Editors' Suggestion

Efficient implementation of multicontrolled quantum gates

Ben Zindorf* and Sougato Bose
Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, United Kingdom

(Received 21 August 2024; revised 11 September 2025; accepted 15 September 2025; published 9 October 2025)

We present an implementation of multicontrolled (MC) quantum gates that provides significant reductions of cost compared to state-of-the-art methods. The operator applied on the target qubit is a unitary, special unitary, or the Pauli X operator (multicontrolled Toffoli), and requires one clean ancilla, no ancilla, or one dirty ancilla, respectively. We generalize our methods for any number of target qubits, and provide further cost reductions if additional ancilla qubits are available. For each type of MC gate, we provide implementations for unrestricted (all-to-all) connectivity and for linear-nearest-neighbor (LNN) connectivity. All of the methods use a linear cost of gates from the Clifford+T (fault-tolerant) set. In the context of LNN architecture, the cost and depth of our circuits scale linearly irrespective of the position of the qubits on which the gate is applied. Our methods directly improve the compilation process of many quantum algorithms, providing optimized circuits. Given the scale of our improvements, for example, the quadratic-to-linear controlled-NOT count for LNN, they will naturally result in a large reduction of errors.

DOI: 10.1103/8blx-nfcr

I. INTRODUCTION

Multicontrolled single- or multiple-target quantum gates are pivotal ingredients for quantum computation. These gates can be seen as the quantum counterparts of conditional if-then-else statements [1,2], which are frequently used in classical computation, making them a convenient logical building block for quantum algorithm design. Thus, one may expect these elementary gates [3] to appear in algorithms that will be developed in the future, in addition to existing ones. For example, such gates are shown to be very useful in circuits for quantum chemistry, e.g., particle-number-conserving Hamiltonians [4], for constructing a circuit implementing a quantum neuron [5], in the quantum alternating operator ansatz algorithm [6], quantum circuits for isometries [7,8], port-based teleportation [9], distributed quantum computation [10], constructing a quantum RAM [11], reversible logic [12], in the circuit for Grover's quantum search algorithm [13], and generally for quantum arithmetics [14], such as for the arithmetic part of Shor's algorithm [15]. Moreover, a constructive method for implementing arbitrary unitaries, which is, in fact, used for the proof of the sufficiency of fundamental gate sets for universal quantum computation, uses such gates [16,17]. Of course, that constructive method may not be the most efficient, but nonetheless emphasizes that such gates are at the heart of quantum computation.

Unfortunately, while it may be possible to apply multiqubit gates directly in some quantum architectures [18–20], multicontrolled (MC) gates with more than one control qubit are not available for direct implementation in most existing quantum hardware and must be decomposed into smaller available gates. In most near-term NISQ (noisy intermediate-scale quantum) devices, tunable single-qubit gates [e.g., $R_z(\theta)$ and $R_y(\theta)$] are used along with some fixed two-qubit gates [e.g., controlled-NOT (CNOT), controlled-Z (CZ), and echoed cross-resonance (ECR) gates [21–23]] as the fundamental building blocks, such that the latter gates are usually characterized by a higher error rate and may require longer execution time, making them a more "expensive" resource than the former gates. As in most cases, these two-qubit gates are equivalent to a CNOT up to single-qubit rotations, it is common to design a circuit using CNOT as the two-qubit gate—a convenient choice due to its similarities to the classical XOR logic gate. As the errors introduced by each CNOT gate accumulate and propagate, when used to construct a large circuit, minimizing the gate count is an important task, as it directly improves the fidelity of the circuit. Then, a circuit designed using these gates can be directly converted to a circuit that uses another available two-qubit gate by

*Contact author: ben.zindorf.19@ucl.ac.uk

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

only increasing the number of the cheaper single-qubit gates.

The CNOT is also part of the Clifford+T gate set (along with T, H, and S) [24–26], which is a prime candidate for the upcoming error-corrected fault-tolerant quantum computation. However, in most error-correction codes, the single-qubit T gate is the most expensive resource, followed by the CNOT. A decomposition of MC gates that minimizes both CNOT and T gates will therefore be beneficial both in the present and in the future, directly improving the fidelity of any circuit that is built using these logical building blocks. Other error-correction codes exist [27,28], in which the most expensive Clifford+T gate is the Hadamard (H), followed by the CNOT, and therefore a low count of H gates is also beneficial. The Clifford+T gate set is universal and therefore can be used to approximately implement arbitrary single-qubit gates, such that greater accuracy requires a higher number of T and H gates [24]. The use of arbitrary single-qubit gates should therefore be minimized in the fault-tolerant regime. There are other universal sets which are widely used, such as the Toffoli+H and the NCV library (X, CNOT, $C\sqrt{X}$, $C\sqrt{X}^\dagger$) [3,29]. As any of the mentioned gate sets can be used to implement the Toffoli, it may be a good strategy to start by decomposing an MC gate using Toffoli gates and then further decomposing these into smaller gates, thus producing a circuit that can be easily converted to the gate set of choice.

In addition to the limited available gates, another obstacle is the restricted connectivity between qubits, which is common to many architectures. While some allow for all-to-all connectivity [30,31], others are restricted to neighboring qubits in two-dimensional (2D) or one-dimensional (1D) arrays. Since there are many options for the arrangement of 2D arrays, such as the square lattice used by Google, and the heavy-hex used by IBM, it can be useful to find an efficient implementation for the highly restrictive 1D linear-nearest-neighbor (LNN) connectivity, as it can be used directly in some technologies, and can also be mapped to many 2D arrays with no or low overhead. On the other extreme, while decompositions designed for all-to-all connectivity can be applied in some architectures, these are especially useful as a conceptual unrestricted playground, which can be a good first step before attempting to find decompositions in more restrictive connectivities.

Although any all-to-all implementation can be mapped to LNN using SWAP gates, doing so can drastically increase the cost, e.g., increasing the CNOT count of a circuit from linear to quadratic scaling in the number of qubits [32–38]. To this day, this was true for the implementations of MC gates as well—while all-to-all decompositions require a linear CNOT count using a limited number of ancilla qubits [3,7,39–49], the best-known upper bound of CNOT gates in LNN connectivity is quadratic [29,50–55] due to the large number of SWAP gates. Without ancilla, some MC (with a target operator $U \in U(2)\setminus SU(2)$) gate implementations in all-to-all connectivity require a quadratic CNOT count [48,50,51,56–58], while others allow for a linear count [59], such that in both cases the count of arbitrary single-qubit rotations scales at least linearly and these gates require exponential accuracy. The decomposition of MC gates with a small number of controls under various connectivity and ancillary constraints has been addressed as well [60–67].

In addition to gate count, an important metric is the depth of the circuit, i.e., the number of time slots holding several gates which can be executed in parallel. Minimizing the depth results in a shorter run-time, which reduces decoherence-induced errors. Some approaches aim to minimize the depth of MC gates [68,69]; however, this comes with a trade-off, increasing either the gate count or the number of ancillae used. Low-depth implementations in all-to-all connectivity may rely on the ability to execute many two-qubit gates in parallel. However, the number of gates that can be efficiently applied in parallel may be limited in some all-to-all connected devices [70], and such implementations may increase the gate count overhead when mapped to more restricted connectivities. As the depth of each gate type is no larger than its count, low-gate-count implementations may provide the lowest depth as well, in case efficient parallel gate execution is not available or is restricted to a constant number of gates.

The use of logical building blocks when constructing an algorithm can be preferable to constructing the corresponding quantum circuit using the smallest available gates directly. This may be seen as analogous to developing a classical software using assembly—while this might provide more efficient implementations than the use of built-in functions in high-level programming languages like C++ or PYTHON, most would choose the latter for complicated tasks. A compilation step is then required to convert these functions into assembly code, such that an improved compilation of a frequently used function can improve the efficiency of the entire program. On the other hand, one may wish to define the entire quantum program as one large unitary matrix, then apply a compilation step to decompose the entire operator to the smallest gates. Although various methods exist [71–73], the resulting gate-count upper bound is exponential, due to the number of parameters in the unitary—making it implausible to achieve any improvement over a classical computer.

In this paper, we focus on minimizing the gate count of CNOT, T, and H (Clifford+T) required for the implementation of various types of multicontrolled gates, which, as discussed above, is beneficial to both fault-tolerant as well as near-term applications. Specifically we address the cases in which the target operator is in $SU(2)$, $U(2)$, or the

Pauli X. We wish to make minimal assumptions regarding the hardware used and the available resources in order to cover as many use-cases as possible. For each type of gate, we use the minimal number of ancilla qubits allowing for a linear scaling of Clifford+T gates and for a constant number of single-qubit arbitrary rotations. The type of ancilla qubit is important as well, such that a "clean" ancilla that must be kept in a specific state is a more expensive resource than a borrowed "dirty" ancilla that can be any qubit that is unused at the time of the MC gate execution. We implement each gate in all-to-all and LNN connectivity. When the depth can be reduced without increasing the gate count, we apply these reductions as well. Finally, while requiring the minimal number of ancilla qubits, we provide gate count reductions, which can be achieved in the case that the MC gate is applied only to a subset of all available qubits, and the rest can be borrowed as ancillae. In all-to-all connectivity we achieve a CNOT count reduction of up to 62.5% compared to the state-of-the-art value, and for the multicontrolled X case we reduce the Toffoli count by 50%. In LNN connectivity, our methods provide a linear upper bound of the CNOT gate count for any qubit ordering, a *quadratic* reduction compared to state-of-the-art methods.

A. Notation

In this paper, we use the notation defined in Ref. [66]. We list some of this notation here for convenience.

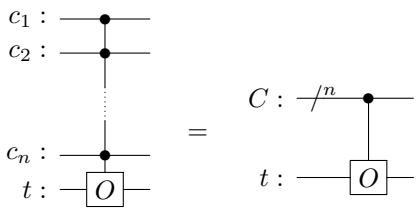
Any single-qubit $SU(2)$ operator can be represented as a rotation by angle λ about an axis $\hat{v}$ as $R_{\hat{v}}(\lambda)$ [16]. We use the notation $\Pi(\hat{v}) := iR_{\hat{v}}(\pi)$ to describe a Hermitian gate applying a π rotation about an arbitrary axis. The three-dimensional rotation matrix by angle θ about the vector $\hat{v}$ is marked as $\hat{R}_{\hat{v}}(\theta)$.

These items of notation have been used to describe the decomposition of any $SU(2)$ operator in terms of two Π gates in Ref. [66, Lemma 1] as follows.

Lemma 1. Any $R_{\hat{v}}(\lambda) \in SU(2)$ operator can be implemented as $\Pi(\hat{v}_2)\Pi(\hat{v}_1)$ such that $\hat{v}_1$ can be chosen as any unit vector perpendicular to $\hat{v}$, and $\hat{v}_2 = \hat{R}_{\hat{v}}(\lambda/2)\hat{v}_1$ with $\lambda/2 \in (-\pi, \pi]$.

This lemma can be seen as a generalization of known identities of the Pauli matrices $X = \Pi(\hat{x})$, $Y = \Pi(\hat{y})$, and $Z = \Pi(\hat{z})$. For example, since the unit vector $\hat{y}$ can be obtained by rotating $\hat{x}$ about $\hat{z}$, as $\hat{y} = \hat{R}_{\hat{z}}(\pi/2)\hat{x}$, and $\hat{x} \perp \hat{z}$, we get $\Pi(\hat{y})\Pi(\hat{x}) = R_{\hat{z}}(\pi)$, i.e., $YX = -iZ$.

