Compilation of Qubit Circuits to Optimized Qutrit Circuits

RITVIK SHARMA, Stanford University, USA
SARA ACHOUR, Stanford University, USA

Quantum computers are a revolutionary class of computational platforms that are capable of solving computationally hard problems. However, today's quantum hardware is subject to noise and decoherence issues that together limit the scale and complexity of the quantum circuits that can be implemented. Recently, practitioners have developed qutrit-based quantum hardware platforms that compute over $|0\rangle$, $|1\rangle$, and $|2\rangle$ states, and have presented circuit depth reduction techniques using qutrits' higher energy $|2\rangle$ states to temporarily store information. However, thus far, such quantum circuits that use higher order states for temporary storage need to be manually crafted by hardware designers. We present Dare, an optimizing compiler for qutrit circuits that implement qubit computations. Dare deploys a qutrit circuit decomposition algorithm and a rewrite engine to construct and optimize qutrit circuits. We evaluate Dare against hand-optimized qutrit circuits and qubit circuits, and find Dare delivers up to 65% depth improvement over manual qutrit implementations, and 43-75% depth improvement over qubit circuits. We also perform a fidelity analysis and find Dare-optimized qutrit circuits deliver up to 8.9× higher fidelity circuits than their manually implemented counterparts.

CCS Concepts: • **Computer systems organization → Quantum computing**; • **Hardware → Quantum technologies**.

Additional Key Words and Phrases: Quantum computing, Qutrits, Synthesis, Rewriting Tools

ACM Reference Format:
Ritvik Sharma and Sara Achour. 2024. Compilation of Qubit Circuits to Optimized Qutrit Circuits. *Proc. ACM Program. Lang.* 8, PLDI, Article 158 (June 2024), 24 pages. https://doi.org/10.1145/3656388

1 INTRODUCTION

Quantum computers are a revolutionary class of computational platforms with applications in combinatorial and global optimization, machine learning, and other domains involving computationally hard problems. Typically, quantum computers operate on qubits, quantum information elements that can occupy superpositions of the basis $|0\rangle$ and $|1\rangle$ states. Therefore, they only utilize the two lower energy states of the quantum device. However, quantum devices have a multitude of higher energy states that can encode information and can, therefore, theoretically represent a much larger state space. These higher order $|2\rangle ... |d-1\rangle$ states can be used to implement qubit computations at substantially reduced depth and with significantly fewer hardware resources, or can more naturally implement new quantum computational workloads that are more naturally expressed in their larger basis [9, 18].

1.1 Qutrits and Other Higher-Order Quantum Systems

In recent years, researchers have demonstrated quantum computers that use higher-order quantum states are practically implementable and can be achieved through the use of new device technologies, such as photonic circuits [11, 20] and nitrogen-vacancy [6], as well as modifying the control logic of more commonly seen quantum hardware technologies like superconducting [9, 27] and trapped-ion

Authors' addresses: Ritvik Sharma, Stanford University, Stanford, USA, rsharma3@stanford.edu; Sara Achour, Stanford University, Stanford, USA, sachour@stanford.edu.

ACM 2475-1421/2024/6-ART158
https://doi.org/10.1145/3656388

systems [10, 25]. In these hardware realizations, the higher-order states are comparatively less stable than the $|0\rangle$ and $|1\rangle$ states and therefore remain coherent for a shorter period of time. For this reason, qutrit-based systems, which operate on the $|0\rangle$, $|1\rangle$, and $|2\rangle$ basis states, are especially promising higher-order quantum systems, as they are comparatively more stable.

Thus far, practitioners have been highly successful in using qutrits to construct more efficient circuits. In particular, researchers have achieved significant depth reductions with qutrit-based circuits by internally using the qutrit's higher-order $|2\rangle$ state to temporarily store quantum information [9]. This depth reduction approach naturally parallels ancilla circuits, which also use temporary storage to reduce circuit depth [7, 8]. However with qutrits, unlike with ancilla circuits, additional quantum devices are not expended to achieve these depth reductions. So far, these reduced-depth qutrit circuit implementations are hand-crafted, so much of the effort has been focused on optimizing basic gates and small circuits that can be tractably constructed by hand [9]. To our knowledge, the process of mapping a qubit circuit to an efficient qutrit-based implementation is done completely manually, and as a result, these optimized circuits are very small and involve anywhere between 3-6 qubits/qutrits.

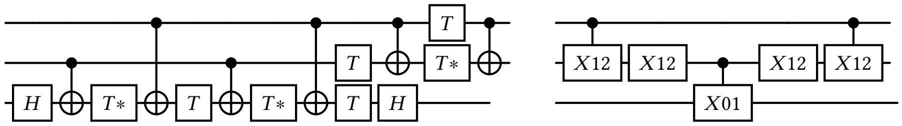
Overall, there has been an interest in exploring quantum devices that implement more than two states [1, 9, 16], including qutrits (d=3) [17, 18] and ququarts (d=4) [19]. These higher-order systems have been demonstrated to be practically implementable and have proof-of-concept implementations on quantum hardware. Hardware researchers have demonstrated qutrit (3-state), ququarts (4-state), and ququants (5-state) are physically implementable in NISQ hardware and have also architected novel devices that support higher-order states [27].

1.2 Optimized Qutrit Construction with Dare

We present the Decomposition and Rewrite Engine (Dare), an optimizing compiler that automatically translates qubit-based quantum circuits into optimized qutrit-based circuits that use the higher order $|2\rangle$ state to temporarily store information. We make the following contributions:

- **Qubit-Equality and Circuit Transformations.** Dare produces qutrit circuits that are *qubit-equal* to the original qubit-based circuit. Two circuits are qubit-equal if they implement the same computation when qubits are provided as circuit inputs. We formalize qubit equality and introduce nine circuit transformations that maintain this relaxed definition of equality.
- **Dare Circuit Compiler.** The Dare compiler orchestrates circuit lifting and decomposition procedures to translate a qubit-based circuit into a qutrit circuit that effectively uses the $|2\rangle$ state. Dare then aggressively rewrites the unoptimized qutrit circuit using rewrites built on above-mentioned circuit transforms. It deploys circuit cutting and stitching to compile large qubit-based circuits.
- **Qutrit Circuit Decomposition Algorithm.** Dare deploys a novel unitary matrix decomposition algorithm for qutrit circuits, adapted from the generalized Quantum Shannon decomposition (QSD) algorithm [4]. To our knowledge, Dare provides the first practical implementation of the QSD algorithm for qutrits and deploys optimizations to reduce the generated circuit's resource footprint.

Evaluation. We evaluate Dare on 20 quantum circuit benchmarks with 3-30 qubits with 6-168 2-qubit gates, and circuit depths ranging from 11-148. The Dare-optimized implementations deliver 43-75% depth reductions and use up to 79% fewer total gates than qubit-based implementations. Dare-generated circuits improve upon hand-optimized qutrit implementations, delivering depth improvements of up to 65% and total gate count improvements of up to 75%. We also compare the fidelity of manually optimized and Dare-optimized circuits and find that Dare can achieve up to 8.9× better average fidelity metric (AFM) than hand-optimized qutrit-based circuits. We perform a fidelity study where we identify what hardware advances need to be made for qutrit circuit fidelity to surpass qubit circuit fidelity. Given that the development of qutrit devices is still in its nascency, these projections help encapsulate the benefits of investing in quantum hardware that supports higher-order states.



(a) Logical Qubit-Based Toffoli Gate

(b) Logical Qutrit-Based Toffoli Gate

Fig. 1. Qubit-Based and Qutrit-Based Toffoli Circuits

2 MOTIVATION AND DARE OVERVIEW

In qubit-based quantum computing, quantum devices are in a superposition of the $|0\rangle$ and $|1\rangle$ basis states. On the other hand, a *qutrit*'s state can be described as $\alpha|0\rangle+\beta|1\rangle+\gamma|2\rangle$ where $|\alpha|^2+|\beta|^2+|\gamma|^2=1$. Thus, a qutrit will effectively act as a qubit when the $|2\rangle$ state has an amplitude (γ) of 0, or equivalently has no energy in the $|2\rangle$ state. Dare targets *qutrit* circuits which utilize all qutrit states during the course of the circuit but have a qubit interface. Furthermore, Dare uses a standard qutrit basis [4] and in particular, uses self-inverting swap (X_{ij}) gates that can switch the $|i\rangle$ and $|j\rangle$ states of a qutrit.

Consider the case where we want to efficiently implement a Toffoli gate, a 3-qubit gate that is controlled over the first two qubits and inverts the state of the third qubit when enabled. The generalized Toffoli gate is an important primitive used in many quantum algorithms and has been the focus of extensive past optimization work. Figure 1a presents the classical qubit circuit implementation of the Toffoli gate; it has a depth of eleven gates and contains six two-qubit gates and nine one-qubit gates.

Now, with qutrits, we can get a logical Toffoli gate implementation (Figure 1b). This circuit uses $|2\rangle$ state to temporarily store information internally within the circuit. In this example, all 2-qutrit gates are controlled on state $|1\rangle$. The controlled gates in this circuit are the (controlled) X_{01},X_{12} gates. These gates swap $|0\rangle-|1\rangle$, and the $|1\rangle-|2\rangle$ states with each other respectively, when the control qutrit is in state $|1\rangle$. We observe the qutrit-based implementation is significantly more resource-efficient than the qubit-based implementation, using only two one-qutrit gates and three two-qutrit gates and attaining a depth of five gates. The circuit uses the $|2\rangle$ state to store information temporarily to achieve these reductions. Note that the logical qutrit and qubit circuit both implement the same operation over the $|0\rangle$ and $|1\rangle$ states. Both circuits enforce a *qubit input/output interface* and require circuit inputs to be qubits and produce qubits as outputs. This property enables practitioners to transparently replace the qubit implementation of the Toffoli gate with the qutrit implementation. The Dare compiler works with this notion of equivalence, termed qubit-equality, and produces qutrit circuits that implement qubit interfaces.

2.1 Compilation with Dare

The Dare compiler (Figure 2) automatically compiles qubit-based circuits to efficient qutrit circuits that internally use the $|2\rangle$ state for temporary storage. The qutrit circuit implements a qubit interface: It accepts qubits as inputs and produces qubits as outputs. Internally, it operates in three stages:

- **Lifting and Decomposition.** The qubit circuit is lifted to an equivalent qutrit-based logical circuit, translated to a $3^n \times 3^n$ unitary matrix, and then decomposed using our novel qutrit unitary decomposition algorithm implementation. The produced qutrit effectively uses the $|2\rangle$ state to store information and exactly implements the unitary of the lifted circuit.
- **Optimization.** Dare then rewrites the unoptimized, decomposed circuit to reduce the depth and gate count. Dare's rewrites change the underlying unitary implemented by the circuit but still preserve the original qubit computation over the $|0\rangle$ and $|1\rangle$ states. We refer to this form of equality

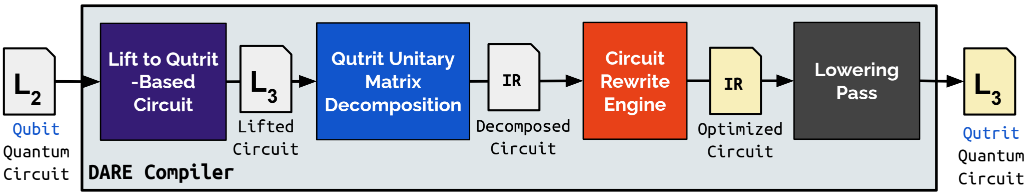


Fig. 2. Dare compilation flow. L_2 circuits are logical qubit quantum circuits, L_3 circuits are logical qutrit quantum, **IR** circuits are expressed in Dare IR. ■ circuits (before rewrite engine) implement exact same unitary, 🟨 circuits implement a different unitary but preserve the behavior of the original qubit circuit.

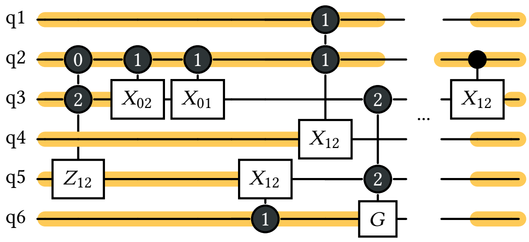


Fig. 3. Dare-IR Example Circuit. Wires with qubit basis states are highlighted yellow. Control states are written inside control. Z_{12} gate only changes the phase of $|2\rangle$ state.

$$\begin{bmatrix} 1 & 0 \\ 0 & e^{\pi/4} \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{\pi/4} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Fig. 4. Lifted T gate (top) and H gate.

as *qubit-equality.* A qubit-equal, qutrit-based circuit implements the same behavior as the original qubit-based quantum circuit when qubits are supplied as circuit inputs.

- **Lowering to Basis Gates.** Dare internally works with quantum circuit IR that supports multi-controlled gates with controls over $|0\rangle$, $|2\rangle$ quantum states. However, these circuits require arbitrary qutrit control states. Therefore, Dare implements a lowering step that translates qutrit circuits, written in the Dare IR, to a logical circuit comprised of qutrit basis gates with the control state of $|1\rangle$ state only (this control state is also generally used for qubit circuits and is well supported by most tools).

