Theme Article: Top Picks

Extending the Frontier of Quantum Computers With Qutrits

Pranav Gokhale, Jonathan M. Baker, Casey Duckering, and Frederic T. Chong
University of Chicago

Natalie C. Brown
Georgia Institute of Technology

Kenneth R. Brown
Duke University

***Abstract*—We advocate for a fundamentally different way to perform quantum computation by using three-level qu*trits* instead of qubits. In particular, we substantially reduce the resource requirements of quantum computations by exploiting a third state for temporary variables (ancilla) in quantum circuits. 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 runtime decomposition of the Generalized Toffoli gate using no ancilla—an exponential improvement over the best qubit-only equivalent. Our approach features a 70x improvement in total two-qudit gate count over the qubit-only decomposition. This results in improvements for important algorithms for arithmetic and QRAM. Simulation results under realistic noise models indicate over 90% mean reliability (fidelity) for our circuit, versus under 30% for the qubit-only baseline. These results suggest that qutrits offer a promising path toward extending the frontier of quantum computers.**

Recent advances in both hardware and software for quantum computation have demonstrated significant progress toward practical outcomes. In the coming years, we expect quantum computing will have important applications in fields ranging from machine learning and optimization to drug discovery. While early research efforts focused on longer term systems employing full error correction to execute large instances of algorithms like Shor factoring and Grover search, recent work has focused on noisy

Digital Object Identifier 10.1109/MM.2020.2985976
Date of publication 16 April 2020; date of current version 22 May 2020.

intermediate scale quantum (NISQ) computation. 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 to extend the amount of useful computation possible. In this article, we consider another technique: quantum trits (qutrits).

> 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.

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[1] has emphasized the information compression advantages of qutrits. For example, N qubits can be expressed as $N/\log_2(3)$ 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.[2,3]

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 nondata workspace qubits known as ancilla. The lack of ancilla in the frontier zone is a costly constraint that generally leads to inefficient circuits. For this reason, typical circuits instead operate below the frontier zone, with many machine qubits used as ancilla. This article demonstrates that ancilla can be substituted with qutrits, enabling us to extend the ancilla-free frontier zone of quantum computation.

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 preserve valid probability distributions that sum to 1 and 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.

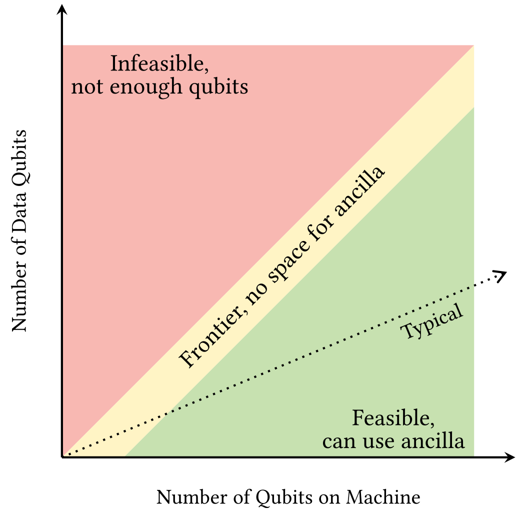

Figure 1. Frontier of what quantum hardware can execute is the yellow region adjacent to the 45° line. In this region, each machine qubit is a data qubit. Typical circuits rely on nondata ancilla qubits for workspace and therefore operate below the frontier.

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 *ancilla* bits initialized to $|0\rangle$.

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. As d increases, the number of opportunities for error—termed error channels—increases. Here, we focus on $d = 3$ which is sufficient to achieve 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} 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 $X_{02}|\psi\rangle = \gamma|0\rangle + \beta|1\rangle + \alpha|2\rangle$. There are two additional nontrivial operations on a single trit. They are X_{+1} and X_{-1} operations, which perform the addition/subtraction modulo 3.

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.

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 from input to output.