When referring to an operator O applied on a set of qubits $\tau = \{t_1, t_2, \ldots, t_m\}$, we use the notation $[O]_\tau$. If the operator is controlled by a qubit set $C = \{c_1, c_2, \ldots, c_n\}$ and applied on a qubit t, we write $[O]_t^{\{c_1,c_2,\ldots,c_n\}}$ or $[O]_t^C$, and use one of the following for gate notation.



Two lemmas in Ref. [66, Lemmas 4 and 13] provide methods that allow one to transform the rotation axis of MC gates. We repeat those here for convenience.

The set $M(\hat{v}_1, \hat{v}_2)$ is defined for any two unit vectors $\hat{v}_1$ and $\hat{v}_2$ such that $\hat{v}_M \in M(\hat{v}_1, \hat{v}_2)$ if and only if $\hat{v}_2 = \hat{R}_{\hat{v}_M}(\pi)\hat{v}_1$. In other words, $\hat{v}_M$ can be chosen as a unit vector that is in the "middle" between $\hat{v}_1$ and $\hat{v}_2$. A well-known special case of Lemma 2 is achieved by noting that $H = \Pi(\hat{h})$, with $\hat{h} \in M(\hat{x}, \hat{z})$.

Lemma 2. $[e^{i\psi}R_{\hat{v}}(\lambda)]_t^C = [\Pi(\hat{v}_M)]_t[e^{i\psi}R_{\hat{v}'}(\lambda)]_t^C[\Pi(\hat{v}_M)]_t$ for any angles λ and ψ, unit vectors $\hat{v}$ and $\hat{v}'$, and $\hat{v}_M \in M(\hat{v}, \hat{v}')$.

Lemma 3. If $\hat{v} = \hat{R}_{\hat{\sigma}}(\phi)\hat{\tau}$ for any unit vectors $\hat{\tau}$ and $\hat{\sigma}$ satisfying $\hat{\tau} \perp \hat{\sigma}$, then $[\Pi(\hat{v})]_t^C = [R_{\hat{\sigma}}(\phi)]_t[\Pi(\hat{\tau})]_t^C[R_{\hat{\sigma}}(-\phi)]_t$.

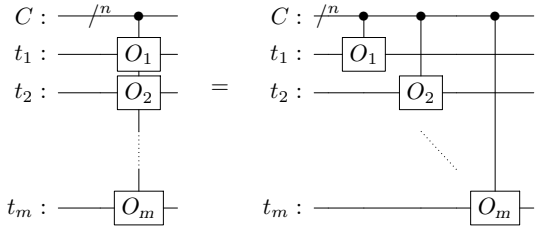
The phase gate is defined as

$$P(\lambda) := e^{i\lambda/2}R_{\hat{z}}(\lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix}.$$

We use the standard notation for two specific phase gates:

$$S := P(\pi/2) = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad \text{and} \quad T := P(\pi/4).$$

The notation $C[a : b] := \{c_a, c_{a+1}, \ldots, c_b\}$ is used to define a subset and $C[a : a] = \{c_a\}$ is written as $C[a]$. When concatenating two sets of qubits, we simply use $\{C, \tau\} := \{c_1, \ldots, c_n, t_1, \ldots, t_m\}$. For a controlled phase gate, the phase is applied if and only if all the control qubits and the target qubit are in the state $|1\rangle$, and therefore it is not required to specify the target. For example, the gate $[Z]_t^C$ can also be written as $[Z]^{\{C,t\}}$. We refer to any gate diagonal in the computational basis as a *relative-phase gate*, usually marked as $[\Delta]$. These gates can be decomposed using only multicontrolled (or single-qubit) phase gates. Finally, a multicontrolled multitarget (MCMT) gate $\prod_{j=1}^m [O]_{t_j}^C$ is marked as follows.



II. ALL-TO-ALL CONNECTIVITY

We cover the decompositions of three types of MC gates in all-to-all (ATA) connectivity, with the aim of minimizing the gate count of the Clifford+T gates CNOT, H, and T. We start by providing a structure that can implement any multicontrolled $SU(2)$ gate with one or more targets and no ancilla, then using this to implement the multicontrolled Pauli X gate with one dirty ancilla qubit. Finally, using a new trick, we show that the multicontrolled $SU(2)$ gate with two targets can be used along with one clean ancilla to implement any multicontrolled $U(2)$ gate. As all three gate types are based on the implementation of multicontrolled $SU(2)$, their gate counts are the same up to a small constant. The gate count provides an upper bound to the resulting circuit depth; however, we show that the depth can be lowered in the case that parallel gate execution is allowed.

A. Multicontrolled $SU(2)$

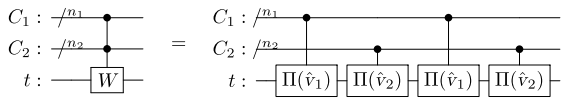
In this section, we focus on a cost-efficient implementation of multicontrolled $SU(2)$ gates (MCSU2) with a single or multiple targets and no ancilla qubit. We start by decomposing the MCSU2 using four large many-qubit diagonal gates in Sec. II A 1. Then in Sec. II A 2 we decompose these many-qubit gates using three-qubit gates and finally to single- or two-qubit Clifford+T gates. We extend our methods to allow for an efficient multicontrolled multitarget $SU(2)$ decomposition in Appendix C. For those who prefer a low T depth, we provide some alternative implementations in Appendix B, which allows one to reduce the depth of the T gate while only increasing the CNOT depth, or to further reduce the T depth by increasing the CNOT count as well.

Although our methods do not require any ancilla qubits, we recognize that in some cases the MCSU2 gate may not apply on all available qubits, in which case idle qubits can be used as dirty ancilla to reduce the circuit cost. In Appendix E we address these cost reductions so that each additional ancilla contributes to the cost reduction, until reaching the lowest known gate count achieved for the multicontrolled X gate in Ref. [74] using $O(n)$ dirty ancilla, thus finding a bridge between no-ancilla and $O(n)$ dirty ancilla decompositions.

1. MCSU2 *macrostructure*

We provide a decomposition of any MC gate $[W]_t^C$ such that $W \in SU(2)$. Any such W can be expressed as a rotation about an axis $\hat{v}$ by an angle λ as $W = R_{\hat{v}}(\lambda)$. The decomposition requires eight single-qubit gates from $\{R_{\hat{x}}, R_{\hat{z}}\}$, in addition to four large relative-phase gates.

As a first step, we provide the following decomposition.

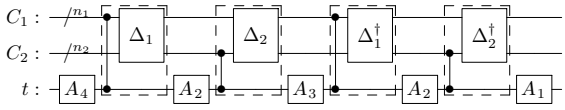


(1)

Lemma 4. Any $[R_{\hat{v}}(\lambda)]_t^C$ gate with $R_{\hat{v}}(\lambda) \in SU(2)$ can be implemented as $[\Pi(\hat{v}_2)]_t^{C_2}[\Pi(\hat{v}_1)]_t^{C_1}[\Pi(\hat{v}_2)]_t^{C_2}[\Pi(\hat{v}_1)]_t^{C_1}$, such that $C_1 \cup C_2 = C$. And $\hat{v}_1$ can be chosen as any unit vector perpendicular to $\hat{v}$, and $\hat{v}_2 = \hat{R}_{\hat{v}}(\lambda/4)\hat{v}_1$.

Proof. Since $\Pi(\cdot)^2 = I$ for any axis, if one or more of the control sets C_1 and C_2 is not in the computational basis state $|11\ldots1\rangle$, then the circuit applies identity to the target qubit t. Otherwise the circuit applies $(\Pi(\hat{v}_2)\Pi(\hat{v}_1))^2 = R_{\hat{v}}(\lambda)$ according to Lemma 1. ■

We could use Lemma 3 to decompose the multicontrolled Π gates in circuit (1) as four multicontrolled-Toffoli (MCT) [also known as multicontrolled-X (MCX)] gates and single-qubit rotations, achieving a structure similar to the one used in Ref. [39] to implement a subset of MCSU2, while ours can implement any MCSU2. Following their analysis, this step alone provides a reduction of CNOT count from $20n$ to $16n$. However, we find it useful to allow for an implementation of these MCX gates as multicontrolled-Z (MCZ) gates with an added relative phase, as in circuit (2), resulting in a reduced Toffoli count, which provides a further reduction of CNOT count.



(2)

To minimize the number of arbitrary single-qubit rotations from $\{R_{\hat{x}}, R_{\hat{z}}\}$, it is beneficial to use Lemma 4 in order to implement a multicontrolled $SU(2)$ rotation about the $\hat{x}$ axis and then transform the axis using Lemma 2. Identity gates in the form of relative-phase gates $[\Delta]_C$, and their inverse $[\Delta^\dagger]_C$, can be added to produce the following decomposition of $[R_{\hat{v}}(\lambda)]_t^{\{C_1,C_2\}}$. The additional $[\Delta]_C$ gates can be set as any relative-phase gates. Decomposing each MCZ gate, combined with its corresponding $[\Delta]_C$ in terms of basic gates, allows for an efficient implementation, as we show in Sec. II A 2.

Lemma 5. Any $[R_{\hat{v}}(\lambda)]_t^C$ gate with $R_{\hat{v}}(\lambda) \in SU(2)$ can be implemented with eight gates from $\{R_{\hat{x}}, R_{\hat{z}}\}$, two $[Z]_t^{C_1}$, and two $[Z]_t^{C_2}$ gates as circuit (2) such that $A_2 = R_{\hat{x}}(-\lambda/4)$, $A_3 = R_{\hat{x}}(\lambda/4)$, $A_4 = R_{\hat{z}}(\theta_2)R_{\hat{x}}(\theta_1)$, $A_1 = A_4{}^\dagger A_3$, and $C_1 \cup C_2 = C$. The $[\Delta]_C$ gates apply a relative phase, and the angles θ_1 and θ_2 are Euler angles of $\Pi(\hat{v}_M)$ with $\hat{v}_M \in M(\hat{v}, \hat{x})$.

Proof. Using Lemma 2 with $\psi = 0$, we can write

$$[R_{\hat{v}}(\lambda)]_t^C = [\Pi(\hat{v}_M)]_t[R_{\hat{x}}(\lambda)]_t^C[\Pi(\hat{v}_M)]_t.$$

The gate $\Pi(\hat{v}_M)$ can be decomposed up to a global phase using the XZX Euler angles as $R_{\hat{x}}(\theta_3)R_{\hat{z}}(\theta_2)R_{\hat{x}}(\theta_1) = R_{\hat{x}}(\theta_3)A_4$, and, since $\Pi(\hat{v}_M)$ is Hermitian, it can be implemented as $A_4^\dagger R_{\hat{x}}(-\theta_3)$ as well. Substituting and applying cancellations provides

$$[R_{\hat{v}}(\lambda)]_t^C = [A_4^\dagger]_t[R_{\hat{x}}(\lambda)]_t^C[A_4]_t.$$

From Lemma 4 with $\hat{v}_1 = \hat{z}$ and $\hat{v}_2 = \hat{R}_{\hat{x}}(\lambda/4)\hat{z}$ we get

$$[R_{\hat{x}}(\lambda)]_t^C = [\Pi(\hat{v}_2)]_t^{C_2}[Z]_t^{C_1}[\Pi(\hat{v}_2)]_t^{C_2}[Z]_t^{C_1}.$$

Finally, each $[\Pi(\hat{v}_2)]_t^{C_2}$ gate can be implemented as

$$\begin{aligned}[\Pi(\hat{v}_2)]_t^{C_2} &= [R_{\hat{x}}(\lambda/4)]_t[Z]_t^{C_2}[R_{\hat{x}}(-\lambda/4)]_t \\ &= [A_3]_t[Z]_t^{C_2}[A_2]_t\end{aligned}$$

according to Lemma 3. ■

2. MCSU2 *decomposition*

In this section, we provide a decomposition of $[\Delta_1]_C[Z]_t^{C_1}$ and $[\Delta_2]_C[Z]_t^{C_2}$ and their inverse [the boxed gates in circuit (2)] in terms of Clifford+T gates, as required for the implementation described in Lemma 5. The following lemma will be used to develop our structure.

