma + \sum_{jk \in \{0,1\}^2 \setminus 00} p_1 (X^j Z^k) \sigma (X^j Z^k)^\dagger \quad (3)$$

For two-qubit gate errors, the Kraus operators are the Cartesian product of the two single-qubit gate error Kraus operators, leading to the noise channel:

$$\mathcal{E}(\sigma) = (1 - 15p_2)\sigma + \sum_{jklm \in \{0,1\}^4 \setminus 0000} p_2 K_{jklm} \sigma K_{jklm}^\dagger \quad (4)$$

where p_2 is the probability of each error term and $K_{jklm} = X^j Z^k \otimes X^l Z^m$.

Next, for qutrits, we have a similar form, except that there are now more possible error channels. We now use the generalized Pauli matrices:

$$X_{+1} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad Z_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{2\pi i/3} & 0 \\ 0 & 0 & e^{4\pi i/3} \end{pmatrix}$$

The Cartesian product of $\{I, X_{+1}, X_{+1}^2\}$ and $\{I, Z_3, Z_3^2\}$ constitutes a basis for all 3x3 matrices. Hence, this Cartesian product also constitutes the Kraus operators for the single-qutrit gate error [42, 65, 66]:

$$\mathcal{E}(\sigma) = (1 - 8p_1)\sigma + \sum_{jk \in \{0,1,2\}^2 \setminus 00} p_1 (X_{+1}^j Z_3^k) \sigma (X_{+1}^j Z_3^k)^\dagger \quad (5)$$

and similarly, the two-qutrit gate error channel is:

$$\mathcal{E}(\sigma) = (1 - 80p_2)\sigma + \sum_{jklm \in \{0,1,2\}^4 \setminus 0000} p_2 K_{jklm} \sigma K_{jklm}^\dagger \quad (6)$$

Note that in this model, the dominant effect of using qutrits instead of qubits is that the no-error probability for two-operand gates diminishes from $1 - 15p_2$ to $1 - 80p_2$, as expressed by equations 4 and 6 respectively.

A.1.2 Idle Errors. For qubits, the Kraus operators for amplitude damping are:

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1 - \lambda_1} \end{pmatrix} \quad \text{and} \quad K_1 = \begin{pmatrix} 0 & \sqrt{\lambda_1} \\ 0 & 0 \end{pmatrix} \quad (7)$$

For qutrits, the Kraus operator for amplitude damping can be modeled as [66, 67]:

$$K_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \sqrt{1 - \lambda_1} & 0 \\ 0 & 0 & \sqrt{1 - \lambda_2} \end{pmatrix}, K_1 = \begin{pmatrix} 0 & \sqrt{\lambda_1} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\text{and } K_2 = \begin{pmatrix} 0 & 0 & \sqrt{\lambda_2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (8)$$

As discussed in Section 6.1, these noise channels are incoherent (non-unitary), which means that the probability of each error occurring depends on the current state. Specifically, the probability of the K_i channel affecting the state $|\Psi\rangle$ is $\|K_i |\Psi\rangle\|^2$ [13].

A.2 Superconducting QC

We picked four noise models based on superconducting quantum computers that are expected in the next few years. These noise models comply with the generic noise model above and are thus parametrized by p_1 , p_2 , λ_1 , and λ_2 . The λ_m terms are given by [67]:

$$\lambda_m = 1 - e^{-m\Delta t/T_1} \quad (9)$$

where Δt is the duration of the idling and T_1 is associated with the lifetime of each qubit.

A.3 Trapped Ion $^{171}\text{Yb}^+$ QC

Based on calculations from experimental parameters for the trapped ion qutrit, we know the specific Kraus operator types for the error terms, which deviate slightly from those in the generic error model. The specific Kraus operator matrices are provided at our GitHub repository [33].

We chose three noise models: `TI_QUBIT`, `BARE_QUTRIT`, and `DRESSED_QUTRIT`. Both `TI_QUBIT` and `DRESSED_QUTRIT` take advantage of clock states and thus have very small idle errors. They both would be ideal candidates for a qudit. The `BARE_QUTRIT` will suffer more from idle errors as it is not strictly defined on clock states but will require less experimental resources to prepare. Idle errors are very small in magnitude and manifest as coherent phase errors rather than amplitude damping errors as modeled in Section 7.1.2. We also do not consider leakage errors. These errors could be handled for Yb^+ by treating each ion as a $d = 4$ qudit, regardless of whether we use it as a qubit or a qutrit.