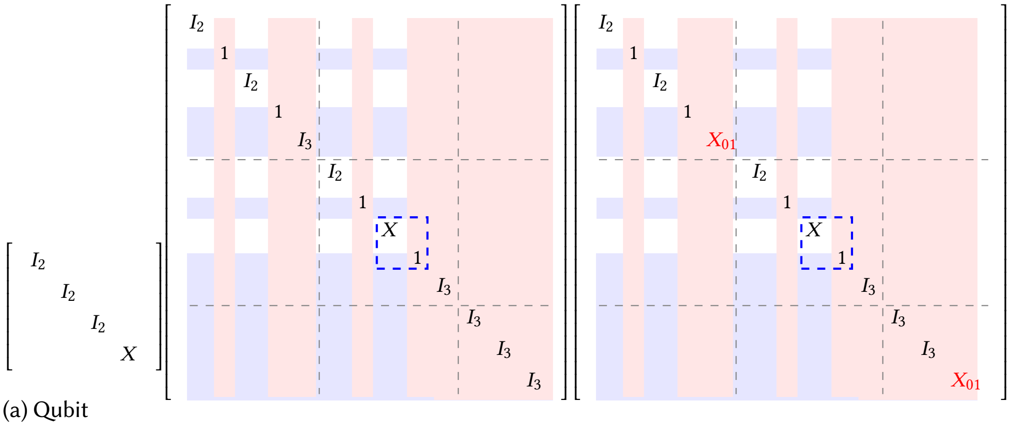
2.2 The Qutrit Circuit IR

The Dare compiler works with the Dare IR, an intermediate representation for qutrit circuits that support multi-controlled gates with controls over the $|0\rangle$, $|1\rangle$, and $|2\rangle$ states. Circuits expressed in the Dare IR can be automatically lowered to a logical circuit. Figure 3 presents an example quantum circuit, in Dare IR. The Dare IR has two key differences from traditional quantum circuit representations:

- **Multicontrolled Gates.** The Dare IR supports multicontrolled gates with controls that operate on $|0\rangle$,$|1\rangle$ or $|2\rangle$ state of the qutrit. The quantum state each control operates on is written inside the control. For example, in circuit shown in Figure 3, Z_{12} is applied when q_2,q_3 are in states $|0\rangle$,$|2\rangle$.
- **Wire Annotations.** The Dare IR automatically annotates wires depending on the basis states in use. Qubit wires are shaded yellow and only use the $|0\rangle$,$|1\rangle$ basis states of the qutrit and always have zero energy in the $|2\rangle$ state ($\gamma = 0$). Qutrit wires use all three states. These annotations are statically derived from the circuit structure using the analysis described in Section 5.

2.3 Step 1: Lifting Qubit Circuits to Qutrit Circuits

The Dare compiler first lifts the input qubit circuit to a qutrit circuit by extending each qubit to a qutrit and extending each qubit gate to implement the identity over the $|2\rangle$ state. The lifted qutrit circuit



(a) Qubit Unitary

(b) Original Qutrit Unitary

(c) Optimized Qutrit Unitary

Fig. 5. Toffoli gate qubit unitary and qutrit circuit unitaries before and after optimization. 🟥 columns map $|2\rangle$ input states to output states, 🟦 entries map inputs to $|2\rangle$ output states, white entries involve $|0\rangle$ and $|1\rangle$ basis states only, differing submatrices shown in red text. Dashed lines partition unitaries into 9×9 sub-blocks. Blue dashed region applies the qutrit X_{01} gate on the target. All blank entries are zeroes.

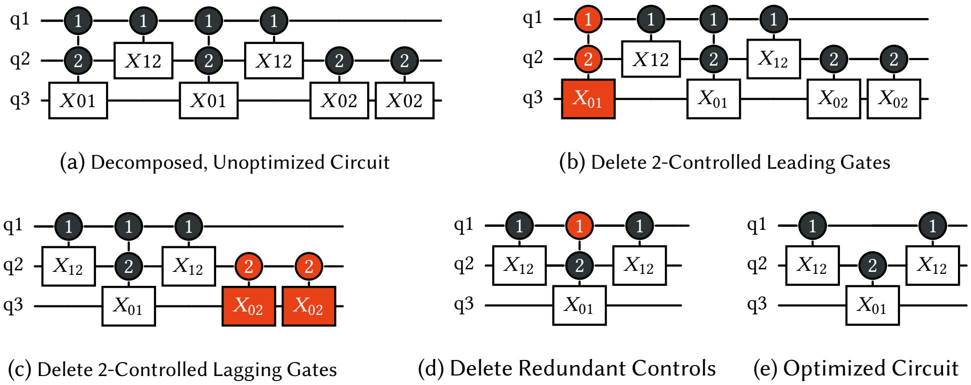
implements the original qubit computation over the $|0\rangle$ and $|1\rangle$ qutrit states and does not modify the $|2\rangle$ qutrit state. Figure 4 presents the original 2×2 and the lifted 3×3 unitaries for the uncontrolled T and H gates in the Toffoli circuit. This lifted unitary implements the same computation as the original over the $|0\rangle$ and $|1\rangle$ states (shown by white-shaded cells). Thus circuit lifting will maintain the original qubit computation and its outputs will only contain energy in the $|0\rangle$ and $|1\rangle$ states (and act as qubits).

Figure 5a presents the original unitary matrix for the qubit-based Toffoli, and Figure 5b presents the unitary matrix of the lifted, qutrit-based Toffoli. The unitary matrix of the circuit maps the quantum circuit input states to the quantum circuit output states. Both unitaries implement the same behavior over the $|0\rangle$ and $|1\rangle$ states (white cells), and apply the identity over the $|2\rangle$ state (red, blue cells). Applying only qubit states at the input interface eliminates all unitary matrix entries that map input states with energy in the $|2\rangle$ basis state to the gate output (red columns). Entries that map input states with no energy in $|2\rangle$ basis state to output states with energy in the $|2\rangle$ basis state (blue rows) are zero.

2.4 Step 2: Qutrit Unitary Matrix Decomposition

After the qubit circuit is lifted to a qutrit circuit, and Dare generates the $3^n \times 3^n$ qutrit unitary, we utilize qutrit unitary matrix decomposition to generate a qutrit circuit implementation that uses the $|2\rangle$ state. This qutrit circuit is expressed in Dare-IR. Dare provides the first practical implementation of the qutrit decomposition algorithm presented in [4, 14] and deploys optimizations to reduce the complexity of the decomposed circuit for certain block matrix patterns. The full description of the decomposition algorithm implementation and the associated optimizations are presented in Section 6.

In the Toffoli example, the lifted qutrit-based toffoli unitary from Figure 5b is decomposed into a qutrit circuit using Dare's qutrit decomposition algorithm. Figure 6a presents the qutrit-based Toffoli circuit produced by the decomposition algorithm. We observe that the decomposed qutrit circuit has a radically different structure than the original Toffoli circuit implementation presented in Figure 1a, and actively uses the $|2\rangle$ state in both the gate controls and operations. Therefore, the decomposition step is critical to ensuring the qutrit circuit internally uses the $|2\rangle$ state. The unitary matrix of the qutrit produced by the decomposition algorithm circuit exactly matches the starting unitary.



(a) Decomposed, Unoptimized Circuit

(b) Delete 2-Controlled Leading Gates

(c) Delete 2-Controlled Lagging Gates

(d) Delete Redundant Controls

(e) Optimized Circuit

Fig. 6. Toffoli circuit decomposition and rewriting steps

2.5 Step 3: Rewrite-Based Circuit Optimization

Dare employs an aggressive rewrite engine that uses the fact that certain gates and wires in the unoptimized qutrit circuit only operate on a subset of $|0\rangle$, $|1\rangle$, and $|2\rangle$ basis states. Figure 6e presents the Dare-optimized Toffoli circuit derived from the unoptimized, decomposed Toffoli circuit from Figure 6a. Dare's rewrite-based optimizer is able to eliminate three gates and one control from the unoptimized Toffoli circuit, halving the number of gates and reducing the depth from 6 to 3 gates. Dare applies the following rewrites to the unoptimized circuit to yield the optimized circuit:

- *Deleting $|2\rangle$-Controlled Gates.* (Figures 6b-6c) The rewrite engine safely deletes the leading controlled `X01` and the lagging controlled `X02` gates since the controls of these gates are never satisfied. They can only be applied when the control qubit `q2` is in $|2\rangle$ state. However, `q2` at the circuit input/output interface is only in states $|0\rangle$ and $|1\rangle$ (qubit basis). Hence, these controls cannot be satisfied.
- *Deleting Redundant Controls.*(Figure 6d) The rewrite engine also safely deletes the $|1\rangle$ control on controlled `X01` gate. Because the `q2` qutrit is exclusively in the $|0\rangle$ and $|1\rangle$ states at the input interface, the $|2\rangle$ control over `q2` on the `X01` gate is only satisfied if the preceding `X12` gate is applied. Therefore, `X01` gate's $|1\rangle$ control on `q1` is already implicitly enforced by the preceding `X12` gate.

2.5.1 *Qubit-Equality.*

Figure 5 compares the unitary matrix of the unoptimized circuit (Figure 6a) with the unitary matrix of the rewritten circuit (Figure 6e). While the applied rewrites safely remove gates and controls that will never be exercised, the rewrites change the underlying unitary implemented by the circuit and, therefore, implement a different quantum computation. We define a new global circuit property called *qubit-equality*, where two qutrit circuits are *qubit-equal* if the same computation is implemented when qubits are applied to the circuits' input interface. Dare's rewrite system produces a qutrit circuit that is qubit-equal to the original, unoptimized circuit. This condition is sufficient for ensuring correctness since Dare-generated circuits are always exercised with qubit inputs.

We next provide an intuitive explanation for why the original and transformed unitaries presented in Figure 5 are qubit-equal. Recall the quantum circuit's unitary captures the mapping of the superposition of input quantum states (columns) to output quantum states (rows). We observe that while the unitaries of the two Toffoli implementations differ, three key properties hold:

- *The computation over the $|0\rangle$ and $|1\rangle$ states is preserved.* All entries where the input and output states for all qutrits are in $|0\rangle$ and $|1\rangle$ states (white background) are the same in the original and optimized unitary matrix.

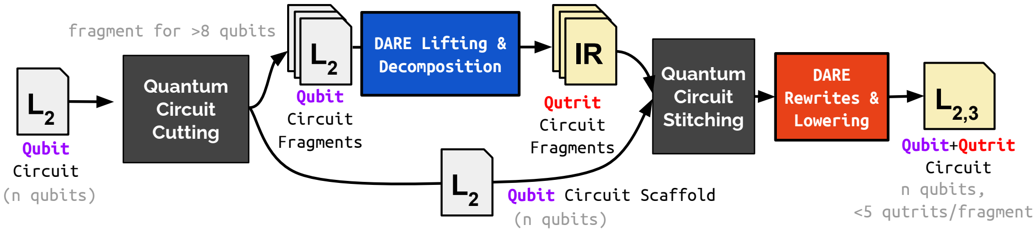


Fig. 7. Scalable qubit circuit compilation with Dare+circuit cutting and stitching optimization

- *The* $|1\rangle$ *and* $|0\rangle$ *input states never affect the* $|2\rangle$ *output state.* All matrix entries that may transfer energy from states $|0\rangle$ and $|1\rangle$ to state $|2\rangle$ (blue cells) are zero in the optimized matrix. Therefore, all output states that involve the $|2\rangle$ basis will resolve to zero for qubit inputs.
- *All non-matching unitary matrix entries require circuit inputs occupy the* $|2\rangle$ *state.* All entries that differ in the unoptimized and optimized unitaries require one of the quantum inputs to be in the $|2\rangle$ state (red columns). Because the quantum circuit is always applied over qubit inputs, the $|2\rangle$ state is always zero, and these fields are zeroed out and have no effect on the distribution of output states.

These properties ensure that the original Toffoli computation on qubit states is implemented by the qutrit circuit, provided input qutrits operate as qubits (use only the $|0\rangle$,$|1\rangle$ states). Put simply, all Toffoli unitary entries highlighted red are allowed to differ – this flexible definition of equality enables Dare to employ rewrites that delete gates/controls and change the semantics of the circuit.

2.5.2 The Dare Optimizer and Correctness. Dare's rewrite engine (Section 4) applies rewrites (Section 4) on matching gate patterns until the quantum circuit cannot be further optimized. To ensure the rewrites preserve the overall behavior of the circuit, we define each rewrite as a sequence of *circuit transforms* (Section 3.4) to optimize the circuit. Each circuit transform is formalized and paired with a proof sketch that demonstrates the transform does not change the behavior of the circuit. After optimization, Dare checks that the unitary of the rewritten circuit is qubit-equal to the original unoptimized circuit's unitary. Section 3.3 presents a rigorous definition of qubit equivalence.