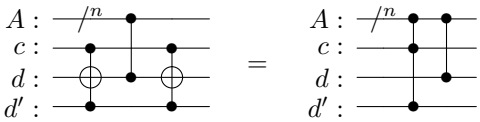
Lemma 6. For two sets of qubits A and B and a qubit $d \notin A \cup B$, the following holds: $[\Pi(\hat{v}_2)]_d^B[\Pi(\hat{v}_1)]_d^A[\Pi(\hat{v}_2)]_d^B = [Z]^{\{A,B\}}[\Pi(\hat{v}_1)]_d^A$ if the unit vectors $\hat{v}_1$ and $\hat{v}_2$ are perpendicular.

Proof. For any choice of two perpendicular vectors $\hat{v}_1 \perp \hat{v}_2$, a unit vector $\hat{v}$ perpendicular to both can be defined as $\hat{v} = \hat{v}_1 \times \hat{v}_2$ such that $\hat{v}_2 = \hat{R}_{\hat{v}}(\pi/2)\hat{v}_1$. Therefore, according to Lemma 4,

$$[\Pi(\hat{v}_2)]_d^B[\Pi(\hat{v}_1)]_d^A[\Pi(\hat{v}_2)]_d^B[\Pi(\hat{v}_1)]_d^A = [R_{\hat{v}}(2\pi)]_d^{\{A,B\}}.$$

We note that for any unit vector $\hat{v}$, $[R_{\hat{v}}(2\pi)]_d^{\{A,B\}} = [-I]_d^{\{A,B\}}$. This operator merely applies a phase of -1 if all qubits in the set $A \cup B$ are in the computational basis state $|1\rangle$, and does not change the state of the target qubit d. Therefore, it is equivalent to applying a multicontrolled Z gate on the set $A \cup B$, and we can write $[-I]_d^{\{A,B\}} = [Z]^{\{A,B\}}$, as mentioned in Ref. [40]. By substituting this into the equation above and applying the Hermitian gate $[\Pi(\hat{v}_1)]_d^A$ on the right-hand side, we get the required equation. ■

For our construction, we use Lemma 6 with $\hat{v}_1 = \hat{z}$, $\hat{v}_2 = \hat{x}$, and set $B = \{c, d'\}$, which results in $[X]_d^{\{c,d'\}}[Z]_d^A[X]_d^{\{c,d'\}} = [Z]_{d'}^{\{A,c\}}[Z]_d^A$ as follows.

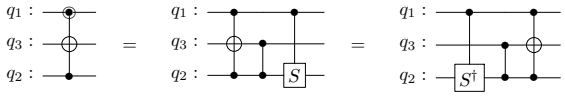


(3)

We introduce the following notation:

$$\{Z\}_{C,D}^m = \prod_{j=2}^{m}[Z]_{d_{j-1}}^{C[1:j]},$$

where $C = \{c_1, c_2, \ldots, c_n\}$ and $D = \{d_1, d_2, \ldots, d_{n-1}\}$ are two qubit sets, and m can be chosen such that $2 \le m \le n$. In addition, we use the notation $[X_\Delta]_{q_3}^{\{q_1,q_2\}}$ for a Hermitian *relative-phase Toffoli* gate (a Toffoli gate, either followed or preceded by a relative-phase gate) which satisfies the following.



(4)

This gate notation allows us to distinguish between the control qubits. We use a symbol which combines the $|0\rangle$-controlled and $|1\rangle$-controlled symbols, noting that, for each of these basis states of q_1, a different controlled operator is effectively applied on q_2 and q_3.

We now show that two $[X_\Delta]$ gates can be used to transform $\{Z\}_{C,D}^{m-1}$ to $\{Z\}_{C,D}^m$.

Lemma 7. $\{Z\}_{C,D}^m = [X_\Delta]_{d_{m-2}}^{\{c_m,d_{m-1}\}}\{Z\}_{C,D}^{m-1}[X_\Delta]_{d_{m-2}}^{\{c_m,d_{m-1}\}}$ for $3 \le m \le |C|$ and $|D| \ge m-1$.

Proof. We prove for the following equivalent equation, in which $[X_\Delta]$ are replaced with Toffoli gates:

$$\{Z\}_{C,D}^m = [X]_{d_{m-2}}^{\{c_m,d_{m-1}\}}\{Z\}_{C,D}^{m-1}[X]_{d_{m-2}}^{\{c_m,d_{m-1}\}},$$

where the equivalence can be easily verified by substituting each $[X_\Delta]$ gate with its definition, which places all of the relative-phase gates at the center, then these are commuted and cancelled out.

For $m = 3$:

$$\begin{aligned}&[X]_{d_1}^{\{c_3,d_2\}}\{Z\}_{C,D}^2[X]_{d_1}^{\{c_3,d_2\}} \\ &\quad= [X]_{d_1}^{\{c_3,d_2\}}[Z]_{d_1}^{\{c_1,c_2\}}[X]_{d_1}^{\{c_3,d_2\}} \\ &\quad\overset{*}{=} [Z]_{d_1}^{\{c_1,c_2\}}[Z]_{d_2}^{\{c_1,c_2,c_3\}} = \{Z\}_{C,D}^3,\end{aligned}$$

where $*$ is according to Lemma 6.

For $m \geq 4$, by definition:

$$\{Z\}_{C,D}^{m-1} = [Z]_{d_{m-2}}^{C[1:m-1]}\{Z\}_{C,D}^{m-2}.$$

Since c_m, d_{m-1}, and d_{m-2} are not accessed in $\{Z\}_{C,D}^{m-2}$, this gate commutes with $[X]_{d_{m-2}}^{\{c_m,d_{m-1}\}}$. Therefore

$$[X]_{d_{m-2}}^{\{c_m,d_{m-1}\}}\{Z\}_{C,D}^{m-1}[X]_{d_{m-2}}^{\{c_m,d_{m-1}\}} = [X]_{d_{m-2}}^{\{c_m,d_{m-1}\}}[Z]_{d_{m-2}}^{C[1:m-1]}[X]_{d_{m-2}}^{\{c_m,d_{m-1}\}}\{Z\}_{C,D}^{m-2}.$$

According to Lemma 6, and since $C[1:m-1]\cup\{c_m\} = C[1:m]$

$$[X]_{d_{m-2}}^{\{c_m,d_{m-1}\}}[Z]_{d_{m-2}}^{C[1:m-1]}[X]_{d_{m-2}}^{\{c_m,d_{m-1}\}} = [Z]_{d_{m-1}}^{C[1:m]}[Z]_{d_{m-2}}^{C[1:m-1]}.$$

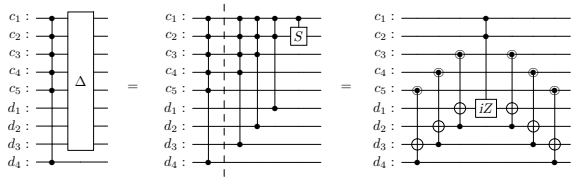
Finally, we can substitute and use

$$[Z]_{d_{m-1}}^{C[1:m]}[Z]_{d_{m-2}}^{C[1:m-1]}\{Z\}_{C,D}^{m-2} = \{Z\}_{C,D}^{m}. \quad \blacksquare$$

The following lemma is simply achieved by repeatedly applying the recursive formula presented in Lemma 7, and, as a final step, applying $[S]_{c_2}^{c_1}$ to both sides. This added gate commutes with any other gate in the circuit and transforms $[Z]_{d_1}^{\{c_1,c_2\}}$ to $[iZ]_{d_1}^{\{c_1,c_2\}}$.

Lemma 8. $\{Z\}_{C,D}^{n}[S]_{c_2}^{c_1} = \left(\prod_{j=3}^{n}[X_\Delta]_{d_{j-2}}^{\{c_j,d_{j-1}\}}\right)^\dagger [iZ]_{d_1}^{\{c_1,c_2\}}\left(\prod_{j=3}^{n}[X_\Delta]_{d_{j-2}}^{\{c_j,d_{j-1}\}}\right)$ for $n \geq 3$, $|C| = n$, and $|D| = n-1$.

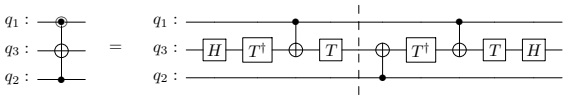
Circuit (5) describes the decomposition of $\{Z\}_{C,D}^{5}[S]_{c_2}^{c_1}$ in terms of $[X_\Delta]_{q_3}^{\{q_1,q_2\}}$ and $[iZ]_{q_3}^{\{q_1,q_2\}}$ gates, or $[X_\Delta]$ and $[iZ]$ for short. The Δ gate in the leftmost circuit holds all diagonal gates which do not apply on d_4 in the middle circuit.



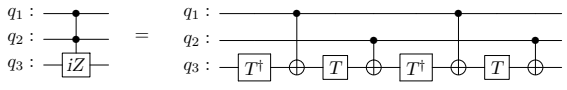
(5)

This structure requires $2n-4$ $[X_\Delta]$ gates and one $[iZ]$ gate. It can be applied using the Toffoli+H gate set by replacing $[X_\Delta] \to [X]$ and $[iZ] \to [Z]$, providing an MCSU2 implementation using $4n + O(1)$ Toffoli gates when applied in circuit (2), in addition to $O(1)$ arbitrary single-qubit gates. Compared to the implementation of multicontrolled $R_{x/y/z} \in SU(2)$ gates in Ref. [39], this construction uses half the number of Toffoli gates while implementing any MCSU2.

It can be noted that the structure in circuit (5) resembles in shape the first half of the structure used in Ref. [39] for a similar purpose in the MCSU2 construction. This Toffoli-based structure may be applied directly in some architectures [20], or further decomposed in terms of single- or two-qubit gates from gate sets such as NCV. As we focus on Clifford+T implementations, we will decompose these gates using T, $T^\dagger$, H, and CNOT gates. We use the following decompositions for $[X_\Delta]_{q_3}^{\{q_1,q_2\}}$ [74] and $[iZ]_{q_3}^{\{q_1,q_2\}}$.



(6)



(7)

Each $[X_\Delta]$ requires two H, three CNOT, and four T gates. It can be seen that, when applied as part of Lemma 8, the gates to the left of the barrier in circuit (6) can commute with any gate that is executed before. This, along with additional similar commutations, allows for depth reductions, as shown in Appendix A. A simple summation provides the following lemma.

Lemma 9. A $\{Z\}_{C,D}^{n}[S]_{c_2}^{c_1}$ gate with $n \geq 3$, $|C| = n$, and $|D| = n-1$ can be implemented with
CNOT cost $6n-8$ and depth $4n-2$,
T cost $8n-12$ and depth $4n$,
H cost $4n-8$ and depth $2n-2$.