REFERENCES

- [1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, pp. 195 EP –, Sep 2016.
- [2] I. Kassal, J. D. Whitfield, A. Perdomo-Ortiz, M.-H. Yung, and A. Aspuru-Guzik, "Simulating chemistry using quantum computers," *Annual review of physical chemistry*, vol. 62, pp. 185–207, 2011.
- [3] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, pp. 1484–1509, Oct. 1997.
- [4] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Annual ACM Symposium on Theory of Computing*, pp. 212–219, ACM, 1996.
- [5] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.
- [6] Y. Ding, A. Holmes, A. Javadi-Abhari, D. Franklin, M. Martonosi, and F. Chong, "Magic-state functional units: Mapping and scheduling multi-level distillation circuits for fault-tolerant quantum architectures," in *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 828–840, IEEE, 2018.
- [7] A. Javadi-Abhari, P. Gokhale, A. Holmes, D. Franklin, K. R. Brown, M. Martonosi, and F. T. Chong, "Optimized surface code communication in superconducting quantum computers," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-50 '17*, (New York, NY, USA), pp. 692–705, ACM, 2017.
- [8] G. G. Guerreschi and J. Park, "Two-step approach to scheduling quantum circuits," *Quantum Science and Technology*, vol. 3, p. 045003, Jul 2018.
- [9] A. Pavlidis and E. Floratos, "Arithmetic circuits for multilevel qudits based on quantum fourier transform," *arXiv preprint arXiv:1707.08834*, 2017.
- [10] J. Randall, S. Weidt, E. D. Standing, K. Lake, S. C. Webster, D. F. Murgia, T. Navickas, K. Roth, and W. K. Hensinger, "Efficient preparation and detection of microwave dressed-state qubits and qutrits with trapped ions," *Phys. Rev. A*, vol. 91, p. 012322, 01 2015.
- [11] J. Randall, A. M. Lawrence, S. C. Webster, S. Weidt, N. V. Vitanov, and W. K. Hensinger, "Generation of high-fidelity quantum control methods for multilevel systems," *Phys. Rev. A*, vol. 98, p. 043414, 10 2018.
- [12] "Cirq: A python framework for creating, editing, and invoking noisy intermediate scale quantum (NISQ) circuits." <https://github.com/quantumlib/Cirq>, 2018.
- [13] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. New York, NY, USA: Cambridge University Press, 10th ed., 2011.
- [14] J.-L. Brylinski and R. Brylinski, "Universal quantum gates," in *Mathematics of quantum computation*, pp. 117–134, Chapman and Hall/CRC, 2002.
- [15] Y.-M. Di and H.-R. Wei, "Elementary gates for ternary quantum logic circuit," *arXiv preprint arXiv:1105.5485*, 2011.
- [16] A. Muthukrishnan and C. R. Stroud, "Multivalued logic gates for quantum computation," *Phys. Rev. A*, vol. 62, p. 052309, Oct 2000.
- [17] A. B. Klimov, R. Guzmán, J. C. Retamal, and C. Saavedra, "Qutrit quantum computer with trapped ions," *Phys. Rev. A*, vol. 67, p. 062313, Jun 2003.
- [18] M. Kues, C. Reimer, P. Roztocki, L. R. Cortés, S. Sciara, B. Wetzl, Y. Zhang, A. Cino, S. T. Chu, B. E. Little, D. J. Moss, L. Caspani, J. Azaña, and R. Morandotti, "On-chip generation of high-dimensional entangled quantum states and their coherent control," *Nature*, vol. 546, pp. 622 EP –, 06 2017.
- [19] T. Bäckegaard, L. B. Kristensen, N. J. S. Loft, C. K. Andersen, D. Petrosyan, and N. T. Zinner, "Superconducting qutrit-qubit circuit: A toolbox for efficient quantum gates," *arXiv preprint arXiv:1802.04299*, 2018.
- [20] A. Fedorov, L. Steffen, M. Baur, M. P. da Silva, and A. Wallraff, "Implementation of a toffoli gate with superconducting circuits," *Nature*, vol. 481, pp. 170 EP –, Dec 2011.
- [21] A. D. Greentree, S. G. Schirmer, F. Green, L. C. L. Hollenberg, A. R. Hamilton, and R. G. Clark, "Maximizing the hilbert space for a finite number of distinguishable quantum states," *Phys. Rev. Lett.*, vol. 92, p. 097901, Mar 2004.
- [22] M. H. A. Khan and M. A. Perkowski, "Quantum ternary parallel adder/subtractor with partially-look-ahead carry," *J. Syst. Archit.*, vol. 53, pp. 453–464, July 2007.
- [23] Y. Fan, "Applications of multi-valued quantum algorithms," *arXiv preprint arXiv:0809.0932*, 2008.
- [24] H. Y. Li, C. W. Wu, W. T. Liu, P. X. Chen, and C. Z. Li, "Fast quantum search algorithm for databases of arbitrary size and its implementation in a cavity QED system," *Physics Letters A*, vol. 375, pp. 4249–4254, Nov. 2011.