2.6 Step 4: Lowering Dare-IR to a Qutrit Logical Circuit

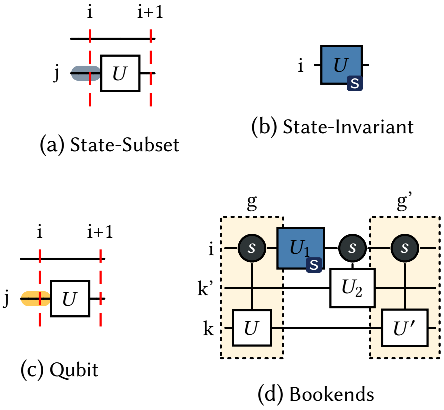
Qutrit circuits expressed in Dare-IR contain multi-controlled single qutrit gates. Dare's lowering pass translates these multi-controlled gates by first inserting swap (X_{01}, X_{12}) gates so that all controls operate on $|1\rangle$ state. This ensures that all lowered circuits use the same control conditions as qubit circuits. Dare then lowers gates to a basis gate set which contains single qutrit rotation gates $[R_{xij}(\theta)$,$R_{yij}(\theta)$,$R_{zij}(\theta)]$ which rotate the qutrit by θ on the x, y, z axis respectively for $i,j = [01,12,02]$ basis. They generalize single qubit rotation gates [23]. Dare's gate set also contains two-qutrit controlled-not gates $CNOT_{ij}$ which are controlled-$X_{i,j}$ gates that invert i,j states with each other (where $i,j = [01,12,02]$). This gateset was previously proposed and used in [4, 9]. For the Toffoli gate example, the qutrit-based circuit shown in 1b is derived from the optimized Toffoli circuit in Figure 6e, expressed in the Dare-IR, during Dare's lowering pass. The lowering pass replaces the control over $|2\rangle$ state with a control over $|1\rangle$ state and inserts **X12** gates on either side – the **X12** gate is self-inverting.

2.7 Scalable Quantum Circuit Compilation with Dare

Dare's basic compilation flow (Figure 2) contains a decomposition step which operates on a $3^n \times 3^n$ sized unitary matrix for a n-qutrit circuit. The runtime of this decomposition algorithm grows exponentially as the number of qutrits increases and becomes intractable beyond eight qutrits. Therefore,

element	description
$s \in S \subseteq \{0,1,2\}$	set of qutrit basis states.
n	number of qutrits in circuit
z,U	3×3 matrix.
$U^{\dagger}$	inverse of U.
U_S	3×3 state invariant unitary matrix.
$C(s)=\lvert s\rangle\langle s\rvert$	control on basis state s.
$I(3)$	a 3×3 identity matrix
$Z_{i,j}=[z_i,...,z_j]$	operators applied to qutrits i-j
$G_i=Gate(Z,k)$	a gate with target on qutrit k
G_i	ith gate of circuit
$\pi(Z)$	outer product of 3x3 matrices in Z.
Q	quantum circuit
$Book(Z,U,k)$	Bookend gate, applies U on k
$Int(Z',U'',k')$	Interior gate, applies U'' on k'

Fig. 8. Notation quick-reference



(a) State-Subset

(b) State-Invariant

(c) Qubit

(d) Bookends

Fig. 9. Circuit conditions.

to ensure Dare can scale to larger circuits, Dare uses a cut-and-stitch compilation flow inspired by the Quest synthesizer [24] when the input circuit exceeds eight qubits. This allows Dare to divide the circuit into fragments which take a much smaller time to decompose. Each circuit fragment is itself a qubit circuit and thus naturally maintains a qubit interface after decomposition. We highlight the benefits of this step to compile larger circuits using the compiler runtime for circuits in Section 7.1.5.
Algorithm. Dare first cuts the circuit, extracting smaller-scale 3-5 qubit circuit fragments from the input circuit along with a quantum circuit scaffold that fits these fragments together. Dare's compilation flow then lifts and decomposes each qubit circuit fragment to produce an unoptimized qutrit circuit fragment in Dare IR. Each of these fragments is then plugged into the quantum circuit scaffold. Since each fragment has a qubit input/output interface, it can be transparently plugged into the scaffold circuit. Dare's rewrite-based optimizer is then used to optimize the resultant hybrid qubit-qutrit circuit. Since the optimizer is applied to the full, re-assembled circuit, it can identify optimization opportunities that span across multiple fragments. Finally, all gates in the Dare IR are then lowered to qutrit basis gates, producing the final optimized quantum circuit.
Fragment Extraction. Dare parses the qubit circuit and greedily assigns any gate encountered to the sub-circuit with the largest overlap in qubits. The current gate's qubits are removed from all other circuits, and the chosen circuit is ordered after all other candidates. This ordering and qubit mapping is used as the circuit scaffold for assembly.

3 DARE CIRCUIT TRANSFORMS

Dare's rewrite system is built upon a collection of circuit transforms that modify the circuit without affecting its functional behavior. Each basic transform imposes a set of conditions over the involved gates that must hold for it to be applied. We provide a proof sketch that demonstrates the transformed and the original circuit deliver the same behavior provided the required conditions hold. To simplify the management of these conditions, this section introduces specialized bookend, state-invariant, interior, and state-subsetted gates which each meet a set of conditions commonly required across transformations.

We use an outer product formalization of circuits introduced in Section 3.1, with different gate conditions described in Section 3.2. We introduce qubit-equivalence and strict equivalence in Section 3.3 along with useful lemmas and their proofs. Section 3.4 introduces the atomic transforms that follow qubit equivalence (marked with †) or strict equivalence (no †). Figure 8 summarizes the notation used in this formalization.

3.1 Quantum Circuits and Gates

Circuits. A n-qutrit circuit comprises a sequence of n-qutrit gates. Any circuit $Q \in SU(3^n)$ unitary is the product of a sequence of $3^n \times 3^n$ gate unitaries $G \in SU(3^n)$, and is written as $Q = G_m \times ... \times G_j \times ... \times G_0 = \prod_0^m G_j$. The j^{th} gate in the circuit, G_j, is the j^{th} gate from the left.
One-Qutrit Operators. Each n-qutrit gate is comprised of n single-qutrit operators $Z = [z_1,...,z_n]$ applied to each qutrit $1..n$, where z_i is applied to qutrit i. Each z_i is a 3×3 matrix that may implement a one-qutrit transformation (U), a wire ($I(3)$ if not controlled on i), or a control on state $s \in S = \{0,1,2\}$.
N-Qutrit Gates. $Gate([z_1,...z_{k-1},U_k,z_{k+1},...z_m],k)$ applies a unitary transform (U) on the target qutrit (k), and applies identity or control operations on all other qutrits, so $z_j \in \{C(0),C(1),C(2),I(3)\}$ for $j \neq k$. Sub-sequences of operators are denoted with $Z_{i,j} = [z_i...z_j]$, and sequences are concatenated together as $Z_{1,j}Z_{j+1,n}$. Gates are written in this notation as $Gate(Z,k) = G(Z_{1,k-1}[U_k]Z_{k+1,n},k)$.
Outer Products. The $\pi(Z) = \pi([z_1,...z_k]) = z_1 \otimes z_2 ... \otimes z_k$ operator computes the outer product over a list of 3x3 operator matrices $Z = [z_1,..,z_k]$ and produces a $3^k \times 3^k$ matrix over qutrits 1 to k. We note that preserving the order of the unitaries is required to apply the k-qubit gate correctly and the resulting matrix is *not guaranteed* to be unitary.
Gate Unitary. Each gate $Gate(Z,k)$ applies a $3^n \times 3^n$ unitary that can be described with an outer product matrix expression comprised of 3x3 operator matrices. Equations below represent the gate using the standard and reduced-form matrix expressions for a gate $Gate(Z,k)$ respectively:

$$Gate(Z,k) = \pi(Z) + [I(3^n) - \pi(Z_{1,k-1}) \otimes I(3) \otimes \pi(Z_{k+1,n})] = \pi(Z_{1,k-1}) \otimes (U - I(3)) \otimes \pi(Z_{k+1,n}) + I(3^n) \quad (1)$$

In the standard form, the $\pi(Z)$ term applies the gate operation, and the remaining terms normalize the resulting matrix. The reduced form expression is derived from the standard expression by factoring out the $\pi(Z_{1,k-1})$ and $\pi(Z_{k+1,n})$ terms out of the expression. See appendix for full derivation.

3.2 Gate Conditions: Bookends, State-Subsetted, and State-Invariant Gates

We next define bookend gate pairs, state-subsetted/qubit-subsetted gates, state-invariant gates, and interior gates. These special types of gates appear in the gate patterns matched by our rewrites.
State-Subsetted Gates. A qutrit j is state-subsetted to states $S \subset \{0,1,2\}$, where S is a strict subset of the basis states if it can only occupy a superposition of states in S. The state-subset conditions are computed with the Dare IR wire annotations, derived through static analysis described in Section 5. If a qutrit j is state-subsetted to S, then any gate $Gate(Z,k)$ that has a control on state $s' \notin S$ of qutrit j will apply the identity to qutrit k since its control conditions will not be satisfied:

$$Gate(Z,k) = \pi(Z_{1,j-1}) \otimes C(s') \otimes \pi(Z_{j+1,k-1}) \otimes (U - I(3)) \otimes \pi(Z_{k+1,n}) + I(3^n) = I(3^n)$$

A special case of a state-subsetted qutrit is qubit-subsetted (Figure 8c), which only occupies the $|0\rangle$ and $|1\rangle$ basis states ($S = \{0,1\}$). Hence, it is unaffected by gates that operate on $|2\rangle$ basis and cannot satisfy controls on $|2\rangle$ state. At the qutrit circuit interface, all qutrits for Dare circuits are qubit-subsetted.
State-Invariant Gates. A gate $SInv(S,Z,k) = Gate(Z,k)$ is state invariant if it does not modify quantum states involving basis states $s \in S$. A state invariant gate's unitary is shown using U_S, where U_S applies the identity over basis states $s \in S$ ($U_S|s\rangle = |s\rangle$, thus $U_s \times C(s) = C(s)$ because $C(s)$ is just $|s\rangle\langle s|$). The gate implements the following matrix expression over all qutrits $Gate(Z_{1,k-1}[U_S]Z_{k+1,n})$.
Bookend, Interior Gates. Bookend gates are pairs of gates that are applied together. Bookend gates often appear in qutrit circuits: the leading gate is commonly used to lift a qubit-subsetted qutrit to use the $|2\rangle$ state, and the lagging gate pushes it back into the qubit-subsetted state. A leading gate $G_m = Book(Z,U,k) = Gate(Z_{1,k-1}[U]Z_{k+1,n})$ and lagging gate $G_{m+i} = Book(Z,U',k) = Gate(Z_{1,k-1}[U']Z_{k+1,n})$ at positions m and $m+i$ comprise a bookend. The bookends have identical control conditions and apply a transform U, U' to the same target qutrit k.

For a bookend gate pair to be applied together, all the gates G_j that lie between the two bookends ($m < j < m+i$) must be an interior gate $Int(Z',U'',k')$. Interior gates do not affect the controls of the lagging bookend, ensuring the bookends are applied together. To deliver this property, each interior gate must satisfy one of two conditions: The interior gate's target qutrit is not one of the bookend's controls. $z_{k'} = I(3)$, or, the interior gate's target qutrit acts on one of the bookend' controls ($z_{k'} = C(s)$) but the interior gate's transform is invariant to the bookend's control state ($U'' = U_s$).

3.3 Strict Equivalence and Qubit Equivalence

Dare works with two notions of equivalence: Strict Equivalence and Qubit Equivalence. If two circuits are strictly equivalent, they implement the same unitary. If two circuits are qubit-equivalent, they implement the same computation over the $|0\rangle$ and $|1\rangle$ states.

Definition 1. Qubit Equivalence. Two qutrit circuits are qubit-equal if they implement the exact quantum computation over the $|0\rangle$ and $|1\rangle$ states i.e. all elements of the unitary that operate on the $|0\rangle$ and $|1\rangle$ states are preserved. Given the original and transformed circuit unitaries Q and Q', the unitaries are *qubit-equal* if the difference for the unitary of the entire circuit contains an outer product term that is controlled on the $|2\rangle$ state over some qutrit k and the qutrit k is qubit-subsetted:

$$\Delta Q = Q - Q' = \sum U(3^{k-1}) \otimes C(2) \otimes U(3^{n-k}) = \sum U(3^{k-1}) \otimes |2\rangle\langle 2| \otimes U(3^{n-k}) \tag{2}$$

The difference ΔQ is non-zero only if the qubit-subsetted qutrit k contains energy in the $|2\rangle$ state.

Definition 2. Strict Equivalence. The original circuit unitary Q and the transformed unitary Q', are *strictly equivalent*, i.e., equivalent over all possible quantum states, if their difference is zero:

$$\Delta Q = Q - Q' = 0 \qquad\qquad Q = Q' \tag{3}$$

We next define lemmas related to Strict and Qubit Equivalence that help prove Dare transforms.

Lemma 1. *Qubit Equality of Fragments.* A quantum circuit $Q = Q_A \times Q_F \times Q_B$ is qubit equal to the transformed circuit $Q' = Q_A \times Q'_F \times Q_B$ if the circuit fragment Q_F is qubit equal to Q'_F and the k'th qutrit of Q_F is qubit-subsetted on qutrit k at its input and output interface.