We wish to use this to decompose the structure described in Lemma 5, which implements any $[R_{\hat{v}}(\lambda)]_t^C$ (MCSU2) gate. We choose $[\Delta_1]_C = [S]^{C_1[1:2]}\{Z\}_{C_1,C_2'}^{n_1-1}$ and $[\Delta_2]_C = [S]^{C_2[1:2]}\{Z\}_{C_2,C_1'}^{n_2-1}$ as relative-phase gates applied on the set C of size n. For this choice, we get

$$[\Delta_1]_C[Z]_t^{C_1} = [S]^{C_1[1:2]}\{Z\}_{C_1,\{C_2',t\}}^{n_1}$$

and

$$[\Delta_2]_C[Z]_t^{C_2} = [S]^{C_2[1:2]}\{Z\}_{C_2,\{C_1',t\}}^{n_2}.$$

The inverted versions of these gates are implemented in a similar way.

The subsets C_1, C_2, C_1', and C_2' are of size n_1, n_2, n_1', and n_2', respectively, and satisfy $C_1 \cup C_2 = C$, $C_1' \in C_1$, $C_2' \in C_2$, $n_1' = n_2 - 2$, $n_2' = n_1 - 2$, and $n_1 + n_2 = n$. We can set $n_1 = \lfloor n/2 \rfloor$ and $n_2 = n - n_1 = \lceil n/2 \rceil$.

These structures can be used to decompose circuit (2). Each cost and depth from Lemma 9 can be written

as $an + b$, and therefore the counterparts for MCSU2 are equal to $2(an_1 + b + an_2 + b) = 2an + 4b$. As shown in Appendix A, the sets C_1' and C_2' can be chosen such that additional depth reductions are achieved. The following theorem presents the total costs and depths.

Theorem 1. Any multicontrolled $SU(2)$ gate with $n \geq 6$ controls can be implemented without ancilla using eight gates from $\{R_{\hat{x}}, R_{\hat{z}}\}$ in addition to
CNOT cost $12n - 32$ and depth $8n - 8$,
T cost $16n - 48$ and depth $8n - 6$ ($8n - 3$ for odd n),
H cost $8n - 32$ and depth $4n - 11$.

We note that, if $2 \leq n \leq 5$, the costs and depths can be easily found using Lemma 5, such that Lemma 8 is used for a subset of controls of size 3, circuit (7) is used for size 2, and a CZ (as H-CNOT-H) gate is used for size 1. The number of single-qubit gates can be reduced in some of these cases.

Moreover, we show in Appendix B that a trade-off between the depth of CNOT and T gates can be achieved. With no change to any cost, it is possible to reduce the depth of T gates to $5n + O(1)$, while increasing the depth of CNOT gates to $11n + O(1)$. The T depth can be further reduced to $4n$; however, this results in a CNOT cost and depth of $12.5n + O(1)$ and $12n + O(1)$, respectively.

In Appendix E we show that, if a set χ of $n_\chi \leq \lfloor (n-6)/2 \rfloor$ dirty ancilla qubits is available, *each* ancilla can be used to reduce the cost of Clifford+T gates, as presented in Fig. 1. If $n_\chi = \lfloor (n-6)/2 \rfloor$, the cost of both the T and CNOT gates is reduced to $8n + O(1)$.

Moreover, Appendix C provides a method that extends our structure and allows multiple target qubits (MCMTSU2).

B. Multicontrolled *X*

In this section, we use the MCSU2 structure in order to efficiently implement the multicontrolled X gate with a single or multiple targets, and with a single dirty ancilla qubit. In this case, the dirty ancilla is required for an exact implementation of any MCX gate with $n \geq 3$ controls using only Clifford+T gates, from the determinant argument stated in Ref. [74].

In order to implement an $[X]_t^C$ gate with one dirty ancilla qubit a, we use the MCSU2 implementation of $[R_{\hat{v}}(\lambda)]_{t'}^{C'}$ with $\lambda = 2\pi$, $C' = \{C, t\}$, and $t' = a$. As previously shown, $[R_{\hat{v}}(2\pi)]_a^{\{C,t\}} = [-I]_a^{\{C,t\}} = [Z]_t^C$ for any choice of $\hat{v}$, and simply using the well-known special case of Lemma 2, we get $[X]_t^C = [H]_t [R_{\hat{v}}(2\pi)]_a^{\{C,t\}} [H]_t$. Setting $\hat{v} = \hat{y}$ and implementing $[R_{\hat{y}}(2\pi)]_a^{\{C,t\}}$ according to Lemma 4 with $\hat{v}_1 = \hat{z}$ and $\hat{v}_2 = \hat{x}$ is a convenient choice. The relative phase $[\Delta]$ gates are added similarly to Lemma 5. This

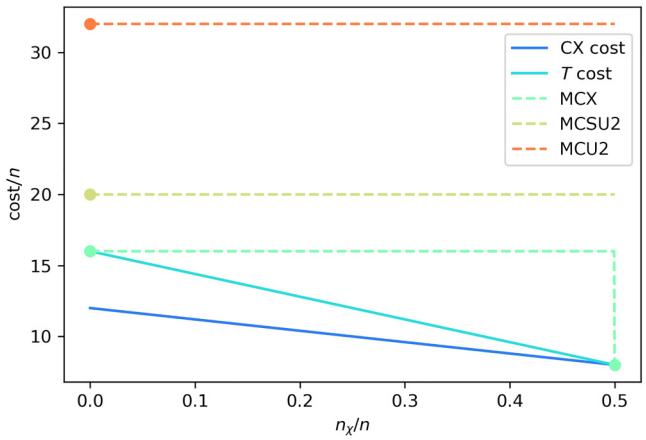
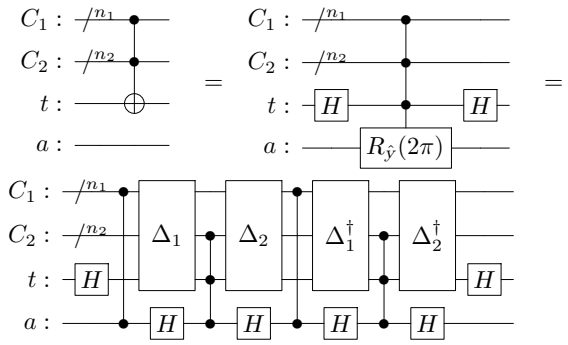


FIG. 1. Presenting the leading coefficient of CNOT and T costs for a single target, as a function of available dirty ancilla qubit. In our implementations, the leading terms are the same for MCX, MCU2, and MCSU2, and are marked as CX or T cost. Each of the best-known methods that we use for comparison (Table I) provides the same leading term for CNOT and T cost. The results provided for these methods are marked as dots, which are continued as a dashed line if there is no improvement due to additional ancilla. For MCX, the results from Ref. [74] are shown for $n/2 + O(1)$ ancilla.

provides the following circuit.



(8)

The cost and depth of this implementation are similar to those of the multicontrolled $SU(2)$ gate with $n + 1$ controls. Arbitrary $\{R_{\hat{x}}, R_{\hat{z}}\}$ gates are removed and six H gates are added instead. Notice that the control set C' can be ordered such that two H gates are cancelled out in the full Clifford+T decomposition. Theorem 2 simply follows.

Theorem 2. Any multicontrolled X gate with $n \geq 5$ controls and a single target can be implemented with one dirty ancilla qubit at the same costs and depths as in Theorem 1, with n replaced by $n + 1$ and four additional H gates. Arbitrary $\{R_{\hat{z}}, R_{\hat{x}}\}$ gates are not needed.

As mentioned above, the MCSU2 gate can also be implemented using the Toffoli+H gate set with only $4n + O(1)$ Toffoli gates, and the same holds for the MCX with one

dirty ancilla. We thus provide a Toffoli count reduction of roughly 50% for the first time compared to the best-known count of $8n + O(1)$, which was introduced in 1995 by Barenco *et al.* [3]. Furthermore, our Toffoli count scales similarly to the best-known method, which requires $O(n)$ dirty ancilla [74]. Although the number of Toffoli gates is the same, each of our Toffoli gates requires three CNOT and four T gates, while $n + O(1)$ dirty ancilla allows each Toffoli to be implemented at a reduced cost using two CNOT and two T gates [74].

In Appendix E we show that, starting from our one dirty ancilla implementation, each added ancilla effectively allows several Toffoli gates to be applied at this reduced cost, and once $n + O(1)$ ancillae are added, we reach the same cost scaling for CNOT and T gates as in Ref. [74]. Then, three-controlled Toffoli gates are introduced to double the reduction allowed by each ancilla, reaching the same gate count of $8n + O(1)$ CNOT and T gates using only $n/2 + O(1)$ dirty ancillae, as shown in Ref. [74] as well. Although we were unable to reduce the cost compared to $O(n)$ ancilla methods, this provides a bridge between 1 and $n/2 + O(1)$ dirty ancilla methods, such that any number of ancilla between those can also be used to reduce the cost (Fig. 1), thus harnessing all available resources.

In Appendix D we extend the multicontrolled X structure to a multicontrolled multitarget X gate (MCMTX) such that, if the number of target qubits is larger than one, an ancilla qubit is not required. Moreover, additional target qubits can also act as dirty ancilla and provide cost reductions due to the structure of circuit (D2).

C. Multicontrolled $U(2)$

In this section, we use the MCMTSU2 structures in order to efficiently implement the multicontrolled $U(2)$ gate (MCU2) using one clean ancilla, with a single or multiple targets. The clean ancilla, while being an expensive resource, seems necessary here since there are no known ways to achieve a linear count of Clifford+T gates without it. Known no-ancilla implementations require either a quadratic ($O(n^2)$) count of Clifford+T gates [32–38], or a linear count ($O(n)$) with a linear number of arbitrary single-qubit gates which require exponential precision [59] since the angles of rotation scale as $O(2^{-n})$, making them hard to implement using finite gate sets.

In order to implement any $[U]_t^C$ gate, with $U \in U(2)$ we present a new trick that allows us to use our MCMTSU2 decomposition with two targets and a clean ancilla to apply any MCU2 gate. We recall that U can be expressed as $U = e^{i\psi} R_{\hat{v}}(\lambda)$, and therefore $[U]_t^C = [R_{\hat{v}}(\lambda)]_t^C [P(\psi)]^C$. We now show that $[U]_t^C = [R_{\hat{v}}(\lambda)]_t^C [R_{\hat{z}}(-2\psi)]_{a_{|0\rangle}}^C$ with $a_{|0\rangle}$ a clean ancilla qubit in state $|0\rangle$. This follows directly from Lemma 10, which shows that any multicontrolled phase gate can be implemented as one MCSU2 using one clean ancilla.

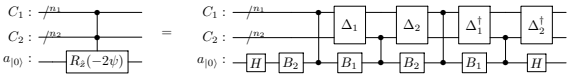
Lemma 10. $[P(\psi)]^C = [R_{\hat{z}}(-2\psi)]_{a_{|0\rangle}}^C$ for any angle ψ with $C = \{c_1, c_2, \ldots, c_n\}$, and $a_{|0\rangle}$ is a clean ancilla qubit in state $|0\rangle$.