PRIOR WORK

Qudits

Qutrits, and more generally qudits, have been studied in past work both experimentally and theoretically. However, in the past work, qudits have conferred only an information compression advantage. For example, N qubits can be compressed to $N/\log_2(d)$ qudits, giving only a constant-factor advantage[1] at the cost of greater errors from operating qudits instead of qubits. Ultimately, the tradeoff between information compression and higher per-qudit errors has not been favorable in the past work. As such, the

Table 1. Asymptotic comparison of N-controlled gate decompositions. The total gate count for all circuits scales linearly (except for Barenco *et al.*,[6] which scales quadratically). Our construction uses qutrits to achieve logarithmic depth without ancilla. We benchmark our circuit construction against Gidney,[4] which is the asymptotically best ancilla-free qubit circuit.

	This Work	Gidney[4]	He *et al.*[5]	Barenco *et al.*[6]	Wang and Perkowski[7]	Lanyon *et al.*[8]
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

past research toward building practical quantum computers has focused on qubits.

This article 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.

Generalized Toffoli Gate

The Toffoli gate itself is a simple extension of the CNOT gate, but has two controls instead of one control. In a Toffoli gate, the NOT is applied if and only if *both* controls are $|1\rangle$. Similarly, a *Generalized Toffoli* gate has N controls and flips the target qubit if and only if all N control qubits are $|1\rangle$. 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 "Generalized Toffoli Gate" section.

Among prior work, Gidney,[4] He *et al.*,[5] and Barenco *et al.*[6] 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. 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 and operating far below the frontier. Nonetheless, in practice, most circuit implementations use these linear-ancilla constructions due to their small depths and gate counts.

As in our approach, circuit constructions from Wang and Perkowski,[7] and Lanyon *et al.*[8] have attempted to improve the ancilla-free Generalized Toffoli gate by using qudits. Wang and Perkowski[7] achieves a linear circuit depth but by operating each control as a qutrit. The Lanyon *et al.*[8] construction, which has been demonstrated experimentally, achieves linear circuit depths by operating the target as a $d = N$-level qudit.

Our circuit construction, presented in the "Generalized Toffoli Gate" section, 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[4] because it is the asymptotically best qubit circuit in the ancilla-free frontier zone. We label these two benchmarks as QUTRIT and QUBIT.

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.

Key Intuition

We develop the intuition for how qutrits can be useful by considering the example of constructing an AND gate. In the framework of quantum computing, which requires *reversibility*, AND is not permitted directly. For example, consider a two-input AND gate that outputs a 0. Given this output value, the inputs cannot be uniquely determined since 00, 01, and 10 all yield an AND output of 0. However, these operations can be made reversible by the addition of an extra temporary workspace bit initialized to 0. Using a single additional such *ancilla*, the AND operation can be computed reversibly via the well-known Toffoli gate (a double-controlled NOT) in reversible computation. While this approach works, it is expensive—its decomposition into hardware-implementable one- and two-input gates requires at least six controlled-NOT gates and several single qubit gates, as depicted in the top circuit of Figure 2.

However, if we break the qubit abstraction and allow occupation of a higher *qutrit* energy level, the cost of the Toffoli AND operation is greatly diminished. The bottom circuit in Figure 2 displays this decomposition using qutrits. The goal is to elevate the $|q_2 = 0\rangle$ ancilla qubit to $|1\rangle$ if and only if the top two control qubits are both $|1\rangle$. This is effectively an AND gate. First a $|1\rangle$-controlled $(+1 \bmod 3)$ 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 $(+1 \bmod 2)$ gate is applied to q_2. Therefore, q_2 is elevated to $|1\rangle$ 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 $(-1 \bmod 3)$ gate, which resets q_1. The key intuition is that temporary information can be stored in the qutrit $|2\rangle$ state, rather than requiring an external ancilla. This allows us to maximize the problem sizes that current-generation hardware can address.