Proof Sketch. The quantum unitary difference ΔQ is defined as $Q_A \times [Q_F - Q'_F] \times Q_B$. Since qutrit k is qubit-subsetted at Q_F's interfaces, the gates before and after the fragment cannot modify the k^{th} qutrit to get into $|2\rangle$ state. Thus the product of all gates before and after the fragment are state-invariant to the $|2\rangle$ state and are of the form $\sum \pi(Z_{1,k-1}) \otimes (U_2) \otimes \pi(Z_{k+1,n})$, where U_2 is state invariant over state $|2\rangle$. Equation 2 is used to replace $\Delta Q_F = Q_F - Q'_F$ of the qubit equal fragments for $\Delta Q = Q - Q'$:

$$\left[\sum \pi(Z_{1,k-1}) \otimes U_2 \otimes \pi(Z_{k+1,n})\right] \times \left[\sum \pi(Z_{1,k-1})' \otimes C(2) \otimes \pi(Z_{k+1,n})'\right] \times \left[\sum \pi(Z_{1,k-1})'' \otimes U''_2 \otimes \pi(Z_{k+1,n})''\right]$$
$$= \sum (\pi(Z_{1,k-1}) \times \pi(Z_{1,k-1})' \times \pi(Z_{1,k-1})'') \otimes (U_2 \times C(2) \times U''_2) \otimes (\pi(Z_{k+1,n}) \times \pi(Z_{k+1,n})' \times \pi(Z_{k+1,n})'')$$

Using the property of state invariant unitaries, $U_2 \times C(2) \times U(2)'' = C(2)$ the above equation matches the qubit equality definition (Equation 2). Hence, for any transform on a circuit fragment, if qutrit k at the circuit fragment interface is qubit-subsetted, we only need to show $Q'_F - Q_F = \sum \pi(Z_{1,k-1}) \otimes C(2) \otimes \pi(Z_{k+1,n})$ to prove qubit-equality.

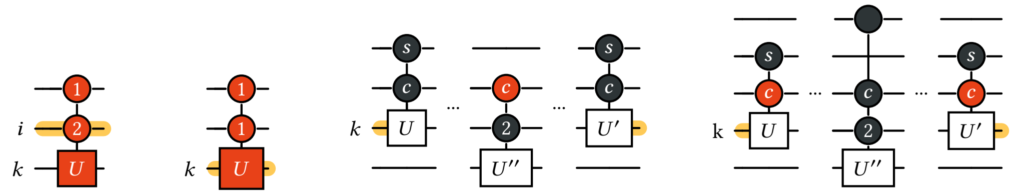
Lemma 2. *Strict Equality of Fragments.* A quantum circuit $Q = Q_A \times Q_F \times Q_B$ is strictly equal to the transformed circuit $Q' = Q_A \times Q'_F \times Q_B$ if the circuit fragment Q_F is strictly equal to Q'_F.

Proof Sketch. If $Q_F - Q'_F = 0$, then $Q_F = Q'_F$ and $Q = Q'$.

Lemma 3. *Gate Commutativity.* Gate $G_1 = Gate(Z_{1,k-1}[U]Z_{k+1,n}, k)$ commutes with gate $G_2 = Gate(Z'_{1,k'-1}[U']Z'_{k'+1,n}, k')$ if for k^{th} qutrit G_1 applies $z_k = U$ and G_2 either has a wire $z'_k = I(3)$ or control $z'_k = C(s)$ if $U = U_s$. The same condition holds in the other direction with U' and G_1's term $z_{k'}$.

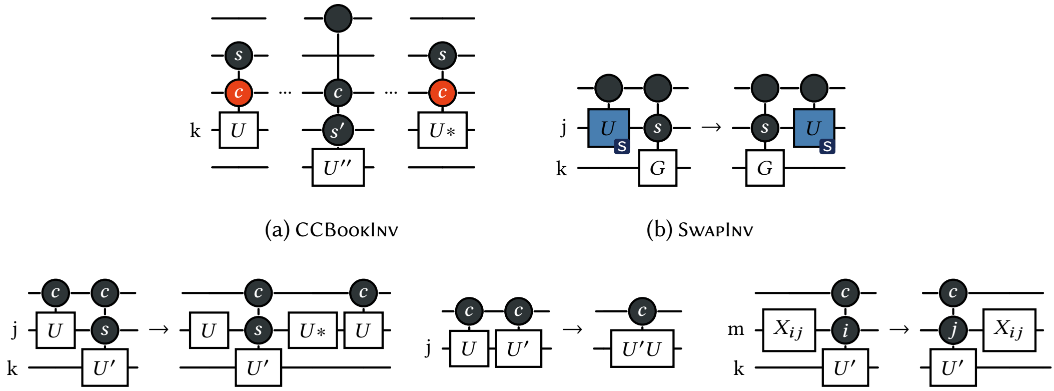
Lemma 4. *Gate Difference Elimination.* For two gates G_1,G_2 both controlled on a qutrit q_l on state $C(s)$ (i.e. given by $G_1 = Gate(Z_{1,l-1}[C(s)]Z_{l+1,n})$,$G_2 = Gate(Z'_{1,l-1}[C(s)]Z_{l+1,n})))$ if a transform deletes the control $C(s)$ on gate $G_2 \rightarrow G'_2 = Gate(Z'_{1,l-1}[I(3)]Z'_{l+1,n})$, then the product of the unaffected gate and the difference of transformed gate unitaries give back the difference term. $G_1 \times (G'_2 - G_2) = G'_2 - G_2$.

Proof Sketches: We provide proofs for lemma 3 and 4 in our appendix.



(a) DelCtrl (b) DelXform (c) CCInter (d) CCBookQubit

Fig. 10. **Qubit equality-preserving basic transforms.** Deleted controls and gates are highlighted in red, wires qubit-subsetted (i.e. subsetted to |0⟩ and |1⟩ states) are shaded in yellow



(a) CCBookInv (b) SwapInv

(c) SwapCtrl (d) Merge (e) ApplyXCtrl

Fig. 11. **Unitary-preserving basic transforms.** $|s\rangle$-invariant gates are shaded in blue with a subscript s.

3.4 Circuit Transforms

3.4.1 DelCtrl†: Delete $|2\rangle$-Controlled Gates. Given a circuit Q containing fragment $Q_F = Gate(Z,k) = Gate(Z_{1,i-1}[C(2)]Z_{i+1,k-1}[U]Z_{k+1,n})$ with a control on state $|2\rangle$ and a qubit-subsetted qutrit i, the gate can be deleted producing $Q'_F = I(3^n)$ while preserving the qubit-equality.
Proof Sketch. Lemma 1 states if Q_F and Q'_F are qubit-equal, then applying the transform to a quantum circuit Q containing Q_F also preserves qubit equality. The unitary difference ΔQ_F for circuit fragments Q_F and Q'_F satisfies Eqn 2, preserving qubit equality over Q_F:

$$\Delta Q_F = Q_F - Q'_F = -\pi(z_1..z_{i-1})\otimes C(2)\otimes\pi(z_{i+1}..z_{k-1})\otimes(U-I_3)\otimes\pi(z_{k+1}..z_n) = U(3^{i-1})\otimes C(2)\otimes U(3^{n-i-1})$$

3.4.2 DelXform†: Delete $|2\rangle$-Transforming Gates. Given a circuit Q containing fragment $Q_F = Gate(Z,k) = Gate(Z_{1,k-1}[U]Z_{k+1,n})$, the gate can be deleted yielding $Q'_F = I(3^n)$ if the unitary $z_k = U_{0,1} = diag(1,1,e^{i\theta})$ is state-invariant to the $|0\rangle$ and $|1\rangle$ states, and the target qutrit k is qubit-subsetted. Such a unitary can only be a phase gate since the U matrix is unitary.
Proof Sketch. According to Lemma 1, if Q_F and Q'_F are qubit-equal, then the transform maintains qubit equality for quantum circuit Q containing Q_F. To prove qubit equality for Q_F,Q'_F we write $U_{0,1}$ as $U_{0,1} = I(3)+(\alpha-1)\times C(2)$ and show the unitary difference $\Delta Q_F = Q'_F - Q_F$ satisfies Eqn 2:

$$\Delta Q = I(3^n) - (Z_{1,k-1}\otimes(U-I(3))\otimes Z_{k+1,n}+I(3^n)) = -\pi(Z_{1,k-1})\otimes(\alpha-1)\times C(2)\otimes\pi(Z_{1,k-1})$$

3.4.3 CCInter†: Control Cancellation of Interior Gates. Given a circuit Q containing a fragment $Q_F = G_m \ldots G_{m+i+1}$ with bookend gates $G_m = Book(Z,U,k)$, $G_{m+i+1} = Book(Z,U',k)$ and i interior gates, a shared control $C(s)$ on some qutrit l can be deleted from interior gates in Q_F if three conditions are met. First, the qutrit k is qubit-subsetted on the input/output interface of circuit fragment Q_F. Second, the interior gate $Int(Z',U'',k')$ is controlled on qutrit k on $|2\rangle$ state. Third, all gates $G_j = Int(Z',U_2,k)$ between bookend G_m and the interior gate that acts on qutrit k are state-invariant to the $|2\rangle$ state.
Proof Sketch. Lemma 1 states if Q_F and Q'_F are qubit-equal, then applying the transform to a quantum circuit Q containing Q_F also preserves qubit equality. We prove by induction, starting with a base case with one interior gate, that qubit equality is preserved for multiple interior gates.
Base Case. We derive the unitary difference $\Delta Q = Q'_F - Q_F$ has a $C(2)$ for a fragment Q_F containing exactly one interior gate $Int(Z',U'',k')$ and two bookends $Book(Z,U',k)$,$Book(Z,U,k)$:

$$\Delta Q_F = Book(Z,U',k) \times \left[\pi(Z'_{1,l-1}) \otimes (I(3)-C(s)) \otimes \pi(Z'_{l+1,k-1}) \otimes (C(2)) \otimes \pi(Z'_{k+1,n})\right] \times Book(Z,U,k)$$

The product of the difference of interior gates and bookends follows lemma 4 eliminating gates in red. Thus, ΔQ_F reduces to the difference of interior gates; hence Q_F, Q'_F are qubit-equal:

$$\Delta Q_F = (\pi(Z'_{1,l-1}) \otimes (I(3)-C(s)) \otimes \pi(Z'_{l+1,k-1}) \otimes C(2) \otimes \pi(Z'_{k+1,n}))$$

Induction. Given qubit-equal original and transformed qubit-equal fragments $Q_{F,i}$, $Q'_{F,i}$ that contain i interior gates (inductive assumption), we show qubit equality is preserved if an interior gate G_y is inserted right before the lagging bookend $Book(Z,U',k)$ of the circuit. We first derive the $\Delta Q_{F,i+1}$ expression is rewritten in terms of $\Delta Q_{F,i}$ and $\Delta Q_{F,1}$ differences:

$$\Delta Q_{F,i+1} = Book(Z,U',k) \times G_{y,T} \times Book(Z,U',k)^{\dagger} \times \Delta Q_{F,i} + \Delta Q_{F,1} \times Book(Z,U,k)^{\dagger} \times \prod G_m \times Book(Z,U,k) \tag{4}$$

With lemma 4 we simplify above $\Delta Q_{F,i+1}$ (eliminating gates in red) and show $\Delta Q_{F,i+1}$ has a $I(3)-C(s)$ term for qutrit l and $C(2)$ for qutrit k. From condition (1) of the rewrite, qutrit k is qubit subsetted and thus ΔQ_{i+1} will satisfy qubit equivalence.