Proof. The gate $[R_{\hat{z}}(-2\psi)]_{a_{|0\rangle}}^C$ applies the following operator on the ancilla qubit if and only if set C is in state $|11\ldots1\rangle$:

$$R_{\hat{z}}(-2\psi) = \begin{pmatrix} e^{i\psi} & 0 \\ 0 & e^{-i\psi} \end{pmatrix} = e^{i\psi} \begin{pmatrix} 1 & 0 \\ 0 & e^{i(-2\psi)} \end{pmatrix}.$$

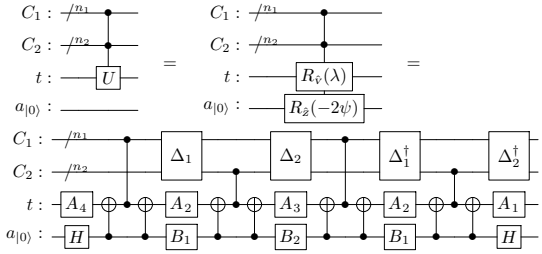
Therefore, it can be written as $[R_{\hat{z}}(-2\psi)]_{a_{|0\rangle}}^C = [P(-2\psi)]_{a_{|0\rangle}}^C [P(\psi)]^C$. Since $a_{|0\rangle}$ is known to be in the state $|0\rangle$, the controlled gate $[P(-2\psi)]_{a_{|0\rangle}}^C$ has no effect and can be removed. ■

The gate $[R_{\hat{z}}(2\psi)]_{a_{|0\rangle}}^C$ can be implemented using Lemma 5, with A_4 transforming the $\hat{z}$ axis to $\hat{x}$, and thus can be replaced with H. The gate $[R_{\hat{z}}(-2\psi)]_{a_{|0\rangle}}^C$ can therefore be implemented as the inverse of this structure as



(9)

with $B_1 = R_{\hat{x}}(\psi/2)$ and $B_2 = R_{\hat{x}}(-\psi/2)$. The first B_2 gate can commute with the first H gate, and transform to an $R_{\hat{z}}$ gate operating on the state $|0\rangle$. This gate therefore only applies a global phase and can be removed. Since $[R_{\hat{v}}(\lambda)]_t^C [R_{\hat{z}}(-2\psi)]_{a_{|0\rangle}}^C$ is an MCMTSU2 gate with $m = 2$ targets, it can be implemented up to a global phase using Lemmas C1 and C2 from Appendix C as follows.



(10)

Moreover, any MCMTU2 gate $\prod_{j=1}^m [U_j]_{t_j}^C$ with $m \geq 1$ targets, such that $U_j = e^{i\psi_j} R_{\hat{v}_j}(\lambda_j) \in U(2)$, can be implemented as the MCMTSU2 gate $[R_{\hat{z}}(-2\psi)\hat{z}]_{a_{|0\rangle}}^C \prod_{j=1}^m [R_{\hat{v}_j}(\lambda_j)]_{t_j}^C$ such that $R_{\hat{z}}(-2\psi), R_{\hat{v}_j}(\lambda_j) \in SU(2)$ and $\psi = \sum_{j=1}^m \psi_j$. Theorem 3 follows.

Theorem 3. Any multicontrolled multitarget $U(2)$ gate with $n \geq 6$ controls and $m \geq 1$ targets can be implemented with one ancilla qubit in state $|0\rangle$ with the same costs and depths as an MCMTSU2 gate implemented using

TABLE I. A summary of our ATA multicontrolled single-target results compared to previous methods.

Gate			CNOT		T	
Type	Ancilla	Source	Cost	Depth	Cost	Depth
MCSU2	None	[3]	$\sim 48n$	$\sim 48n$	$\sim 48n$	$\sim 48n$
		[74]	$\sim 32n$	$\sim 32n$	$\sim 32n$	$\sim 32n$
		[7]	$\sim 28n$	$\sim 28n$	$\sim 28n$	$\sim 28n$
		[39]	$\sim 20n$	$\sim 20n$	$\sim 20n$	$\sim 20n$
		Ours	$12n-32$	$8n-8$	$16n-48$	$8n-3$
MCX	One dirty	[3]	$\sim 24n$	$\sim 24n$	$\sim 24n$	$\sim 24n$
		[7,74]	$\sim 16n$	$\sim 16n$	$\sim 16n$	$\sim 16n$
		Ours	$12n-20$	$8n$	$16n-32$	$8n+5$
MCU2	One clean	[3]	$\sim 48n$	$\sim 48n$	$\sim 48n$	$\sim 48n$
		[7,74]	$\sim 32n$	$\sim 32n$	$\sim 32n$	$\sim 32n$
		Ours	$12n-24$	$8n$	$16n-48$	$8n-3$

Theorem C1 with $m+1$ targets and five $\{R_{\hat{z}}/R_{\hat{x}}\}$ gates replaced with two H gates.

Our results for the discussed multicontrolled single-target gates are summarized in Table I. We compare our results to state-of-the-art methods that achieved linear Clifford+T gate count and a constant number of arbitrary rotations. We list the leading terms for the cost and depth of the CNOT and T gates. As can be seen, we provide improvements in each category. Our cost reductions for the CNOT and T gates are 25%–62.5%, and up to 50%, respectively, and the depth reductions are between 50% and 75%.

Table III in Appendix B presents our results for an alternative method that favors the depth of T gates, as discussed therein.

We discuss cost reductions that can be applied, in the case that additional dirty ancilla qubits are available, in Appendix E. These reductions can also be applied to MCMTX gates with $m > 2$ targets without ancilla, as noted in Appendix D. The results for our multicontrolled single-target gate implementations with additional ancilla are presented in Fig. 1.

III. LINEAR-NEAREST-NEIGHBOR CONNECTIVITY

We focus on the same gate types that we covered for all-to-all connectivity, this time under the restrictions of LNN connectivity. One main difference between these two challenges is that, in the LNN case, when an MC gate with n controls is implemented over a set of $k > n$ qubits, we need to account for any possible choice of the control and target qubits. The approach taken in known methods is first to assume a specific choice for these qubits, and then apply SWAP gates in order to move these qubits to their place [29,52–55]. However, this approach results in an overhead of the CNOT gate count with a quadratic upper bound ($O(kn)$). As we will show, an alternative approach allows for a linear ($O(n+k)$) scaling of the gate count, by only assuming the location of one or two qubits instead of all of them. As moving one qubit to its place only requires a linear ($O(k)$) number of SWAP gates, this overhead maintains the linear scaling. As before, our MCX and MCU2 structures are built upon the MCSU2 and the MCMTSU2 structures, and thus their gate counts scale similarly, only differing by the linear overhead defined by the number (one or two) and type (target or ancilla) of qubits that should be moved into place.

A. Multicontrolled $SU(2)$

We start by constructing the MCSU2 structure, first showing how to map our macrostructure from ATA to LNN connectivity, while assuming only the location of the target qubit. Then in Sec. III A 2 we show how to decompose the structure using a linear count of three- and two-qubit gates and finally decomposing these into the Clifford+T gate set. We continue to generalize our gate and allow for multiple targets in Appendix H, achieving a linear count as well. This specific task may demonstrate some of the challenges in implementing efficient LNN circuits, compared to their all-to-all versions.

1. MCSU2 *macrostructure*

In this section, we provide an implementation of any LNN-restricted multicontrolled $SU(2)$ gate with n control qubits $C = \{c_1, c_2, \ldots, c_n\}$ and a single target qubit t. The gate is implemented in a circuit over $k' \geq n+1$ qubits $Q' = \{q'_1, q'_2, \ldots, q'_{k'}\}$ such that the qubits q'_l and q'_{l+1} are nearest neighbors for $1 \leq l < k'$ (LNN qubits).

We define the set $Q = \{q_1, q_2, \ldots, q_k\}$ of size k, such that $Q \subset Q'$, as the *smallest* set of LNN qubits that satisfies $C \subset Q$ and $t \in Q$. Each control qubit in the set C corresponds to an arbitrary qubit in the set $Q \backslash t$, and it is assumed that the target qubit is located at the bottom, i.e., $t = q_k$, which implies that $q_1 \in C$. We will discuss the additional cost (communication overhead) required in the case $t \neq q_k$ in Sec. III A 2.

The macrostructure described in Lemma 5 can be used directly. In order to allow for an efficient LNN implementation, we require an additional constraint on the subsets C_1 and C_2, such that in each subset only the first two qubits may be nearest neighbors. The importance of this constraint will be demonstrated in Sec. III A 2. For any given sets C and Q, the subsets can be defined such that this constraint is met and $C_1 \cup C_2 = C$. We use the following procedure to define the sets.

Start with $C_1 = \{q_1\}$, $C_2 = \emptyset$, and the iteration index $l = 2$. At each iteration, q_l is added to C_1 if

$$(q_l \in C) \wedge (l = 2 \vee (q_{l-1} \notin C_1 \wedge q_{l-1} \neq C_2[1])).$$

If this condition is not met and $q_l \in C$, the qubit q_l is added to C_2. Finally, l increases by 1 for the next iteration,

stopping when $C_1 \cup C_2$ holds all the control qubits in C, i.e., $n_1 + n_2 = n$.

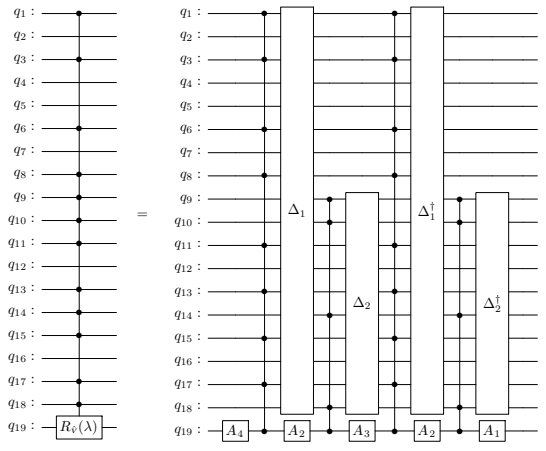
We define two subsets $Q_1, Q_2 \in Q$ of sizes k_1 and k_2 as the *smallest* sets of LNN qubits that satisfy $\{C_1, t\} \in Q_1$ and $\{C_2, t\} \in Q_2$. The relative-phase gates $[\Delta_1]$ and $[\Delta_2]$ from Lemma 5 operate on $Q_1 \setminus t$ and $Q_2 \setminus t$.

For a specific example of a given set $Q = \{q_1, q_2, \ldots, q_{19}\}$, such that $t = q_{19}$, we use a binary string to describe the control sets C, C_1, and C_2 such that the string holds "1" for index l if and only if q_l is in the control set. The following provides an arbitrarily chosen set C, and the corresponding subsets C_1 and C_2 achieved using the procedure above:

$$C = [1,0,1,0,0,1,0,1,1,1,1,0,1,1,1,0,1,1,0],$$
$$C_1 = [1,0,1,0,0,1,0,1,0,0,1,0,1,0,1,0,1,0,0],$$
$$C_2 = [0,0,0,0,0,0,0,0,1,1,0,0,0,1,0,0,0,1,0].$$

As can be seen, for each set C_1 and C_2 it is true that there are no neighboring controls, excluding the first two. In this example, $Q_1 = Q$ and $Q_2 = Q[9:19]$ are of size $k_1 = 19$ and $k_2 = 11$.

The following circuit describes the structure achieved for this example.



(11)

We note that, in the case $C_2 = \emptyset$, each structure controlled by C_2 is implemented as a single Z gate that can be cancelled out. This allows us to reduce the number of $\{R_{\hat{x}}, R_{\hat{z}}\}$ to five, similarly to the "ABC" structure described in Ref. [3].