- [25] Y. Wang and M. Perkowski, "Improved complexity of quantum oracles for ternary grover algorithm for graph coloring," in *2011 41st IEEE International Symposium on Multiple-Valued Logic*, pp. 294–301, May 2011.
- [26] S. S. Ivanov, H. S. Tonchev, and N. V. Vitanov, "Time-efficient implementation of quantum search with qudits," *Phys. Rev. A*, vol. 85, p. 062321, Jun 2012.
- [27] A. Bocharov, M. Roetteler, and K. M. Svore, "Factoring with qutrits: Shor's algorithm on ternary and metaplectic quantum architectures," *Phys. Rev. A*, vol. 96, p. 012306, Jul 2017.
- [28] C. Gidney, "Constructing large controlled nots," 2015.
- [29] Y. He, M.-X. Luo, E. Zhang, H.-K. Wang, and X.-F. Wang, "Decompositions of n-qubit Toffoli Gates with Linear Circuit Complexity," *International Journal of Theoretical Physics*, vol. 56, pp. 2350–2361, July 2017.
- [30] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, pp. 3457–3467, Nov 1995.
- [31] B. P. Lanyon, M. Barbieri, M. P. Almeida, T. Jennewein, T. C. Ralph, K. J. Resch, G. J. Pryde, J. L. O'Brien, A. Gilchrist, and A. G. White, "Simplifying quantum logic using higher-dimensional hilbert spaces," *Nature Physics*, vol. 5, pp. 134 EP –, 12 2008.
- [32] T. C. Ralph, K. J. Resch, and A. Gilchrist, "Efficient toffoli gates using qudits," *Phys. Rev. A*, vol. 75, p. 022313, Feb 2007.
- [33] "Code for asymptotic improvements to quantum circuits via qutrits." <https://github.com/epiqc/qutrits>, 2019.
- [34] F. Tacchino, C. Macchiavello, D. Gerace, and D. Bajoni, "An artificial neuron implemented on an actual quantum processor," *npj Quantum Information*, vol. 5, no. 1, p. 26, 2019.
- [35] F. Tacchino. Personal Communication.
- [36] T. G. Draper, "Addition on a quantum computer," *arXiv preprint quant-ph/0008033*, 2000.
- [37] C. Gidney, "Factoring with n+2 clean qubits and n-1 dirty qubits," *arXiv preprint arXiv:1706.07884*, 2017.
- [38] T. Häner, M. Roetteler, and K. M. Svore, "Factoring using $2n + 2$ qubits with toffoli based modular multiplication," *Quantum Info. Comput.*, vol. 17, pp. 673–684, June 2017.
- [39] D. G. Cory, M. D. Price, W. Maas, E. Knill, R. Laflamme, W. H. Zurek, T. F. Havel, and S. S. Somaroo, "Experimental quantum error correction," *Phys. Rev. Lett.*, vol. 81, pp. 2152–2155, Sep 1998.
- [40] E. Dennis, "Toward fault-tolerant quantum computation without concatenation," *Phys. Rev. A*, vol. 63, p. 052314, Apr 2001.
- [41] M. Otten and S. K. Gray, "Accounting for errors in quantum algorithms via individual error reduction," *npj Quantum Information*, vol. 5, no. 1, p. 11, 2019.
- [42] D. Miller, T. Holz, H. Kampermann, and D. Bruf, "Propagation of generalized pauli errors in qudit clifford circuits," *Physical Review A*, vol. 98, no. 5, p. 052316, 2018.
- [43] N. Khammassi, I. Ashraf, X. Fu, C. G. Almudever, and K. Bertels, "Qx: A high-performance quantum computer simulation platform," in *Proceedings of the Conference on Design, Automation & Test in Europe, DATE '17*, (3001 Leuven, Belgium, Belgium), pp. 464–469, European Design and Automation Association, 2017.
- [44] "Quantum devices and simulators." <https://www.research.ibm.com/ibm-q/technology/devices/>, 2018.
- [45] D. Venturelli, M. Do, E. Rieffel, and J. Frank, "Compiling quantum circuits to realistic hardware architectures using temporal planners," *Quantum Science and Technology*, vol. 3, p. 025004, feb 2018.
- [46] K. E. Booth, M. Do, J. C. Beck, E. Rieffel, D. Venturelli, and J. Frank, "Comparing and integrating constraint programming and temporal planning for quantum circuit compilation," in *Twenty-Eighth International Conference on Automated Planning and Scheduling*, 2018.
- [47] J. Biamonte and V. Bergholm, "Tensor networks in a nutshell," *arXiv preprint arXiv:1708.00006*, 2017.
- [48] T. A. Brun, "A simple model of quantum trajectories," *American Journal of Physics*, vol. 70, no. 7, pp. 719–737, 2002.
- [49] R. Schack and T. A. Brun, "A C++ library using quantum trajectories to solve quantum master equations," *Computer Physics Communications*, vol. 102, no. 1-3, pp. 210–228, 1997.