3.4.4 CCBookQubit†: Control Cancellation of Bookends on Qubit Basis. Consider a circuit Q containing a fragment Q_F with gates $G_m \ldots G_{m+i+1}$ containing two bookend gates $G_m = Book(Z,U,k)$, $G_{m+i+1} = Book(Z,U',k)$ and i interior gates $G_j = Gate(Z,k)_j = Int(Z',U'',k'')_j$ where $j \in m+1 \ldots m+i$. We may delete a shared control given by $z_l = C(s)$ acting on qutrit l from both bookend gates if the following conditions are met. First, the qutrit k is qubit-subsetted on the input/output interface of circuit fragment. Second, the product of bookend unitaries is 0,1-state invariant, $U \times U' = diag(1,1,a)$. Third all interior gates fall in *one of the two cases*: if the interior gate either targets the bookend target k, then it is invariant to $|2\rangle$ state, or it is controlled on the bookend gate's target qutrit k over state $|2\rangle$ ($z'_k = C(2)$). In both cases it contains the removed control $z'_l = C(s)$ as well.
Proof Sketch. We will prove by induction that the difference between the original and transformed circuit fragments $\Delta Q_F = Q'_F - Q_F$ has a $C(2)$ term over qutrit k, which is qubit-subsetted (condition (1)). The base case proves qubit equality for Q_F with one interior gate, and the inductive step proves qubit equality for multiple interior gates. With lemma 1 we know that if Q_F and Q'_F are qubit-equal and qubit-subsetted over qutrit k, then the transform maintains qubit equality over the entire quantum circuit Q.
Base Case. Consider the original fragment $Q_F = G_{m+2} \times G_{m+1} \times G_m = Book(Z,U,k) \times Int(Z',U'',k') \times Book(Z,U',k)$ with bookends containing $q_l = C(s)$, and a transformed fragment Q'_F with bookends $G'_m = Book(Z_T,U,k)'$, $G'_{m+2} = Book(Z_T,U^{\dagger},k)$ have their control deleted ($z_{T,l} = I(3)$). After simplification, the unitary difference ΔQ_F is as follows:

$$\Delta Q_F = G'_{m+2} \times G_{m+1} \times [Book(Z_T,U,k) - Book(Z,U,k)] + \left[Book(Z_T,U^{\dagger},k) - Book(Z,U',k)\right] \times G_{m+1} \times G_m$$

We next apply Lemma 4 to eliminate the highlighted G_{m+1} and G_m terms and derive ΔQ:

$$\pi(Z_{1,l-1}) \otimes ((I(3)-C(s)) \otimes \pi(Z_{l+1,k-1}) \otimes \left[(U'-I(3)) \times (U-I(3)) + (U'-I(3)) + (U'-I(3))\right] \otimes \pi(Z_{k+1,n}) \tag{5}$$

The ΔQ_F expression is simplified to satisfy qubit equality (Defn 1). The simplified expression applies the $z_k = C(2)$ operator on qutrit k, which is qubit-subsetted according to condition (1) of the transform.
Induction. Given a quantum circuit fragment $Q_{F,i}$ containing i interior gates and a qubit-equal transformed fragment $Q'_{F,T,i}$ obtained by deleting $q_l = C(s)$ on the fragment's bookends. We create a fragment $Q_{F,i+1}, Q_{F,T,i+1}$ from $Q_{F,i}, Q_{F,T,i}$ with a new interior gate G_y inserted before lagging bookend G_{m+i+1}, totaling $i+1$ interior gates. The unitary difference ΔQ_{i+1} is

$$\Delta Q_{F,i+1} = G'_{m+i+1} \times G_y \times \prod G_i \times (G'_m - G_m) + (G'_{m+i+1} - G_{m+i+1}) \times G_y \times \prod G_i \times G_m \tag{6}$$

We apply Lemma 4 to delete interior gates/original bookend gate (shown in red) and Lemma 3 to commute interior gates. We derive the following expression for ΔQ_{i+1} which satisfies the definition of qubit-equality, as it applies a $C(2)$ operator to the qubit-subsetted qutrit k:

$$\Delta Q_{F,i+1} = \pi(Z_{1,l-1}) \otimes (I(3) - C(s)) \otimes \pi(Z_{l+1,k-1}) \otimes (a \times C(2)) \otimes \pi(Z_{k+1,n}) \times \prod Gate(Z'_{1,k-1}[I]Z'_{k+1,n}, k')$$

Hence qubit-equality holds by induction when an interior gate is inserted before the lagging bookend.

3.4.5 CCBookInv: Control Cancellation of Inverse Bookends. Consider circuit Q containing a fragment $Q_F : G_m ... G_{m+i+1}$ containing bookend gates $G_m = Book(Z,U,k)$, $G_{m+i+1} = Book(Z,U^\dagger,k)$ where $U \times U^\dagger = I$. We can remove some shared control $z_l = C(s)$ on a qutrit l from both bookend gates provided every interior gate $G_{i+j} = Int(Z',U'',k')$ in $j \in m..m+i+1$ satisfies *at least one* of three conditions. First, the interior gate is not controlled by the bookend's target qutrit k. Second, the interior gate targets the bookend's target qutrit $k' = k$ and also applies the shared control $z'_l = C(s)$. Third, the interior gate is controlled on the bookend's target k and also applies the shared control $z'_l = C(s)$ on qutrit l.
Proof Sketch. We can prove by induction that the transform produced a transformed fragment Q'_F that is strictly equal to Q_F. For the base case, Q_F, Q'_F are proven strictly equal with one interior gate, and induction is used to prove qubit equality for multiple interior gates. Lemma 2 is then applied: since Q_F and Q'_F are strictly-equal, then Q and Q' containing Q_F, Q'_F are also strictly equal.
Base Case. We use the same procedure as in Section 3.4.4 and derive ΔQ_F. This gives an expression similar to equation 5, where $U' \longrightarrow U^\dagger$. Now, we use $U \times U^\dagger = I(3)$, to show $\Delta Q_F = 0$, satisfying strict equality:

$$\Delta Q_F = \pi(Z_{1,k-1}) \otimes \Big[(U^\dagger - I(3)) \times (U - I(3)) + (U^\dagger - I(3)) + (U - I(3)) \Big] \otimes \pi(Z_{k+1,n}) = 0$$

Induction. Given strictly equal circuit fragments $Q_{F,i} = Q'_{F,i} = G_{m+i+1} \times G_{m+i} \times ... \times G_m$ and $Q_{F,1} = Q'_{F,1} = G_{m+i+1} \times G_y \times G_m$ containing i interior gates and 1 interior gate respectively. We prove that strict equality holds for a quantum circuit $Q_{F,i+1}, Q'_{F,i+1}$ constructed from $Q_{F,i}, Q'_{F,i}$ with a gate G_y inserted before the lagging bookend G_{m+i+1}. First, we show that $Q_{F,i+1} = Q_{F,1} \times Q_{F,i}$: $Q_{F,1} \times Q_{F,i} = [G_{m+i+1} \times G_y \times G_m] \times [G_{m+i+1} \times G_{m+i} \times ... \times G_m]$. But $G_m \times G_{m+i+1} = I(3^n)$ (they have same control conditions and $U \times U^\dagger = I(3)$). So $Q_{F,i+1} = Q_{F,1} \times Q_{F,i}$. But, $Q_{F,i} = Q'_{F,i}$ and $Q_{F,1} = Q'_{F,1}$. Hence, with substitution $Q_{F,i+1} = Q_{F,i} \times Q_{F,1} = Q'_{F,i} \times Q'_{F,1} = Q'_{F,i+1}$. Therefore, $Q_{F,i+1} = Q'_{F,i+1}$, proving strict equality.

3.4.6 SwapCtrl: Control-Subsetted Gate Swap. Consider a circuit Q with fragment unitary $Q_F = G_{m+1} \times G_m$ where $G_m = Gate(Z,j)$, $G_{m+1} = Gate(Z',k)$ and $z_j = U$. These gates may be swapped producing $Q'_F = G_y \times G_m \times G_y \dagger \times G_{m+1}$, where $G_y = Gate(I_{1,j}[U]I_{j,n})$,$G_y\dagger = Gate(I_{1,j}[U^\dagger]I_{j,n})$ are applied before and after G_{m+1}. This transform ensures strict equivalence, provided two conditions hold. First, gate G_{m+1} is either controlled on qutrit j ($z_j = C(s)$) or targets the same qutrit as G_m ($k = j$). Second, controls of G_m are a subset of the controls of G_{m+1} ($z_l = C(s) \implies z'_l = C(s)$).
Proof Sketch: For Q_F and Q'_F to be strictly equal, then $Q_F = Q'_F$. Starting from Q_F, we rewrite the expression to derive Q'_F, therefore proving Q_F and Q'_F are strictly equal. To perform this rewrite, we leverage the property $\pi(Z_{i,j}) \times \pi(Z'_{i,j}) = \pi(Z'_{i,j})$, which holds since the controls of the first gate are a subset of the controls of the second gate. Therefore the $\Delta Q = 0$ and strict equality is preserved. Lemma 2 is then applied to demonstrate Q containing Q_F is strictly equal to Q', given Q_F is strictly equal to Q'_F.

3.4.7 SwapInv: State Invariant Gate Swap. Consider a circuit Q containing a fragment $Q_F = G_m \times G_{m+1}$ with $G_m = Gate(Z,j)$ and $G_{m+1} = Gate(Z',k)$. We can swap the gates producing transformed fragment $Q'_F = G_{m+1} \times G_m$ provided G_{m+1} applies a control on state s of qutrit j ($z'_j = C(s)$), and G_m applies a state-invariant unitary to s ($z_j = U_s$) and is not controlled on target of G_{m+1} ($z_k = I(3)$).
Proof Sketch. The same proof strategy used in action 3.4.6 is applied to prove strict equality. To rewrite Q_F to Q'_F, we apply Lemma 3 to commute unitaries, applied here since U_s is invariant to $C(s)$.

3.4.8 Merge: Merge Gates. Consider a circuit Q containing a fragment $Q_F = G_{m+1} \times G_m$ where $G_m = Gate(Z_{1,k-1}[U]Z_{k+1,n},k)$ and $G_{m+1} = Gate(Z_{1,k-1}[U']Z_{k+1,n},k)$ are gates that have identical controls and target qutrit k. The transformed circuit fragment $Q'_F = G_y$ merges gates G_m and G_{m+1} to produce the merged gate $G_y = Gate(Z_{1,k-1}[U' \times U]Z_{k+1,n},k)$ – this action produces a transformed fragment Q'_F that is strictly equal to Q_F, the original circuit fragment.
Proof Sketch. The same proof strategy used in action 3.4.6 is applied. To rewrite Q_F to Q'_F, we leverage the property that G_m, G_{m+1} have same control conditions.

3.4.9 ApplyXCtrl: Apply X Gate to Control. Consider a circuit Q containing a fragment $Q_F = G_{m+1} \times G_m$ with a non-controlled gate $G_m = Gate(Z,k)$ which applies $z_k = X_{ij}$ on qutrit k. $G_{m+1} = Gate(Z',l)$ is a controlled gate with a control $z'_k = C(i)$ on qutrit k. X_{ij} gate swaps the $|i\rangle,|j\rangle$ states of the target qutrit. These gates can be swapped producing fragment $Q'_F = G_m \times G'_{m+1}$ where the transformed gate $G'_{m+1} = Gate(Z_T,l) = Gate(Z_{1,k-1,T}[C(j)]Z_{k+1,n,T},l)$ applies control $C(j)$ to qutrit k ($z_{k,T} = C(j)$).
Proof Sketch. The same proof strategy used in action 3.4.6 is applied to prove strict equality. Since $U = U^\dagger = X_{ij}$ in this case, the G_m gate unitaries can be applied to $z_k = C(i)$ to yield $z_{k,T} = C(j)$.

4 QUTRIT CIRCUIT REWRITES

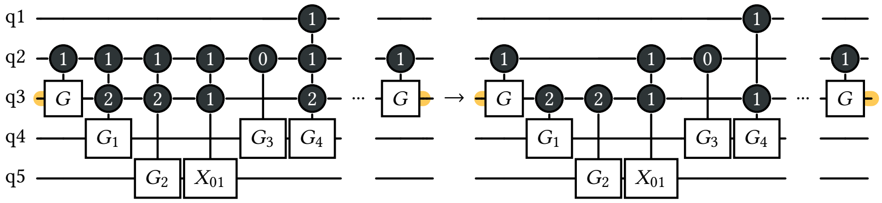


Fig. 12. Rule 1: Control cancellation of gates.

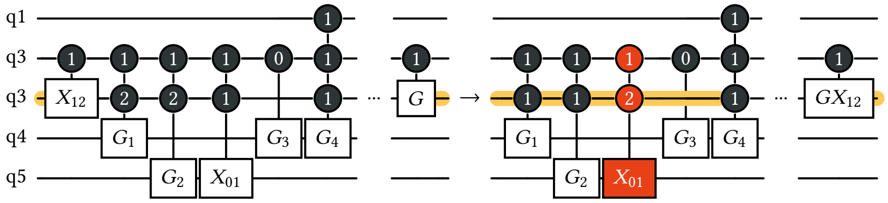


Fig. 13. Rule 2: Gate cancellation of bookends.

The circuit rewrites employed by the Dare rewrite engine are constructed from the circuit transforms from Section 3. Each rewrite matches a gate pattern and applies one or more transforms.
Rule 1: *Control cancellation of gates.* Rule 1 (Figure 12) selects a pair of bookends and eliminates all shared interior controls that can be safely eliminated. Dare finds a bookend pair that satisfies the

required conditions for the application of the CCInter transformation. The CCInter transformation is iteratively applied until all shared controls have been eliminated from the interior gates.

Rule 2: *Gate cancellation of bookends.* Rule 2 (Figure 13) eliminates a pair of bookend gates by relocating the leading bookend gate to the lagging bookend gate and then merging them together. Rule 2 then deletes any formerly interior gates that are now impossible to apply. The rewrite first finds a pair of bookend gates with interior gates that can potentially be swapped with the leading bookend. The rewrite then swaps the leading bookend g with all interior gates using the SwapInv, SwapCtrl, and ApplyXCtrl transforms until it is adjacent to gate g'. The Merge transform is then applied to merge g and g'. After merging the bookends, Dare iteratively applies the DelCtrl and DelXform transforms to eliminate gates. If the target bookend cannot be merged, the rewrite is reverted.

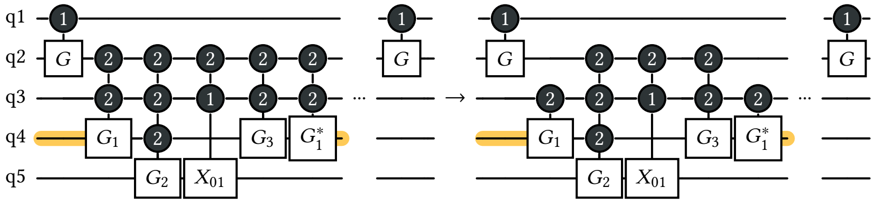


Fig. 14. Rule 3: Control cancellation on bookends.