Now, notice that this AND decomposition is actually also a Toffoli gate decomposition. In particular, suppose that the bottom qubit is now arbitrary, rather than initialized to $|0\rangle$. Then, the net effect of this circuit is to preserve the controls, while flipping the bottom (target) qubit, if and only if q_0 and q_1 were both $|1\rangle$. For this decomposition, since all inputs and outputs are binary qubits (we only occupy the $|2\rangle$ state temporarily during computation), we can insert this circuit constructions into any preexisting qubit-only circuits. Again, the key intuition in this decomposition is that the qutrit $|2\rangle$ state can be used instead of ancilla to store temporary information.

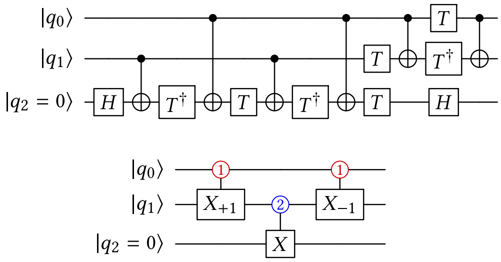

Figure 2. Toffoli AND via qubits (top) versus qutrits (bottom).

Generalized Toffoli Gate

We now present our circuit decomposition for the generalized Toffoli gate in Figure 3. 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[9] that requires six two-qutrit and seven single-qutrit physically implementable quantum gates.

Our circuit decomposition is most intuitively understood by treating the left half of 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 that 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 $|1\rangle$ 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.

APPLICATION TO ALGORITHMS

The Generalized Toffoli gate is an important primitive in a broad range of quantum algorithms. Here, we note two important applications of our circuit decomposition.

Arithmetic Circuits

The Generalized Toffoli is a key subcircuit in many arithmetic circuits such as constant addition, modular multiplication, and modular exponentiation. The circuit for computing a square root is also improved by a more efficient Generalized Toffoli gate. As shown by Gokhale,[10] the circuit for the initial approximation to $1/\sqrt{x}$ involves a sequence of standard Toffoli gates terminated by a large $O(n)$-width Generalized Toffoli gate. Our circuit construction is directly applicable to this terminal gate.

Quantum Machine Learning

A fundamental component of most algorithms for quantum machine learning is a quantum random access memory (QRAM). QRAM has the classically familiar property of mapping input index bits to output data bits. However, unlike classical RAM, QRAM also acts over superpositions of qubits. The initialization of QRAM is often the bottleneck for quantum machine learning algorithms—an expensive procedure for storing training data into QRAM can negate any potential quantum advantage.

However, the QRAM circuit is yet another application that can be improved with the qutrit-assisted Generalized Toffoli gate. In particular, the flip-flop QRAM is bottlenecked by the application of a wide Generalized Toffoli gate for each classical bitstring stored to the QRAM.[11] Thus, an efficient Generalized Toffoli gate reduces the cost of QRAM, relative to nonqutrit procedures.

SIMULATOR

To simulate our circuit constructions, we developed a qudit simulation library, built on Google's Cirq Python library. Cirq is a qubit-based quantum circuit library and includes a number of useful abstractions for quantum states, gates, circuits, and scheduling. Our software performs noise simulation, described in the following.