2. MCSU2 *decomposition*

In this section, we provide a decomposition of $[\Delta_1]_{\{Q_1 \setminus t\}}[Z]_t^{C_1}$, $[\Delta_2]_{\{Q_2 \setminus t\}}[Z]_t^{C_2}$, and their inverse gates in terms of Clifford+T gates, as required for the structure described in Sec. III A 1. As mentioned, we can assume that only the first two control bits in each of the subsets C_1 and C_2 may be neighboring, and the target qubit t is located at the bottom. First, we define the following notation:

$$\{\overline{Z}\}_{C,Q}^{l} = \prod_{j=l_0}^{l} [\overline{Z}]_{q_j}^{C^j}, \quad [\overline{Z}]_{q_j}^{C^j} = \begin{cases} I & q_j \in C, \\ [Z]_{q_j}^{C^j} & q_j \notin C, \end{cases}$$

with $Q = \{q_1, q_2, \ldots, q_k\}$, $C = \{c_1, c_2, \ldots, c_n\} \in Q \setminus q_k$, and $l_0 \le l \le k$. The subset $C^j := Q[1:j-1] \cap C$ with $2 \le j \le k$ holds all control qubits above q_j. In addition, l_0 is defined as the index of the first noncontrol qubit in Q, i.e., the smallest index that satisfies $q_{l_0} \notin C \wedge C^{l_0} \neq \emptyset$ and $2 \le l_0 \le k$.

Lemma 11 provides a way to transform $\{\overline{Z}\}_{C,Q}^{l-1}$ to $\{\overline{Z}\}_{C,Q}^{l}$ using two MCX gates. As we will show, if none of the control qubits in C are neighboring (or only the first two), each of the used MCX gates is applied on neighboring qubits and only has one or two controls, i.e., LNN CNOT or LNN Toffoli. This provides the LNN version of Lemma 7, with the Toffoli gates serving a similar purpose of "adding a control" to the $\{\overline{Z}\}_{C,Q}^{l}$ gate, while the CNOT gates are added in order to "skip" noncontrol qubits [as we exemplified in circuits (12) and (13)]. This ability to skip noncontrol qubits allows us to apply our structure for any choice of the control qubits under the mentioned constraints. This in turn removes the need to assume a specific choice and then apply many qubit swaps which result in a quadratic overhead of CNOT gates, a strategy that is used in state-of-the-art implementations [29,52–55].

Lemma 11. If $l_0 < k$ and $q_l \notin C$ with $l_0 < l \le k$, then $\{\overline{Z}\}_{C,Q}^{l} = [X]_{q_{f_l}}^{Q[f_l+1:l]} \{\overline{Z}\}_{C,Q}^{l-1} [X]_{q_{f_l}}^{Q[f_l+1:l]}$, with f_l as the smallest index that satisfies $\{\overline{Z}\}_{C,Q}^{l-1} = \{\overline{Z}\}_{C,Q}^{f_l}$ and $l_0 \le f_l < l$.

Proof. The definition of f_l implies that $q_{f_l} \notin C$ and $Q[f_l+1:l] \setminus q_l \in C$. In other words, q_{f_l} is the nearest noncontrol qubit above q_l. Therefore,

$$\{\overline{Z}\}_{C,Q}^{l-1} = \{\overline{Z}\}_{C,Q}^{f_l} = \{\overline{Z}\}_{C,Q}^{f_l-1} [Z]_{q_{f_l}}^{C^{f_l}}.$$

Since the qubits $Q[f_l:l]$ are not used in $\{\overline{Z}\}_{C,Q}^{f_l-1}$, this gate commutes with $[X]_{q_{f_l}}^{Q[f_l+1:l]}$. Therefore,

$$[X]_{q_{f_l}}^{Q[f_l+1:l]} \{\overline{Z}\}_{C,Q}^{l-1} [X]_{q_{f_l}}^{Q[f_l+1:l]} = \{\overline{Z}\}_{C,Q}^{f_l-1} [X]_{q_{f_l}}^{Q[f_l+1:l]} [Z]_{q_{f_l}}^{C^{f_l}} [X]_{q_{f_l}}^{Q[f_l+1:l]}.$$

According to Lemma 6,

$$[X]_{q_{f_l}}^{Q[f_l+1:l]} [Z]_{q_{f_l}}^{C^{f_l}} [X]_{q_{f_l}}^{Q[f_l+1:l]} = [Z]_{q_{f_l}}^{C^{f_l}} [Z]_{q_l}^{C^l}$$

where we have used $[Z]_{q_l}^{C^l} = [Z]^{\{C^{f_l}, Q[f_l+1:l]\}}$, since $Q[f_l+1:l] \setminus q_l \in C$ and $q_{f_l} \notin C$ due to the definition of f_l. Finally, we can substitute and use

$$\{\overline{Z}\}_{C,Q}^{f_l-1} [Z]_{q_{f_l}}^{C^{f_l}} [Z]_{q_l}^{C^l} = \{\overline{Z}\}_{C,Q}^{l-1} [Z]_{q_l}^{C^l} = \{\overline{Z}\}_{C,Q}^{l}.$$ ■

Since there are no two neighboring control qubits for $l_0 < l \leq k$, the only possible values of f_l are

$$f_l = \begin{cases} l-1 & q_l, q_{l-1} \notin C, \\ l-2 & q_l, q_{l-2} \notin C \wedge q_{l-1} \in C, \end{cases}$$

by the definition of f_l. Therefore, the $[X]_{q_f}^{Q[f_l+1:l]}$ gates in Lemma 11 are restricted to $[X]_{q_{l-1}}^{q_l}$ or $[X]_{q_{l-2}}^{\{q_{l-1},q_l\}}$, i.e., either LNN CNOT or LNN Toffoli gates, which can be implemented up to a relative phase.

Using the notation

$$[\overline{X}]_{q_l} = \begin{cases} I & q_l \in C, \\ [X]_{q_{l-1}}^{q_l} & q_l, q_{l-1} \notin C, \\ [X_\Delta]_{q_{l-2}}^{\{q_l,q_{l-1}\}} & q_l, q_{l-2} \notin C \wedge q_{l-1} \in C, \end{cases}$$

and repeatedly applying the recursive formula presented in Lemma 11, we achieve the following lemma.

Lemma 12. If $l_0 < k$, $q_k \notin C$, and $q_l \notin C \vee q_{l+1} \notin C$ for any $l_0 < l < k$, then

$$\{\overline{Z}\}_{C,Q}^{k} = \left(\prod_{l=l_0+1}^{k} [\overline{X}]_{q_l}\right)^{\dagger} [Z]_{q_{l_0}}^{C^{l_0}} \left(\prod_{l=l_0+1}^{k} [\overline{X}]_{q_l}\right)$$

for $n = |C| \geq 1$ and $k = |Q| > n$.

Since the first two controls might be neighboring, we get $|C^{l_0}| \in \{1, 2\}$. Therefore, the gate $[Z]_{q_{l_0}}^{C^{l_0}}$ is restricted to $[Z]_{q_{l_0}}^{q_{l_0-1}}$ or $[Z]_{q_{l_0}}^{\{q_{l_0-2},q_{l_0-1}\}}$, i.e., either LNN CZ or LNN controlled-controlled-Z (CCZ).

The total number of $[\overline{X}]_{q_l}$ gates used to implement $\{\overline{Z}\}_{C,Q}^{k}$ according to Lemma 12 is $2(k - l_0)$. The number of LNN $[X_\Delta]$ gates equals $2|Q[l_0 : k] \cap C| = 2(n - |C^{l_0}|)$, i.e., twice the number of control qubits located below q_{l_0}. The I gates are not required to be implemented and their number is the same as the number of $[X_\Delta]$ gates. Therefore, the number of LNN CNOT gates is $2(k - l_0) - 4(n - |C^{l_0}|)$.

Since $Q[1] \in C$, we get $l_0 = |C^{l_0}| + 1$ with $l_0 \in \{2, 3\}$, and we can define $l_0' = l_0 - 2$ such that $l_0' = 0$ indicates the use of a CZ gate and $l_0' = 1$ indicates the use of a CCZ gate. The number of Toffoli gates can be written as $2(n - l_0' - 1)$, and the number of CNOT gates as $2k - 4n + 2l_0'$.

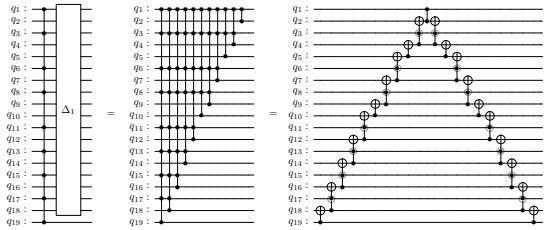
Circuits (12) and (13) describe the implementations of

$$[\Delta_1]_{\{Q_1\setminus t\}}[Z]_t^{C_1} = \{\overline{Z}\}_{C_1,Q_1}^{k_1-1}[Z]_t^{C_1} = \{\overline{Z}\}_{C_1,Q_1}^{k_1}$$

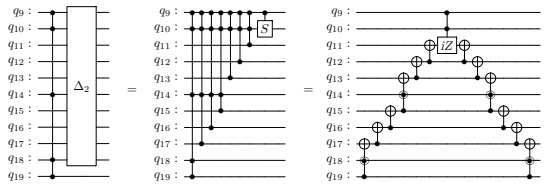
and

$$[\Delta_2]_{\{Q_2\setminus t\}}[Z]_t^{C_2} = [S]^{C_2[1:2]}\{\overline{Z}\}_{C_2,Q_2}^{k_2-1}[Z]_t^{C_2} = [S]^{C_2[1:2]}\{\overline{Z}\}_{C_2,Q_2}^{k_2}$$

according to Lemma 12, for C_1, C_2, Q_1, and Q_2 as in the example defined in Sec. III A 1.



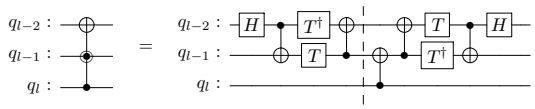
(12)



(13)

Here we add the controlled-S (CS) gate in order to transform the CCZ gate to an $[iZ]$, similarly to the ATA case in Sec. II A 2.

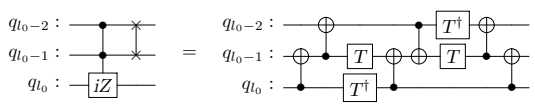
We now wish to decompose the structure in terms of basic Clifford+T gates, under the condition that CNOT gates can only apply on nearest-neighboring qubits. All of the CNOT and CZ gates applied in the structure are already applied on nearest neighbors. All $[X_\Delta]$ gates are of the same form, with controls q_l and q_{l-1} and target q_{l-2}. Each such gate can be implemented as the following LNN restricted circuit.



(14)

As in the ATA case (Appendix A), the gates on the left of the barrier can commute with other gates in the structure, and allow for a reduced depth.

In the case that a CZ gate is used in the circuit, it is simply replaced by a CNOT and two H gates. In the case that an $[iZ]$ gate is used, it can be replaced by an $[iZ]$ with an additional SWAP gate (which is cancelled out in the full MCSU2 structure) operating on the controls. This gate can be implemented as follows.



(15)

Using the l_0' notation, the gate $[Z]_{q_{l_0}}^{C^{l_0}}$ can be replaced with two H and one CNOT gate for $l_0' = 0$, or six CNOT cost or depth along with four T gates in depth two for $l_0' = 1$. Therefore, the CNOT depth or cost is $5l_0' + 1$, T cost is $4l_0'$

and depth is $2l'_0$, and H cost is $2(1-l'_0)$. The H gates can be applied on the top qubit and do not contribute to the depth. Lemma 13 follows.