- [50] R. S. Smith, M. J. Curtis, and W. J. Zeng, "A practical quantum instruction set architecture," *arXiv preprint arXiv:1608.03355*, 2016.
- [51] J. Johansson, P. Nation, and F. Nori, "Qutip: An open-source python framework for the dynamics of open quantum systems," *Computer Physics Communications*, vol. 183, pp. 1760–1772, 8 2012.
- [52] J. Johansson, P. Nation, and F. Nori, "Qutip 2: A python framework for the dynamics of open quantum systems," *Computer Physics Communications*, vol. 184, pp. 1234–1240, 4 2013.
- [53] A. Y. Chernyavskiy, V. V. Voevodin, and V. V. Voevodin, "Parallel computational structure of noisy quantum circuits simulation," *Lobachevskii Journal of Mathematics*, vol. 39, pp. 494–502, May 2018.
- [54] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. 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.
- [55] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O'Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and J. M. Martinis, "Superconducting quantum circuits at the surface code threshold for fault tolerance," *Nature*, vol. 508, pp. 500 EP –, 04 2014.
- [56] E. Barnes, C. Arenz, A. Pitchford, and S. E. Economou, "Fast microwave-driven three-qubit gates for cavity-coupled superconducting qubits," *Phys. Rev. B*, vol. 96, p. 024504, Jul 2017.
- [57] M. Reagor, W. Pfaff, C. Axline, R. W. Heeres, N. Ofek, K. Sliwa, E. Holland, C. Wang, J. Blumoff, K. Chou, M. J. Hatridge, L. Frunzio, M. H. Devoret, L. Jiang, and R. J. Schoelkopf, "Quantum memory with millisecond coherence in circuit qed," *Phys. Rev. B*, vol. 94, p. 014506, Jul 2016.
- [58] N. Earnest, S. Chakram, Y. Lu, N. Irons, R. K. Naik, N. Leung, L. Ocola, D. A. Czaplewski, B. Baker, J. Lawrence, J. Koch, and D. I. Schuster, "Realization of a Λ system with metastable states of a capacitively shunted fluxonium," *Phys. Rev. Lett.*, vol. 120, p. 150504, Apr 2018.
- [59] S. M. Girvin, "Circuit qed: superconducting qubits coupled to microwave photons," *Quantum Machines: Measurement and Control of Engineered Quantum Systems*, p. 113, 2011.
- [60] K. R. Brown, J. Kim, and C. Monroe, "Co-designing a scalable quantum computer with trapped atomic ions," *npj Quantum Information*, vol. 2, p. 16034, 2016.
- [61] N. C. Brown and K. R. Brown, "Comparing zeeman qubits to hyperfine qubits in the context of the surface code: $^{174}\text{Yb}^+$ and $^{171}\text{Yb}^+$," *Phys. Rev. A*, vol. 97, p. 052301, May 2018.
- [62] R. K. Naik, N. Leung, S. Chakram, P. Groszkowski, Y. Lu, N. Earnest, D. C. McKay, J. Koch, and D. I. Schuster, "Random access quantum information processors using multimode circuit quantum electrodynamics," *Nature Communications*, vol. 8, no. 1, p. 1904, 2017.
- [63] M. Soeken, S. Frehse, R. Wille, and R. Drechsler, "Revkit: An open source toolkit for the design of reversible circuits," in *Reversible Computation* (A. De Vos and R. Wille, eds.), pp. 64–76, Springer Berlin Heidelberg, 2012.
- [64] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," in *Proceedings 2003. Design Automation Conference (IEEE Cat. No.03CH37451)*, pp. 318–323, June 2003.
- [65] V. Karimipour, A. Mani, and L. Memarzadeh, "Characterization of qutrit channels in terms of their covariance and symmetry properties," *Phys. Rev. A*, vol. 84, p. 012321, Jul 2011.
- [66] M. Grassl, L. Kong, Z. Wei, Z.-Q. Yin, and B. Zeng, "Quantum error-correcting codes for qutrit amplitude damping," *IEEE Transactions on Information Theory*, vol. 64, no. 6, pp. 4674–4685, 2018.
- [67] J. Ghosh, A. G. Fowler, J. M. Martinis, and M. R. Geller, "Understanding the effects of leakage in superconducting quantum-error-detection circuits," *Phys. Rev. A*, vol. 88, p. 062329, Dec 2013.