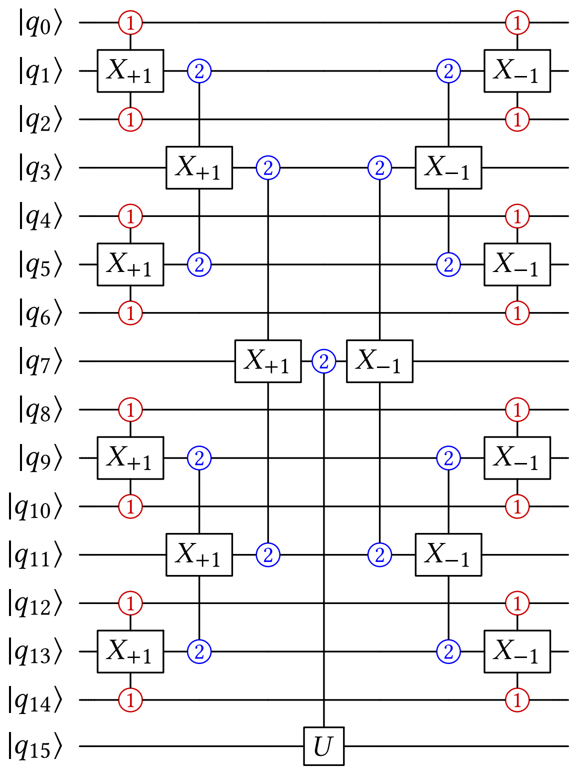

Figure 3. Our circuit decomposition for the Generalized Toffoli gate is shown for 15 controls and 1 target. The inputs and outputs are both qubits, but we allow occupation of the $|2\rangle$ qutrit state in between. The circuit has a tree structure and maintains the property that the root of each subtree 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 six two-input and seven single-input gates in our actual simulation, as based on the decomposition by Di and Wei.[9]

Noise Simulation

Our noise simulation procedure 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 `Moment`'s 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 or idle errors.

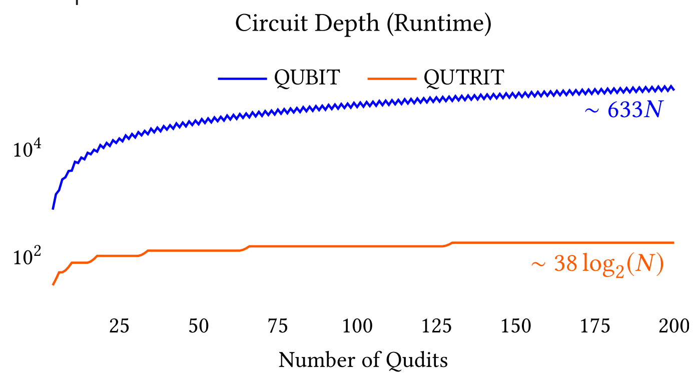

Figure 4. Exact circuit depths for all three benchmarked circuit constructions for the N-controlled Generalized Toffoli up to $N = 200$. QUBIT scales linearly in depth and is bested by QUTRIT's logarithmic depth.

Gate errors arise from the imperfect application of quantum gates. Two-qudit gates are noisier than single-qudit gates, so we apply different noise channels for the two. Idle errors arise from the continuous decoherence of a quantum system due to energy relaxation and interaction with the environment.

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.

The ultimate metric of interest is the mean *fidelity*, which captures the probability of overall successful execution. We do not consider state preparation and measurement (SPAM) errors, because our circuit constructions maintain binary input and output, only occupying the qutrit $|2\rangle$ states during intermediate computation. Therefore, the SPAM errors for our circuits are identical to those for conventional qubit circuits.

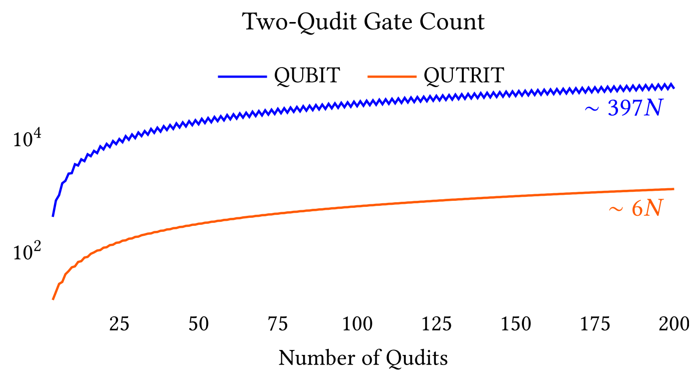

Figure 5. Exact two-qudit gate counts for the two benchmarked circuit constructions for the N-controlled Generalized Toffoli. Both plots scale linearly; however, the QUTRIT construction has a substantially lower linearity constant.

We chose noise models which represent realistic near-term machines. Our models encompassed both superconducting and trapped ion platforms. For superconducting technology, we simulated against four noise models: SC, SC+T1, SC+GATES, and SC+T1+GATES. For trapped ion technology, we simulated against three benchmarks: TI_QUBIT, BARE_QUTRIT, and DRESSED_QUTRIT.