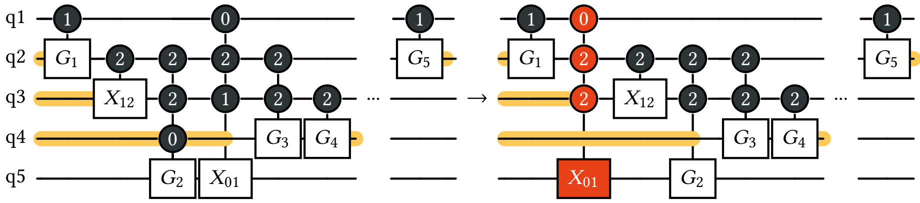


Fig. 15. Rule 4: Delete redundant gates

Rule 3: *Control cancellation on bookends.* Rule 3 (Figure 14) eliminates any redundant or unnecessary controls from the bookend gates. The rewrite first finds a set of bookend gates that implement the identity matrix when applied together. The rewrite iteratively applies the CCBookQubit and CCBookInv to eliminate redundant controls from the bookend gates.

Rule 4: *Delete redundant gates.* Rule 4 (Figure 15) selects and moves a $|2\rangle$-controlled gate until it can be deleted using the DelCtrl and DelXform transformations. The rewrite first selects a gate that can be swapped with its preceding and succeeding gates. The rewrite then speculatively moves the gate g to using SwapInv and SwapCtrl operations until a DelCtrl transformation can be applied. If the target gate cannot be eliminated, the rewrite is reverted.

Circuit Rewrite Algorithm Dare's rewrite engine rewrites a Dare-IR quantum circuit using the four rewrite rules described in Section 4. The algorithm first iteratively applies Rule 4 to eliminate all gates that can be reshuffled and deleted. The algorithm then non-deterministically applies Rules 1, 2, and 3 to gate targets that match the rules' gate patterns. The algorithm prioritizes applying Rule 2 if a site is available and then greedily applies Rules 1 and 3. This process is repeated K times to obtain a collection of optimized circuits. The algorithm estimates the depth for each of the "K" optimized circuits and returns the circuit with the lowest depth. Dare also validates that each optimized circuit is qubit-equal to the original, unoptimized circuit to ensure the original computation is implemented faithfully.

$$\underbrace{\begin{bmatrix}U_0&0&0\\0&U_{A1}&0\\0&0&U_{A2}\end{bmatrix}}_{AU}\underbrace{\begin{bmatrix}I&0&0\\0&C_A&-S_A\\0&S_A&C_A\end{bmatrix}}_{ACS}\underbrace{\begin{bmatrix}I&0&0\\0&V_{A1}&0\\0&0&V_{A2}\end{bmatrix}}_{AV}\underbrace{\begin{bmatrix}C&0&-S\\0&I&0\\S&0&C\end{bmatrix}}_{CS}\underbrace{\begin{bmatrix}U_0&0&0\\0&U_{B1}&0\\0&0&U_{B2}\end{bmatrix}}_{BU}\underbrace{\begin{bmatrix}I&0&0\\0&C_B&-S_B\\0&S_B&C_B\end{bmatrix}}_{BCS}\underbrace{\begin{bmatrix}I&0&0\\0&V_{B1}&0\\0&0&V_{B2}\end{bmatrix}}_{BV}$$

Fig. 16. Decomposed block-diagonal and tile-diagonal matrices after CS Decomposition. The AU, AV, BU, BV matrices are block diagonal, and the ACS, BCS, and CS matrices are tile diagonal.

Search Heuristics. The choice of applying Rule 1 and Rule 3 greedily while using a random sampling for Rule 2 is a heuristic used by the rewrite algorithm. This heuristic is informed by the observation that applying Rule 2 can enable new opportunities for applying Rules 1 and 3. For all evaluated circuits, the number of trials required to find the best circuit implementation was ≤ 10.

5 WIRE STATE ANALYSIS.

Dare's rewrite engine works with wire annotations, which specify the subset of $|0\rangle$,$|1\rangle$,$|2\rangle$ states that are currently in use along that wire Dare employs a static analysis that identifies which wires are using the qubit basis (i.e. in $|0\rangle$,$|1\rangle$ or a binary basis) or the qutrit basis (i.e. in $|0\rangle$,$|1\rangle$,$|2\rangle$). The tool's analysis then propagates input qutrit information forward through the circuit with a forward pass and then propagates output qutrit information backwards through the circuit with a backward pass:

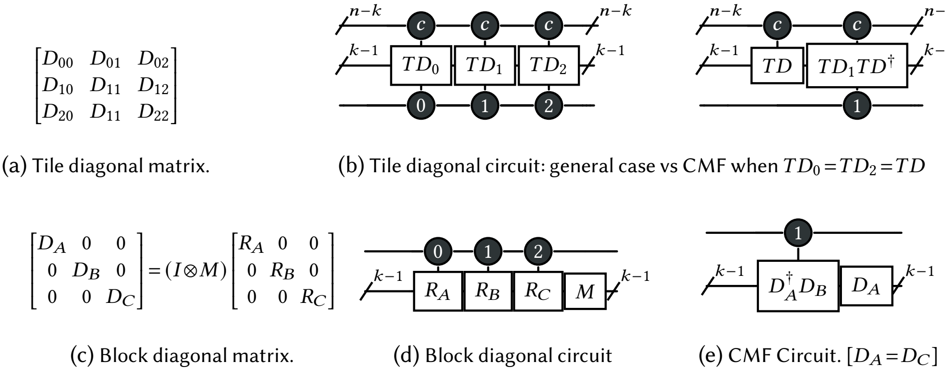
- **Initialization.** All wires at the input and output interface are annotated with a $|0\rangle$, $|1\rangle$ basis since the circuit implements a qubit input/output interface.
- **Forward Pass.** This forward pass labels wires, starting from the circuit inputs, by stepping through gates until the end of the circuit is reached. For each successive gate operating on an annotated input wire, the gate's output wire is annotated with the set of possible basis states after the gate is applied. If the gate is controlled, both the disabled and enabled cases are considered. To improve the precision of the annotations, the algorithm considers cases where gates are applied together in its analysis.
- **Backward Pass.** The analysis then moves backwards through the circuit, starting at the circuit outputs, and propagates wire state information from the qubit output interface. The analysis steps backwards through gates by applying the inverse of each gate's unitary and identifying the set of possible qutrit basis states, given the output wire annotation. If the input wire already has an annotation, the more restrictive of the two annotations is applied.

6 DARE DECOMPOSITION ALGORITHM

Our decomposition algorithm is inspired from the algorithm sketch for generalized QSD in [4, 14]. Dare decomposes the input $3^n \times 3^n$ sized unitary matrix and produces a generate Dare-IR circuit comprised of multi-controlled single qutrit-gates.

Algorithm. The algorithm recursively decomposes the unitary matrix into progressively smaller unitaries until a 3×3 unitary is produced. At each step, the decomposition algorithm involves various components which are as follows:

- **Cosine-Sine Decomposition (CSD).** The first step is the decomposition of the unitary. CSD can partition a $3^k \times 3^k$ unitary into only two sections. Therefore, we apply the decomposition twice, using a 3^{k-1} partition each time. This breaks the unitary neatly into block-diagonal (BD) and tiled diagonal (TD) matrices which comprise $3^{k-1} \times 3^{k-1}$ sub-blocks [28, 29]. Figure 16 presents the structure of the decomposed matrix after CSD. The $3^k \times 3^k$ BD matrix has three $3^{k-1} \times 3^{k-1}$ sub-matrices along the diagonal, while the $3^k \times 3^k$ TD matrix contains nine $3^{k-1} \times 3^{k-1}$ diagonal sub-matrices.



(a) Tile diagonal matrix.

(b) Tile diagonal circuit: general case vs CMF when $TD_0 = TD_2 = TD$

(c) Block diagonal matrix.

(d) Block diagonal circuit

(e) CMF Circuit. $[D_A = D_C]$

Fig. 17. Quantum Circuits after Tile Diagonal Decomposition and Block Diagonal Decomposition. The D_A, D_B, and D_C are $3^{k-1} \times 3^{k-1}$ matrices, and the A_i are diagonal $3^{k-1} \times 3^{k-1}$ matrices.

- **Block Diagonal Decomposition.** For each block diagonal matrix, the algorithm unrolls the block-diagonal matrix into a sequence of three controlled $3^{k-1} \times 3^{k-1}$ sub-matrices. Each controlled sub-matrix has a distinct control on the top qutrit. Therefore, for a target block diagonal matrix shown in Figure 17c, the algorithm implements the circuit as shown in Figure 17d.
- **Common Matrix Factorization.** Our algorithm employs the CMF optimization to reduce the complexity and depth of the produced circuit when the block diagonal matrix has common submatrices. Figure 17e shows the circuit produced by the decomposition procedure when the common matrix factorization is applied towards the right if $D_A = D_C$ (first and last block are identical). CMF optimization estimates the number of controlled gates required to implement the leftover matrix if CMF is performed in the left or right direction (leaving $D_B D_A^\dagger$ vs $D_A^\dagger D_B$ for Figure 17e). Using this estimated cost of factorization, CMF chooses the implementation with lower complexity.
- **Tile Diagonal Decomposition.** The algorithm uses tile diagonal decomposition to implement a $3^k \times 3^k$ tile-diagonal matrix as a sequence of controlled 3×3 sub-matrices. Each matrix is directly mapped to a series of multi-controlled single qutrit gates, with control on the bottom. Figure 17a shows the target tile diagonal matrix. Figure 17b shows the circuit produced after Tile Diagonal Decomposition is applied (left), along with the resulting circuit when CMF is applied (right).

After one step of decomposition is applied, the algorithm then recursively decomposes the unitary sub-matrices (with size $3^{n-1} \times 3^{n-1}$) in the block diagonal and tiled diagonal circuit with the unrolled controls. This continues until each matrix is decomposed into a multi-controlled, single-qutrit gate.

7 EVALUATION

We evaluate Dare on 20 benchmark circuits (Table 1) including circuits from prior work on qutrits [9], arithmetic toffoli-based adder circuits [3, 30], circuits used in evaluating other logical circuit optimization tools [22, 31] along with three toffoli-based synthetic benchmarks. All qutrit circuits are logical circuits implemented with the $Rx_{ij}(\theta), Ry_{ij}(\theta), Rz_{ij}(\theta), CNOT_{ij}$ gate set, where $i,j \in [0,1,2]$. For each benchmark, we study the following circuit implementations:

- **Qubit**: The qubit baseline implements the computation with a qubit-based circuit. We use each benchmark's qubit-based implementation if one is provided (✓), or we generate a T, H, and CNOT gate-based implementation (~) if the benchmark is comprised of Toffoli gates [23]. All qubit circuits are logical circuits implemented with the $Rx, Ry, Rz, CNOT$ gate set.

Table 1. Benchmark quantum circuits. `large` circuits use DARE's scalable compilation flow, ✓ have hand-implemented qutrit circuit baselines, ∼ use qutrit-toffoli circuit baselines, `qubit` have a qubit implementation.

benchmark	cite	# bmarks	# qubits	qubit	manual	large	description
toffoli-<n>	[9]	4	3-6		✓		**n**-qubit Toffoli gates
incrementer-<n>	[9]	2	4-5		✓		n-qubit incrementor circuits
takahashi-<n>	[30]	3	4-8	✓	∼		**n**-qubit takahashi adder.
cuccaro-<n>	[3]	2	4-6	✓	∼		Toffoli gate-based **n**-qubit Cuccaro circuits.
rand-circ-<m>		3	6	✓	∼		Random **m**-Toffoli gate arithmetic circuits
nam-<bmark>	[22, 31]	6	6-30	✓	∼	✓	nam gateset arithmetic circuits

- **Quartz**: Quartz baseline evaluates circuits used in qubit with the Quartz compiler [31]. We generate circuits in the Rx,Ry,Rz,$CNOT$ as well as the nam gateset (used in [31]) and report the best result between the two.
- Manual: The manual baseline implements the computation with a hand-implemented qutrit circuit if one exists (✓). If no such implementation exists (marked ∼), we generate a reference implementation by replacing circuit Toffolis with the qutrit-based Toffoli gate from [9].
- **DARE**: DARE-generated qutrit circuits are generated from the unitary matrix of the corresponding qubit circuit and then mapped down to a qutrit circuit.

7.1 Experimental Setup and Results

7.1.1 Metrics. For each benchmark and baseline, we report the 2-qubit/qutrit gate count, the 1-qubit/qutrit gate count, and the circuit depth. We also report the average fidelity metric (AFM) from circuit simulations; similar simulation framework was used in prior work [9] to estimate mean fidelity for qutrits. The AFM is computed with the formula $\frac{1}{shots}\sum_{i=0}^{shots}|\langle\psi_{ideal}|\psi_{res}\rangle|^2$, and a higher AFM value corresponds to a more accurate circuit. AFM is calculated with resultant state of a noisy simulation given by $|\psi_{res}\rangle$ and ideal simulation given by $|\psi_{ideal}\rangle$ for $shots$ number of random input states.