Lemma 13. A $\{\overline{Z}\}^k_{C,Q}$ gate with $n \geq 3$ and $k > n$, such that only the first two control qubits may be neighboring, and $q_k \notin C$, can be implemented up to a SWAP-CS gate on two neighboring controls using

CNOT cost $2k+6n-3l'_0-9$ and depth $2k+2n+l'_0-1$,

T cost $8n-4l'_0-8$ and depth $2n$,

H cost $4n-6l'_0-2$ and depth $2n-2l'_0$.

This structure can be used in order to efficiently decompose any multicontrolled $SU(2)$ operator using the LNN restricted version of Lemma 5, as discussed in Sec. III A 1. The only assumption is that the target qubit is located at the bottom of the circuit. Each of the large MCZ gates followed by a relative-phase gate can be implemented as $\{\overline{Z}\}^{k_1}_{C_1,Q_1}$ or $\{\overline{Z}\}^{k_2}_{C_2,Q_2}$ with or without a CS and SWAP gate applied on two neighboring control qubits.

If applied on the structures corresponding to C_1, the SWAP gates are simply cancelled out since the first two qubits are unaffected by the structure corresponding to C_2. If applied on the structures corresponding to C_2, the SWAP gates must commute with the $[\Delta_1^\dagger]$ gate in order to cancel out. From the definition of $\{\overline{Z}\}^{k_1}_{C_1,Q_1}$, it is clear that swapping any two neighboring noncontrol qubits has no effect, and thus the SWAP gates cancel out in this case as well.

Each cost and depth described in Lemma 13 can be written as $a_k k + a_n n + b_l l'_0 + b$. Therefore, the total costs and depths of the entire MCSU2 structure can be written as $2a_k(k_1+k_2)+2a_n n+2b_l(l'_{0,1}+l'_{0,2})+4b$. Noting that $Q[1:2] \notin C_2$ for $l'_{0,1}=1$, and $Q[1:3] \notin C_2$ for $l'_{0,1}=0$, we can write $k_2 \leq k_1 - 3 + l'_{0,1}$ and $k_1 = k$. So, the costs and depths in the worst case can be written as

$$4a_k k + 2a_n n + 2a_k(l'_{0,1}-3) + 2b_l(l'_{0,1}+l'_{0,2}) + 4b.$$

The depth of H gates is reduced by three due to paralleled gates in the full structure, and, in addition, if $l'_{0,1}=0$, two H gates can be cancelled out. By choosing the values of $l'_{0,1}$ and $l'_{0,2}$ that maximize each cost and depth, we achieve the following upper bounds.

Lemma 14. Any multicontrolled $SU(2)$ gate with $n \geq 6$ controls can be implemented over $k > n$ qubits in LNN connectivity such that the target qubit is located at the bottom, using eight gates from $\{R_{\hat{x}}, R_{\hat{z}}\}$, in addition to no more than

CNOT cost $8k+12n-48$ and depth $8k+4n-8$,

T cost $16n-32$ and depth $4n$,

H cost $8n-10$ and depth $4n-3$.

Finally, in order to account for any possible qubit permutation, we discuss the added cost in the case $t \neq q_k$. Applying a SWAP chain before and after the MCSU2 gate allows one to relocate the target qubit. Noting that the LNN requirements hold if the order of Q is reversed, the structure can be used if t is either above or below all qubits in C. The maximal distance of t from the nearest edge is $\lfloor (k-1)/2 \rfloor$ qubits, and therefore the maximal number of required SWAP gates is $2\lfloor (k-1)/2 \rfloor$, each equivalent to three CNOT gates. We show in Appendix F that the maximal CNOT cost and depth of these SWAP gates can be reduced to $4\lfloor (k-1)/2 \rfloor$ and $2\lfloor (k-1)/2 \rfloor + 4$, respectively. Theorem 4 therefore follows from Lemma 14, providing upper bounds of the Clifford+T cost and depth. These upper bounds include the added communication overhead in the worst case and therefore hold for any choice of the control and target qubits.

Theorem 4. Any multicontrolled $SU(2)$ gate with $n \geq 6$ controls can be implemented over $k > n$ qubits in LNN connectivity with any qubit permutation using eight gates from $\{R_{\hat{x}}, R_{\hat{z}}\}$, in addition to no more than

CNOT cost $10k+12n-50$ and depth $9k+4n-5$,

T cost $16n-32$ and depth $4n$,

H cost $8n-10$ and depth $4n-3$.

We recall that, in the ATA case, additional ancilla qubits are a useful resource that can help reduce gate counts (Appendix E). In the LNN case, ancilla qubits may be seen as an obstacle, as these increase the value of k and therefore the number of CNOT gates as shown above. However, in certain cases, the ancilla qubits can be used in order to reduce the gate count in LNN connectivity. We discuss this, along with further cost reductions for specific cases, in Appendix G.

B. Multicontrolled X

Similarly to the all-to-all case, the MCX structure is based on the MCSU2. We mainly address the difference in the CNOT overhead of assuming the location of a dirty ancilla, compared to a target. In Appendix I we also address the implementation of the multicontrolled multitarget X gate.

Using the construction described in circuit (8) for the case of LNN restricted architecture, we assume that a dirty ancilla qubit a is located below all qubits in $\{C,t\}$ and use Lemma 14 in order to apply an $[R_{\hat{y}}(2\pi)]^{\{C,t\}}_a = [Z]^C_t$ gate with $\{R_{\hat{x}}, R_{\hat{z}}\}$ gates replaced by four H gates. The location of the target does not matter, as it can be treated as one of the control qubits of the MCZ gate.

The total CNOT cost and depth added in order to locate the ancilla qubit at the bottom are no larger than $4\lfloor (n+1)/2 \rfloor$ and $2\lfloor (n+1)/2 \rfloor + 4$, respectively, similarly to the MCSU2 case, while now the ancilla closest to one of the edges can be labeled as a, and therefore the largest distance is $\lfloor (n+1)/2 \rfloor$.

All other costs and depths are achieved from the MCSU2 case as in Lemma 14 with n replaced by $n+1$. Finally, two Hadamard gates are applied to the MCX target qubit on both sides of the structure. This provides the following.

Theorem 5. Any multicontrolled X gate with $n \geq 5$ controls can be implemented over $k \geq n+2$ qubits in any permutation in LNN connectivity using no more than
CNOT cost $8k+14n-34$ and depth $8k+5n+1$,
T cost $16n-16$ and depth $4n+4$,
H cost $8n+4$ and depth $4n+3$.

As in the MCSU2 case, these upper bounds include the worst-case communication overhead and therefore hold for any location of the ancilla, target, and control qubits.

C. Multicontrolled $U(2)$

Similarly to the ATA case, we can use the MCMTSU2 structure from Appendix H to implement an MCMTU2, using the clean ancilla as one of the targets, and apply a multicontrolled rotation about the $\hat{z}$ axis on it to account for the overall required phase, using the trick shown in Lemma 10. The MCMTU2 can therefore be implemented using Theorem H1, with the adjustments mentioned in Theorem 3.

Note that, when the number of targets is small, it is beneficial to apply chains of partial swap gates in order to place all of the targets at the bottom.

An MCU2 gate requires one target and one clean ancilla qubit to be located at the bottom. In the worst case, the sum of the distances of these qubits from one of the edges is no larger than $k-2$. This can be realized by noting that, if d_1 and d_2 represent the distances of each of the qubits from the bottom edge, then the total number of necessary swaps (pairs of SWAP gates) is $D = d_1 + d_2 - 1$. Similarly, the distance from each of these qubits to the top edge is $k-d-1$, and the number of swaps will be $D' = 2k - D - 4$. As we can choose which edge to use, the required number of swaps is no larger than $\min(D', D)$. The worst case is achieved when $D = D'$ and therefore $D = k-2$. We note that if the qubit that is nearest to the chosen edge is relocated first, partial SWAP gates can be used in this case, as can be realized from Appendix F. The total number of CNOT gates required to place the target and the ancilla qubits at the bottom is no larger than $4k-8$, in depth $2k+8$. The following can be achieved using Lemma 14.

Theorem 6. Any multicontrolled $U(2)$ gate with $n \geq 6$ controls can be implemented over $k > n+1$ qubits with one ancilla qubit in state $|0\rangle$ in any permutation in LNN connectivity using 11 gates from $\{R_{\hat{x}}, R_{\hat{z}}\}$, in addition to no more than
CNOT cost $12k+12n-56$ and depth $10k+4n$,
T cost $16n-32$ and depth $4n$,
H cost $8n-8$ and depth $4n-1$.

Our upper bounds for LNN multicontrolled single-target gates over any qubit permutation are summarized in Table II. These results hold for any possible choice of controls, target, and ancillary qubits, and show for the first time that the number of Clifford+T gates used to implement these MC gates over LNN connectivity scales linearly in the number of control qubits and the overall number of qubits in the 1D array. The leading terms of the cost and depth of the CNOT and T gates are presented.

Previously known methods, such as the MCX decomposition described in Ref. [29], initially assume a specific qubit ordering. The gates used in this implementation are from the NCV library. Therefore, Table II only includes asymptotic scalings, as Clifford+T gate counts are not available. By converting their results into fault-tolerant gates, these could be compared with Theorems G1 and G2 in Appendix G. When implemented over a large circuit with arbitrary qubit permutation, the CNOT cost added due to communication overhead, which is required to reorder the qubits, scales quadratically as $O(nk)$. In contrast, our methods maintain a linear $O(k+n)$ scaling over any qubit ordering, even for the multicontrolled multitarget versions of our implementations.

IV. CONCLUSION

In this paper, we have presented new methods for the decomposition of various types of multicontrolled gates.

TABLE II. A summary of our LNN upper-bound results compared to the state of the art.

Gate			CNOT		T	
Type	Ancilla	Source	Cost	Depth	Cost	Depth
MCSU2	None	[29,52–55]	$O(nk)$	$O(k+n)$	$O(n)$	$O(n)$
		Ours	$10k+12n-50$	$9k+4n-5$	$16n-32$	$4n$
MCX	One dirty	[29,52–55]	$O(nk)$	$O(k+n)$	$O(n)$	$O(n)$
		Ours	$8k+14n-34$	$8k+5n+1$	$16n-16$	$4n+4$
MCU2	One clean	[29,52–55]	$O(nk)$	$O(k+n)$	$O(n)$	$O(n)$
		Ours	$12k+12n-56$	$10k+4n$	$16n-32$	$4n$

These gates are a crucial ingredient of many quantum algorithms [4–15]. We have focused on the implementation of MCSU2, MCX, and MCU2, along with their multitarget versions, in both ATA and LNN connectivities. For each type of gate, we use the minimal ancilla requirement that is known to allow an ATA implementation with a linear Clifford+T gate count and a constant number of arbitrary rotations. Specifically, the MCSU2 gate is implemented without an ancilla, while MCX and MCU2 use one dirty ancilla and one clean ancilla, respectively. The underlying techniques used to decompose these gates are closely related.