RESULTS

Figure 4 plots the exact circuit depths for the QUTRIT versus QUBIT circuit construction. Note that the QUBIT construction is linear in depth, with a high linearity constant. This is significantly improved by our QUTRIT construction, which scales logarithmically in N and has a relatively small leading coefficient.

Figure 5 plots the total number of two-qudit gates for both circuit constructions. Our circuit construction is not asymptotically better in total gate count—both plots have linear scaling. However, as emphasized by the logarithmic vertical axis, the linearity constant for our qutrit circuit is $70\times$ smaller than for the equivalent ancilla-free qubit circuit.

We simulated these circuits under realistic noise models 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 6. All simulations are for the 14-input (13 controls, 1 target) Generalized Toffoli gate. We simulated both circuit constructions against each of our noise models (when applicable), yielding the 11 bars in the figure.

DISCUSSION AND FUTURE WORK

Figure 6 demonstrates that our QUTRIT construction (orange bars) can significantly

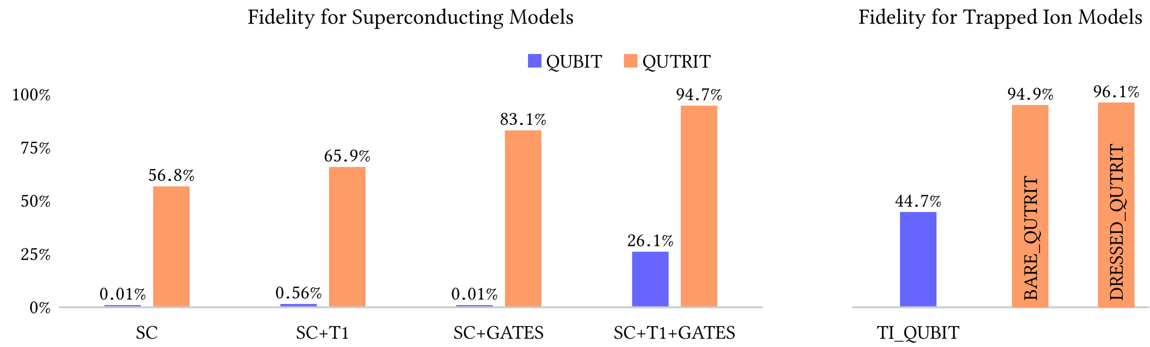

Figure 6. Circuit simulation results for all possible pairs of circuit constructions and noise models. Each bar represents 1000+ trials, so the error bars are all $2\sigma < 0.1\%$. Our QUTRIT construction significantly outperforms the QUBIT construction.

outperform the ancilla-free QUBIT benchmark (blue bars) in fidelity (success probability) by more than 10 000×.

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, 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.

Clever use of qutrits offers a path to more sophisticated quantum computation today, without needing to wait for better hardware. We are optimistic that continued hardware–software codesign may further extend the frontier of quantum computers.

Our qutrit-assisted Generalized Toffoli gate has already attracted interest from both device physics and algorithms communities. To this end, major quantum software packages like Cirq are now compatible with qutrit (and qudit) simulations. We have also been working with hardware groups to experimentally implement the ideas presented here. One promising direction is to use OpenPulse, an open standard for pulse-level quantum control, to experimentally demonstrate a generalized Toffoli gate. We also envision other advantages to higher radix quantum computing. For example, the information-compression advantage of qudits may be particularly well suited to the NISQ hardware, where device connectivity—and therefore diameter—is a bottleneck. Compressing a qubit computation via qudits would allow us to reduce the graph diameter.

The results presented in this article are applicable to quantum computing in the near term on machines that are expected within the next five years. The net result of this article is to extend the frontier of what is computable by quantum hardware, and hence to accelerate the timeline for practical quantum computing. 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 circuit simulator coupled with realistic noise models, our circuits are more reliable than qubit-only equivalents. In sum, clever use of qutrits offers a path to more sophisticated quantum computation today, without needing to wait for better hardware. We are optimistic that continued hardware–software codesign may further extend the frontier of quantum computers.