7.1.2 Fidelity Simulations. We simulate qutrit and qubit-based circuits using Cirq, where we report AFM for 5000 runs for each circuit. We chose Cirq for simulation since it supports noise modeling for qutrit circuits. The qutrit noise model, which modeled superconducting and trapped-ion qutrit devices, was initially developed in [9] and merged into the main Cirq branch. We developed models for the projections by adjusting parameters in Cirq's qutrit model. We perform simulations for all the circuits with less than nine qutrits since larger qutrit benchmarks cannot be tractably simulated. We use the simulation approach described in [9] using state-vector simulation. The qubit/qutrit error models for the simulation were derived from [13, 21] to simulate qutrit errors. These error models were also used in evaluation of [9]. The superconducting error model here are based on IBM's superconducting quantum computers, [9, 12, 15] and the trapped ion error models are from a $^{131}Yb^{+}$ ion-based implementation [9].

- The **sce** error model captures the behavior of superconducting quantum hardware and takes gate error to be 10^{-3} for 1-qubit and 10^{-2} for 2-qubit gates. Whereas qutrit-based hardware globally operates at a 2-6x lower fidelity (due to scaling of error across a symmetric error channel) than qubit-based hardware. Furthermore, the $|2\rangle$ state collapses 2x faster than $|1\rangle$ for state damping. The gate times taken are 100 and 300 ns for 1- and 2-qutrit gates respectively.
- The **stbl2** model modifies sce to have qutrit gates with similar error characteristics as qubit superconducting hardware, capturing the effects of improving the stability of $|2\rangle$ state.
- The **proj** model is based on sce but reduces the error rates for all quantum devices and gates by 100x for all technologies, capturing the effects of globally improving the stability of quantum hardware.

Table 2. Gate count and depth comparison between manual and **Dare**.

	manual			**Dare**			**relative change**		
benchmark	2-qutrit gates	1-qutrit gates	depth	2-qutrit gates	1-qutrit gates	depth	2-qutrit Δ	1-qutrit Δ	depth Δ
toffoli-3†	3	2	5	3	2	5	**0**	**0**	**0**
toffoli-4†	7	9	15	5	4	9	-2	-5	-6
toffoli-5†	14	21	20	10	7	11	-4	-14	-9
incrementer-4†	9	12	16	7	5	12	-2	-7	-4
incrementer-5†	28	47	45	10	7	17	**-18**	**-40**	-28
takahashi-4	11	8	16	5	2	6	-6	-6	-10
takahashi-6	23	16	29	12	6	15	-11	-10	-14
takahashi-8	35	24	42	22	18	29	-13	-6	-13
cuccaro-4	16	12	27	8	4	12	-8	-8	-15
cuccaro-6	31	24	41	18	10	26	-13	-14	-15
random-circuit-6	24	26	48	18	12	30	-6	-14	-18
random-circuit-11	49	48	85	37	28	56	-12	-20	-29
random-circuit-16	71	66	125	33	26	44	-38	-40	**-81**
nam-csla-mux-3	70	56	64	50	20	35	-20	-36	-29
nam-csum-mux-9	140	140	50	84	84	31	-56	-56	-19
nam-gf4̂-mult	83	72	70	51	32	40	-32	-40	-30
nam-mod5-4	24	22	41	16	9	24	-8	-13	-17
nam-mod-mult-55	35	28	39	27	22	28	-8	-6	-11
nam-vbe-adder-3	54	52	70	30	16	26	-24	-36	-44

Table 3. Gate count and depth comparison between qubit-based circuits and Dare.

	qubit			**Dare**			**relative change**		
benchmark	2-qubit gates	1-qubit gates	depth	2-qutrit gates	1-qutrit gates	depth	2-qutrit Δ	1-qutrit Δ	depth Δ
toffoli-3	6	9	11	3	2	5	-3	-7	-6
takahashi-4	16	18	24	5	2	6	**-11**	**-16**	-18
takahashi-6	33	36	44	12	6	15	-21	-30	-29
takahashi-8	50	54	65	22	18	29	-28	-36	-36
cuccaro-4	18	20	27	8	4	12	-10	-16	-15
cuccaro-6	35	40	50	18	10	26	-17	**-30**	-24
random-circuit-6	36	55	59	18	12	30	-18	-43	-29
random-circuit-11	66	100	98	37	28	56	-29	-72	**-42**
random-circuit-16	96	145	148	33	26	44	-63	-119	**-104**
nam-csla-mux-3	80	130	80	50	20	35	**-30**	-110	-45
nam-csum-mux-9	168	364	72	84	84	31	-84	-280	-41
nam-gf24̂-mult	99	190	133	51	32	40	-48	-158	-93
nam-mod5-4	28	51	59	16	9	24	-12	-42	-35
nam-mod-mult-55	48	99	60	27	22	28	-21	-77	-32
nam-vbe-adder-3	70	120	97	30	16	26	-40	-104	-71

- The **tie** trapped ion error model considers noise in trapped ion quantum computers which are more stable but less scalable than superconducting quantum computers and have fewer qubits/qutrits.

7.1.3 Gate Counts and Depth. Table 2 compares gate counts and depth for Dare qutrit circuits against the manual qutrit circuits. For the handcrafted benchmarks (†), Dare reduces depth by up to 28 gates and uses up to 18 fewer 2-qutrit gates than manual implementations. Therefore, Dare produces competitive or better qutrit circuits than even the handcrafted baselines from [9]. For the remaining Toffoli-based adders and other nam benchmarks, Dare attains up to 81 lower depth, up to 38 fewer

Table 4. Gate count and depth comparison for nam benchmarks between Quartz and Dare.

	Quartz			**Dare**			**relative change**		
benchmark	2-qubit gates	1-qubit gates	depth	2-qutrit gates	1-qutrit gates	depth	2-qutrit Δ	1-qutrit Δ	depth Δ
toffoli-3	6	9	12	3	2	5	-3	-7	-7
takahashi-4	10	12	16	5	2	6	-5	-10	-10
takahashi-6	23	24	38	12	6	15	-11	-18	-23
takahashi-8	37	36	57	22	18	29	-15	-18	-28
cuccaro-4	13	15	24	8	4	12	-5	-11	-12
cuccaro-6	25	30	45	18	10	26	-7	-30	-24
random-circuit-6	26	35	42	18	12	30	-8	-13	-12
random-circuit-11*	53	61	81	37	28	56	-16	-33	-15
random-circuit-16∗	75	87	112	33	26	44	**-42**	-61	**-68**
nam-csla-mux-3∗	76	78	65	50	20	35	-26	**-58**	-30
nam-csum-mux-9∗	128	140	42	84	84	31	-44	-56	-11
nam-gf24̂-mult∗	96	81	101	51	32	40	-45	-59	-61
nam-mod5-4	15	11	17	16	9	24	**+1**	**-2**	**+7**
nam-mod-mult-55∗	39	54	45	27	22	28	-12	-32	-17
nam-vbe-adder-3∗	40	45	57	30	16	26	-10	-29	-31

2-qutrit gate counts and up to 40 fewer 1-qutrit gates. For all benchmarks, Dare achieves relative depth reductions of 0-65%, 2-qutrit gate count reductions of 0-64% and 1-qutrit gate count reductions of 0-85%. We mark all the benchmarks with the highest and lowest percentage improvements in **bold** in Table 2. We find that Dare delivers gate count and depth improvements, even when compared against benchmarks that have been hand-optimized as well as benchmarks that use optimized Toffoli circuits. Therefore, optimizing across gates delivers additional benefits that are otherwise missed when optimizing individual gate implementations.

Table 3 compares Dare's qutrit circuits against hand-implemented and Toffoli-based qubit circuits. Dare attains a 6-104 depth reduction and uses 3-84 fewer 2-qubit/qutrit gates and 7-280 fewer 1-qubit/qutrit gates over manually implemented qubit circuits. This translates to a 43-75% reduction in depth, a 38-69% improvement in 2-qubit gate count and a 67-89% improvement in 1-qubit gate count. Therefore, Dare-optimized circuits achieve significant depth and gate count reductions over qubit circuits. Therefore, qutrit circuits reduce circuit depth without requiring extra quantum elements.

Table 4 compares the results for circuits from Dare and Quartz [31] compiler. For almost all benchmarks, Dare finds more efficient circuits than Quartz. Dare delivered up to 45 fewer 2-qutrit gates, 2-61 fewer 1-qutrit gates and improves depth by up to 74 gates. This is a -0.06 to 56.6% reduction in 2-qubit gates, 18.1 to 74.4% reduction in 1-qubit gates and up to 62.7% reduction in depth. For this comparison, we keep Dare in its gateset and report Quartz's performance with its preferred gateset (between nam and Rx,Ry,Rz,$CNOT$). From the 15 tests in the qubit baseline, Quartz timed out for 7 tests after 24 hours (marked with ∗ in table 4). Hence, we validated the results from Quartz against those reported in [31] (for nam benchmarks) and found our reported numbers are better or at least identical.

7.1.4 Circuit Fidelity Comparison. Table 5 reports the fidelity results for the noise models for all benchmarks smaller than 9 qutrits. For the sce noise model, Dare circuits achieve up to 8.9× higher fidelity than manual qutrit circuits. Hence, Dare's depth improvements translate to improved fidelity. However, with sce noise model, qubit-based circuits can attain 1.07-10.2 times *better* fidelity than manual qutrit circuits. While Dare's circuits close some of this gap, the stability of qubit-based hardware today is still significantly greater, leading to worse fidelity with qutrits. So, we also evaluate the effect of improving quantum hardware technology with two other models, stable2 and proj.

Table 5. Fidelity study.

	qubit			manual				Dare			
benchmark	sce	Proj	tie	sce	Stbl2	Proj	tie	sce	Stbl2	Proj	tie
toffoli-3	0.916	0.999	0.995	0.853	0.999	0.953	0.977	0.821	1.000	0.952	0.987
toffoli-4	-	-	-	0.628	0.995	0.871	0.967	0.724	0.998	0.922	0.979
toffoli-5	-	-	-	0.383	0.995	0.740	0.919	0.530	0.995	0.806	0.952
incrementer-4	-	-	-	0.550	0.996	0.828	0.942	0.646	0.996	0.882	0.954
incrementer-5	-	-	-	0.131	0.990	0.461	0.858	0.523	0.995	0.794	0.935
takahashi-4	0.769	0.998	0.979	0.513	0.995	0.790	0.940	0.738	0.996	0.934	0.974
takahashi-6	0.539	0.988	0.962	0.230	0.984	0.631	0.881	0.482	0.989	0.806	0.944
takahashi-8	0.329	0.981	0.927	0.078	0.974	0.441	0.841	0.219	0.990	0.567	0.877
cuccaro-4	0.740	0.998	0.987	0.378	0.988	0.736	0.912	0.581	0.996	0.874	0.960
cuccaro-6	0.473	0.990	0.952	0.132	0.984	0.504	0.834	0.287	0.991	0.662	0.901
random-circuit-6	0.468	0.988	0.967	0.179	0.983	0.583	0.881	0.299	0.993	0.686	0.911
random-circuit-11	0.233	0.989	0.925	0.057	0.974	0.317	0.775	0.092	0.977	0.469	0.827
random-circuit-16	0.132	0.968	0.878	0.013	0.967	0.228	0.674	0.116	0.986	0.496	0.819
nam-mod5-4	0.526	0.983	0.949	0.210	0.991	0.614	0.904	0.373	0.992	0.726	0.932

For the Stbl2 error model where we improve the stability of the $|2\rangle$ state, Dare attains 1.1-7.5× the fidelity of qubit. For the proj error model, Dare attains comparable fidelity to qubit for the same improved qubit gate counterparts. Therefore, improving the stability of qutrit-based hardware can enable gate count/depth reductions to translate to fidelity improvements. For trapped ion quantum hardware (tie), the Dare, manual, and qubit attain similar fidelity across all benchmarks (within 1-3%). Therefore, independent of hardware realization, Dare's gate and depth reductions help bridge the fidelity gap between qubit and qutrit-based circuits and also possibly deliver fidelity improvements if the stability of qutrits is increased in the future. Furthermore, the reduced 2-qubit gate counts can lead to further improvements when considering mapping to hardware.

7.1.5 Dare Execution Time and Scalability. Decomposing n-qutrit circuit unitary requires decomposing a $3^n\times3^n$ matrix. Without circuit splitting, Dare's decomposition algorithm is a major bottleneck in compilation, as it processes $3^n\times3^n$ matrices and therefore scales exponentially with qubit count and exhibits worse scalability properties than qubit circuit decomposition algorithms, which work with $2^n\times2^n matrices$. For example, qutrit and qubit circuit decomposition algorithms require 8 hours and 103.23 seconds respectively to decompose an 8-qubit circuit. Dare solves this issue with the circuit-splitting pass that breaks up the circuit into several much smaller unitary matrices. This allows Dare to decompose larger benchmarks from nam gate set, with the largest circuit (csla-mux-9 with 30 qutrits) taking only 52.40 seconds to decompose. With circuit splitting, twenty-five $3^3\times3^3-3^4\times3^4$ sized unitary matrices are analyzed instead of a $3^30\times3^30$ matrix. Dare's rewrite algorithm takes only 3.20-968.76 seconds to run, depending on the benchmark, and is not a significant bottleneck in scalability.

8 DISCUSSION

Novelty of Quantum Shannon Decomposition. Previously developed generalized Quantum Shannon Decomposition (QSD) methods only provide a high-level description of how to perform decomposition and perform a pen-and-paper analysis of their method [4, 14]. To realize Dare, we translate these descriptions into a numeric algorithm implementable with standard numerical computing libraries. In addition, the circuits produced by the original decomposition algorithm are inefficient, so we engineered the common matrix factoring optimization (CMF) to reduce the complexity of generated circuits. CMF optimization is specific to Qutrit QSD and factors out any repeated/common matrices, reducing depth and the number of controlled gates. It supports both block diagonal and tiled diagonal matrices – both matrix structures appear during decomposition. Figures 17d/17e, and 17b show block-diagonal and tiled-diagonal circuits before and after CMF is applied.

Utility of Decomposition Algorithm. Matrix decomposition is an essential step in Dare's compilation procedure, as it produces a qutrit circuit which effective uses of the $|2\rangle$ state from qubit circuit. We note that in the future, the decomposition step can potentially be replaced with novel algorithms which fold transient circuit information into a higher energy quantum states. These methods would sidestep the decomposition process entirely. Dare's circuit rewriting can also be used to optimize these qutrit circuits. However, for novel non-decomposition-based qutrit circuit construction methods to be successful, it's critical to study efficient qutrit implementations of qubit circuits, which the Dare compiler can provide.

9 RELATED WORK

Qutrit-based Circuits. Researchers have previously devised hand-implementations of qutrit circuits for only a small set of quantum gates and analyzed the tradeoffs associated with such circuits [1, 9, 16]. Dare builds upon this work and automatically generates such circuits that use higher-order states for computation from a qubit unitary matrix. While, researchers have also developed software techniques for mapping computations to ququarts [17, 18], these techniques specifically exploit the fact that ququarts, like qubits, offer 2^n basis states and hence are inapplicable to qutrit circuits.

Software Tools for Qubit Circuits. Existing quantum circuit decomposition tools [5, 28] target only $2^n \times 2^n$ matrices for qubit computations. While algorithms for qutrit decomposition exist, an implementation for them has not yet been developed [2, 4]. Dare offers a practical implementation of the qutrit decomposition algorithm and incorporates optimizations to reduce circuit depth. Similarly, other works that directly synthesize circuits from unitaries are restricted to qubits [24]. Such techniques are not transferable, as qutrit circuit decomposition requires significant restructuring of the underlying computation. Finally, previously developed circuit rewrite methods target qubits circuits and maintain strict equality in the rewrite. In contrast, Dare generates and optimizes qutrit circuits that implement qubit computations and deploys rewrites that work with a more relaxed definition of equality.

10 CONCLUSION

Quantum computers are a revolutionary class of computing platforms capable of solving computationally hard problems but suffer from fidelity and resource availability issues. Practitioners have recently architected qutrit quantum hardware platforms that compute over $|0\rangle$, $|1\rangle$, and $|2\rangle$ states and have developed circuit depth reduction techniques that use the qutrits' higher energy $|2\rangle$ state to temporarily store information. We presented Dare, an optimizing compiler that maps qubit computations to optimized qutrit circuits at reduced depth. We anticipate that, with further investment in qutrit-based quantum hardware and new qubit equality-preserving cross-gate optimizations, we will better realize the capabilities of this class of hardware platforms.

DATA AVAILABILITY STATEMENT

The code/artifact for this work is made available with zenodo [26].

ACKNOWLEDGMENTS

We thank Pu (Luke) Yi, Yu-Neng Wang, and Mark Horowitz for their insights and feedback. Ritvik Sharma was supported by the Stanford Graduate Fellowship. Furthermore, this work was also supported in part by the Stanford Agile Hardware (AHA) Center as well as System X Alliance. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the aforementioned funding agencies.

REFERENCES

[1] Jonathan M. Baker, Casey Duckering, and Frederic T. Chong. 2020. Efficient Quantum Circuit Decompositions via Intermediate Qudits. In *2020 IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL)*. IEEE, 303–308. https://doi.org/10.1109/ISMVL49045.2020.9345604

[2] Stephen S. Bullock, Dianne P. O'Leary, and Gavin K. Brennen. 2005. Asymptotically Optimal Quantum Circuits for d-Level Systems. *Phys. Rev. Lett.* 94 (Jun 2005), 230502. Issue 23. https://doi.org/10.1103/PhysRevLett.94.230502

[3] Steven A. Cuccaro, Thomas G. Draper, Samuel A. Kutin, and Moulton David Petrie. 2004. A new quantum ripple-carry addition circuit. *Arxiv* (2004). https://doi.org/arXiv:quant-ph/0410184

[4] Yao-Min Di and Hai-Rui Wei. 2013. Synthesis of multivalued quantum logic circuits by elementary gates. *Phys. Rev. A* 87 (Jan 2013), 012325. Issue 1. https://doi.org/10.1103/PhysRevA.87.012325

[5] Byron Drury and Peter Love. 2008. Constructive Quantum Shannon Decomposition from Cartan Involutions. *J. Phys. A: Math. Theor* (2008), 1250–1262.

[6] Yue Fu, Wenquan Liu, Xiangyu Ye, Ya Wang, Chengjie Zhang, Chang-Kui Duan, Xing Rong, and Jiangfeng Du. 2022. Experimental Investigation of Quantum Correlations in a Two-Qutrit Spin System. *Phys. Rev. Lett.* 129 (Aug 2022), 100501. Issue 10. https://doi.org/10.1103/PhysRevLett.129.100501

[7] Craig Gidney. 2015. Constructing Large Controlled Nots. https://algassert.com/circuits/2015/06/05/Constructing-Large-Controlled-Nots.html.

[8] Craig Gidney. 2015. Constructing Large Increment Gates. https://algassert.com/circuits/2015/06/12/Constructing-Large-Increment-Gates.html.

[9] 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 *Proceedings of the 46th International Symposium on Computer Architecture* (Phoenix, Arizona) *(ISCA '19)*. Association for Computing Machinery, New York, NY, USA, 554–566. https://doi.org/10.1145/3307650.3322253

[10] Pavel Hrmo, Benjamin Wilhelm, Lukas Gerster, Martin W. van Mourik, Marcus Huber, Rainer Blatt, Philipp Schindler, Thomas Monz, and Martin Ringbauer. 2023. Native qudit entanglement in a trapped ion quantum processor. *Nat Commun* 14 (April 2023). https://doi.org/10.1038/s41467-023-37375-2

[11] Xiao-Min Hu, Chao Zhang, Bi-Heng Liu, Yu Cai, Xiang-Jun Ye, Yu Guo, Wen-Bo Xing, Cen-Xiao Huang, Yun-Feng Huang, Chuan-Feng Li, and Guang-Can Guo. 2020. Experimental High-Dimensional Quantum Teleportation. *Phys. Rev. Lett.* 125 (Dec 2020), 230501. Issue 23. https://doi.org/10.1103/PhysRevLett.125.230501

[12] IBM. 2018. Quantum devices and simulators. https://www.research.ibm.com/bm-q/technology/devices/.

[13] N. Khammassi, I. Ashraf, X. Fu, C. G. Almudever, and K. Bertels. 2017. QX: A High-Performance Quantum Computer Simulation Platform. In *Proceedings of the Conference on Design, Automation & Test in Europe* (Lausanne, Switzerland) *(DATE '17)*. European Design and Automation Association, Leuven, BEL, 464–469.

[14] Faisal Shah Khan and Marek Perkowski. 2005. Synthesis of Ternary Quantum Logic Circuits by Decomposition. arXiv:quant-ph/0511041 [quant-ph]

[15] Norbert M. Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Caroline Figgatt, Kevin A. Landsman, Kenneth Wright, and Christopher Monroe. 2017. Experimental comparison of two quantum computing architectures. *Proceedings of the National Academy of Sciences of the United States of America* (March 2017). https://doi.org/10.1073/pnas.1618020114

[16] Andrew Litteken, Jonathan M. Baker, and Frederic T. Chong. 2022. Communication Trade Offs in Intermediate Qudit Circuits. In *2022 IEEE 52nd International Symposium on Multiple-Valued Logic (ISMVL)*. IEEE, 43–49. https://doi.org/10.1109/ISMVL52857.2022.00014

[17] Andrew Litteken, Lennart Maximilian Seifert, Jason Chadwick, Natalia Nottingham, Frederic T. Chong, and Jonathan M. Baker. 2023. Qompress: Efficient Compilation for Ququarts Exploiting Partial and Mixed Radix Operations for Communication Reduction. *ASPLOS* (2023), 646–659. https://doi.org/10.1145/3575693.3575726

[18] Andrew Litteken, Lennart Maximilian Seifert, Jason D. Chadwick, Natalia Nottingham, Tanay Roy, Ziqian Li, David Schuster, Frederic T. Chong, and Jonathan M. Baker. 2023. Dancing the Quantum Waltz: Compiling Three-Qubit Gates on Four Level Architectures. In *Proceedings of the 50th Annual International Symposium on Computer Architecture* (Orlando, FL, USA) *(ISCA '23)*. Association for Computing Machinery, New York, NY, USA, Article 71, 14 pages. https://doi.org/10.1145/3579371.3589106

[19] Wen-Qiang Liu and Hai-Rui Wei. 2020. Optimal synthesis of the Fredkin gate in a multilevel system. *New Journal of Physics* 22, 6 (jun 2020), 063026. https://doi.org/10.1088/1367-2630/ab8e13

[20] Yi-Han Luo, Han-Sen Zhong, Manuel Erhard, Xi-Lin Wang, Li-Chao Peng, Mario Krenn, Xiao Jiang, Li Li, Nai-Le Liu, Chao-Yang Lu, Anton Zeilinger, and Jian-Wei Pan. 2019. Quantum Teleportation in High Dimensions. *Phys. Rev. Lett.* 123 (Aug 2019), 070505. Issue 7. https://doi.org/10.1103/PhysRevLett.123.070505

[21] Daniel Miller, Timo Holz, Hermann Kampermann, and Dagmar Bruß. 2018. Propagation of generalized Pauli errors in qudit Clifford circuits. *Phys. Rev. A* 98 (Nov 2018), 052316. Issue 5. https://doi.org/10.1103/PhysRevA.98.052316

[22] Yunseong Nam, Neil J. Ross, Yuan Su, Andrew M. Childs, and Dmitri Maslov. 2018. Automated optimization of large quantum circuits with continuous parameters. *npj Quantum Information* 5 (May 2018).

[23] Michael A. Nielsen and Isaac L. Chuang. 2010. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511976667

[24] Tirthak Patel, Ed Younis, Costin Iancu, Wibe de Jong, and Devesh Tiwari. 2022. QUEST: Systematically Approximating Quantum Circuits for Higher Output Fidelity. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (Lausanne, Switzerland) *(ASPLOS '22)*. Association for Computing Machinery, New York, NY, USA, 514–528. https://doi.org/10.1145/3503222.3507739

[25] Martin Ringbauer, Michael Meth, Lukas Postler, Roman Stricker, Rainer Blatt, Philipp Schindler, and Thomas Monz. 2022. A universal qudit quantum processor with trapped ions. *Nat. Phy.* 5 (May 2022). https://doi.org/10.1038/s41567-022-01658-0

[26] Sara Achour Ritvik Sharma. 2024. Dare Compiler. https://doi.org/10.5281/zenodo.10807175

[27] Tanay Roy, Ziqian Li, Eliot Kapit, and DavidI. Schuster. 2023. Two-Qutrit Quantum Algorithms on a Programmable Superconducting Processor. *Phys. Rev. Appl.* 19 (Jun 2023), 064024. Issue 6. https://doi.org/10.1103/PhysRevApplied.19.064024

[28] Vivek V. Shende, Stephen S. Bullock, and Igor L. Markov. 2005. Synthesis of Quantum Logic Circuits. In *Proceedings of the 2005 Asia and South Pacific Design Automation Conference* (Shanghai, China) *(ASP-DAC '05)*. Association for Computing Machinery, New York, NY, USA, 272–275. https://doi.org/10.1145/1120725.1120847

[29] B.D. Sutton. 2009. Computing the complete CS decomposition. *Numerical Analysis* 50 (2009), 33–65. https://doi.org/10.1007/s11075-008-9215-6

[30] Yasuhiro Takahashi, Seiichiro Tani, and Noboru Kunihiro. 2010. Quantum Addition Circuits and Unbounded Fan-Out. *Quantum Info. Comput.* 10, 9 (sep 2010), 872–890.

[31] Mingkuan Xu, Zikun Li, Oded Padon, Sina Lin, Jessica Pointing, Auguste Hirth, Henry Ma, Jens Palsberg, Alex Aiken, Umut A. Acar, and Zhihao Jia. 2022. Quartz: Superoptimization of Quantum Circuits. In *Proceedings of the 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation* (San Diego, CA, USA) *(PLDI 2022)*. Association for Computing Machinery, New York, NY, USA, 625–640. https://doi.org/10.1145/3519939.3523433

Received 2023-11-16; accepted 2024-03-31