ACKNOWLEDGMENTS

We would like to thank Michel Devoret and Steven Girvin for suggesting to investigate qutrits. We also acknowledge David Schuster for helpful discussion on superconducting qutrits. This work was supported in part by EPiQC, an NSF Expedition in Computing, under Grant CCF-1730449/1832377; in part by STAQ under Grant NSF Phy-1818914; and in part by DOE Grants DE-SC0020289 and DE-SC0020331. The work of Pranav Gokhale was supported by the Department of Defense through the National Defense Science and Engineering Graduate Fellowship Program.

REFERENCES

1. A. Pavlidis and E. Floratos, "Arithmetic circuits for multilevel qudits based on quantum Fourier transform," 2017*, arXiv:1707.08834*.
2. J. Randall *et al.*, "Efficient preparation and detection of microwave dressed-state qubits and qutrits with trapped ions," *Phys. Rev. A*, vol. 91, 2015, Art. no. 012322.
3. 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, 10 2018, Art. no. 043414.
4. C. Gidney, "Constructing large controlled nots," 2015.
5. Y. He, M.-X. Luo, E. Zhang, H.-K. Wang, and X.-F. Wang, "Decompositions of n-qubit toffoli gates with linear circuit complexity," *Int. J. Theor. Phys.*, vol. 56, pp. 2350–2361, Jul. 2017.
6. A. Barenco *et al.*, "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, pp. 3457–3467, Nov. 1995.
7. Y. Wang and M. Perkowski, "Improved complexity of quantum oracles for ternary grover algorithm for graph coloring," in *Proc. 41st IEEE Int. Symp. Multiple-Valued Logic*, May 2011, pp. 294–301.
8. B. P. Lanyon *et al.*, "Simplifying quantum logic using higher-dimensional Hilbert spaces," *Nature Phys.*, vol. 5, pp. 134–140,, 2009.
9. Y.-M. Di and H.-R. Wei, "Elementary gates for ternary quantum logic circuit," 2011, *arXiv:1105.5485*.
10. P. Gokhale, "Implementation of square root function using quantum circuits," *Undergraduate Awards*, 2014.
11. D. K. Park, F. Petruccione, and J.-K. K. Rhee, "Circuit-based quantum random access memory for classical data," *Sci. Rep.*, vol. 9, no. 1, 2019, Art. no. 3949.

Pranav Gokhale is currently working toward the Ph.D. degree with the University of Chicago. His research focuses on breaking the abstraction barrier between quantum hardware and software. He is the founder of Super.tech. Contact him at pranavgokhale@uchicago.edu.

Jonathan M. Baker is currently working toward the Ph.D. degree with the University of Chicago. His research is primarily focused on vertical integration of the quantum computing hardware–software stack. Contact him at jmbaker@uchicago.edu.

Casey Duckering is currently working toward the Ph.D. degree with the University of Chicago, aiming to efficiently bring together quantum algorithms with their physical implementation on quantum computers. Contact him at cduck@uchicago.edu.

Frederic T. Chong is the Seymour Goodman Professor with the Department of Computer Science, University of Chicago. He is also Lead Principal Investigator for the EPiQC Project (Enabling Practical-scale Quantum Computing), an NSF Expedition in Computing. Contact him at chong@cs.uchicago.edu.

Natalie C. Brown is currently working toward the Ph.D. degree with Georgia Institute of Technology. Her research focuses on leakage error correction and mitigation in topological surface codes. Contact her at natalie.c.brown@duke.edu.

Kenneth R. Brown is an Associate Professor of electrical and computer engineering with Duke University and the Director of the NSF Software Enabled Architectures for Quantum co-design (STAQ) project developing applications, software, and hardware for ion trap quantum computers. Contact him at kenneth.r.brown@duke.edu.