The decomposition of each type of gate requires a significantly lower number of basic Clifford+T gates, compared to previous methods which use the same ancilla requirements. In the ATA case, our cost reductions for the CNOT and T gates are 25%–62.5%, and up to 50%, respectively [3,7,39,74]. In terms of Toffoli gate count, we have achieved a 50% reduction compared to the best-known implementation of MCX with one dirty ancilla, which was first introduced in 1995 [3]. More strikingly, for our LNN implementations, the cost of each type of gate scales linearly with respect to the number of qubits in the circuit, regardless of the qubit ordering. To our knowledge, previous methods require a CNOT cost that scales quadratically for arbitrary qubit ordering [29,52–55] as a result of a large overhead of SWAP gates added to reorder the qubits.

Moreover, we have shown in Appendixes E and G that our results can be further improved if additional ancilla qubits are available in some cases, and for ATA connectivity we have managed to bridge between methods that use a constant number of dirty ancilla and those that use a linear number. Starting from a constant number of ancilla, we utilize each additional available dirty ancilla in order to reduce the cost until we reach the best-known cost achievable with a linear number of dirty ancilla [74], as demonstrated in Fig. 1.

As many quantum algorithms are designed using multicontrolled gates, our results directly improve the implementations of these algorithms in terms of significantly fewer basic gates, which in turn should automatically provide a corresponding reduction in error rates.

ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council (Grants No. EP/R513143/1 and No. EP/T517793/1).

B.Z. and S.B. declare a relevant patent application: United Kingdom Patent Application No. 2400486.3

DATA AVAILABILITY

The data that support the findings of this article are not publicly available. The data are available from the authors upon reasonable request.

APPENDIX A: ATA MCSU2 DEPTH REDUCTIONS

We provide a description of the depth reductions reported in Theorem 1. The relative-phase Toffoli can be expressed as $[X_\Delta]_{q_3}^{\{q_1,q_2\}} = [O_3]_{\{q_1,q_2,q_3\}}[O_2]_{\{q_1,q_3\}}$, such that $[O_2]_{\{q_1,q_3\}}$ and $[O_3]_{\{q_1,q_2,q_3\}}$ are defined as the gates on the left of the barrier and on the right of the barrier, respectively, in circuit (6). The expression in Lemma 8 can be rewritten using this notation, noting that $\prod_{j=3}^{n}[X_\Delta]_{d_{j-2}}^{\{c_j,d_{j-1}\}} = \left(\prod_{j=3}^{n}[O_3]_{\{c_j,d_{j-1},d_{j-2}\}}\right)\left(\prod_{j=3}^{n}[O_2]_{\{c_j,d_{j-2}\}}\right)$. This commutation is achieved because each $[O_2]_{\{q_1,q_3\}}$ gate is the first to access the qubits q_1 and q_3, as can be seen in circuit (5). We define the following notation:

$$\Lambda_{C,D}^{O_3} = \left(\prod_{j=3}^{n}[O_3]_{\{c_j,d_{j-1},d_{j-2}\}}\right)^{\dagger}[iZ]_{d_1}^{\{c_1,c_2\}} \times \left(\prod_{j=3}^{n}[O_3]_{\{c_j,d_{j-1},d_{j-2}\}}\right)$$

and

$$\Xi_{C,D}^{O_2} = \prod_{j=3}^{n}[O_2]_{\{c_j,d_{j-2}\}}$$

and we can rewrite Lemma 8 as follows.

Lemma A1. $\{Z\}_{C,D}^{n}[S]_{c_2}^{c_1} = (\Xi_{C,D}^{O_2})^{\dagger}\Lambda_{C,D}^{O_3}(\Xi_{C,D}^{O_2})$ for $n \geq 3$, $|C| = n$, and $|D| = n-1$.

Therefore, any $\{Z\}_{C,D}^{n}[S]_{c_2}^{c_1}$ gate can be decomposed using one $[iZ]$ gate, $2(n-2)$ $[O_3]_{\{q_1,q_2,q_3\}}$ gates, and $2(n-2)$ $[O_2]_{\{q_1,q_3\}}$ gates. Since $[O_2]_{\{q_1,q_3\}}$ gates can be applied simultaneously in each Ξ operator, the depth is reduced to one $[iZ]$ gate, $2(n-2)$ $[O_3]_{\{q_1,q_2,q_3\}}$ gates, and two $[O_2]_{\{q_1,q_3\}}$ gates, as presented in Lemma 9.

We define the following control subsets:

$$C_1 = C[1:n_1],$$
$$C_2 = C[n_1+1:n_1+n_2],$$
$$C_1' = \begin{cases} C_1[3:n_2] & n_2 = n_1, \\ \{C_1[3:n_1], C_1[2]\} & n_2 \neq n_1, \end{cases}$$
$$C_2' = C_2[3:n_1],$$

such that $n_1 = \lfloor n/2 \rfloor$, $n_2 = n - n_1 = \lceil n/2 \rceil$, $|C_1'| = n_2 - 2$, and $|C_2'| = n_1 - 2$. We choose $[\Delta_1]_C = [S]^{C_1[1:2]}\{Z\}_{C_1,C_2'}^{n_1-1}$ and $[\Delta_2]_C = [S]^{C_2[1:2]}\{Z\}_{C_2,C_1'}^{n_2-1}$ as relative-phase

gates applied on the set C. For this choice we get

$$[\Delta_1]_C[Z]_t^{C_1} = [S]^{C_1[1:2]}\{Z\}^{n_1}_{C_1,\{C_2',t\}}$$

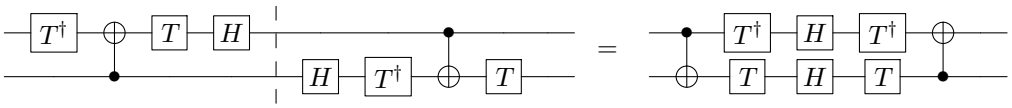
and

$$[\Delta_2]_C[Z]_t^{C_2} = [S]^{C_2[1:2]}\{Z\}^{n_2}_{C_2,\{C_1',t\}}.$$

The inverted versions of these gates are implemented in the same way. These structures can be used in order to decompose circuit (2). Our choice of the sets C_1, C_2, C_1', and C_2' allows for a slight reduction of T depth, considering the full structure. Since the Ξ operators commute with single-qubit gates applied on the target qubit t, there are three copies of $\left(\Xi^{O_2}_{C_1,C_2'}\right)\left(\Xi^{O_2}_{C_2,C_1'}\right)^\dagger$ or its inverse formed in layers between consecutive Λ operators. We note that $C_2'[j-2] = C_2[j]$ and $C_1'[j-2] = C_1[j]$ for $3 \le j \le n_1$. Therefore,

$$(\Xi^{O_2}_{C_1,C_2'})(\Xi^{O_2}_{C_2,C_1'})^\dagger = \begin{cases} \prod_{j=3}^{n_1}([O_2]_{\{C_1[j],C_2[j]\}}[O_2^\dagger]_{\{C_2[j],C_1[j]\}}) & n_2 = n_1, \\ \prod_{j=3}^{n_1}([O_2]_{\{C_1[j],C_2[j]\}}[O_2^\dagger]_{\{C_2[j],C_1[j]\}})[O_2^\dagger]_{\{C_2[n_2],C_1[2]\}} & n_2 \neq n_1. \end{cases}$$

In both cases there are $n_1 - 2$ parallel copies of the following circuit.

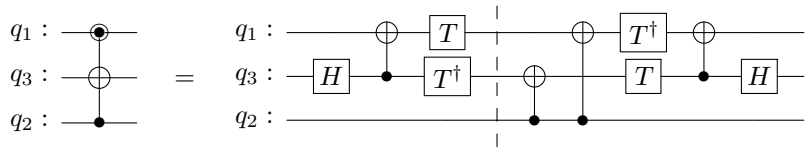


(A1)

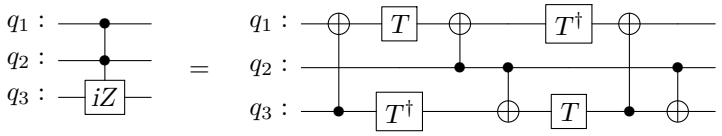
This allows for a reduction of H depth by 3. In addition, the T depth is reduced by 6 if n is even and by 3 if n is odd.

APPENDIX B: ATA MCSU2 ALTERNATIVE

We provide an alternative implementation of the multicontrolled $SU(2)$ with reduced depth of T gates. This is achieved by using the following decompositions of $[X_\Delta]$ and $[iZ]$.



(B1)

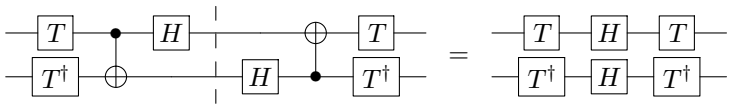


(B2)

In this case, $[O_2]_{\{q_1,q_3\}}$ and $[O_3]_{\{q_1,q_2,q_3\}}$ are defined as the gates on the left of the barrier and on the right of the barrier, respectively, in circuit (B1). Using these gates in Lemma A1 provides the following.

Lemma B1. A $\{Z\}^n_{C,D}[S]^{c_1}_{c_2}$ gate with $n \ge 3$, $|C| = n$, and $|D| = n-1$ can be implemented using
CNOT cost $8n - 11$ and depth $6n - 5$,
T cost $8n - 12$ and depth $2n$,
H cost $4n - 8$ and depth $2n - 2$.

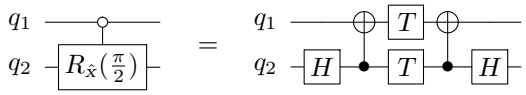
When applied as part of Lemma 5, using the sets C_1, C_2, C_1', and C_2' as defined in Appendix A, some reductions in CNOT gate *count* can be achieved as $\left(\Xi^{O_2}_{C_1,C_2'}\right)\left(\Xi^{O_2}_{C_2,C_1'}\right)^\dagger$ now comprise $n_1 - 2$ parallel copies of the following circuit.



(B3)

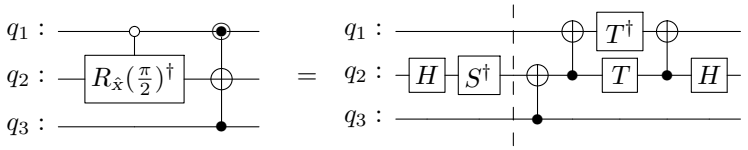
These cancellations can be applied in three layers and reduce the CNOT count by $6(n_1 - 2)$. In addition, the H depth is reduced by 3, and the CNOT depth is reduced by 6 if n is even, or by 3 if n is odd.

We now show that the CNOT cost can be further reduced by applying cancellations for the first $n_1 - 2$ $[X_\Delta]$ gates of the full MCSU2 structure. The following gate is controlled by the state $|0\rangle$, and therefore commutes with the MCSU2 gate if its control qubit is in the set C.

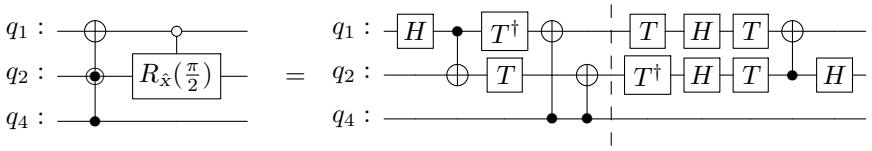


(B4)

This gate and its inverse can be applied on both sides of the MCSU2 structure with $q_1 = C_1[j]$ and $q_2 = C_2[j]$ for each value $3 \leq j \leq n_1$. The following two circuits describe the gates that can replace the first $n_1 - 2$ $[X_\Delta]$ gates, and their counterparts at the end of the MCSU2 structure. Setting $q_3 = \{C_2', t\}[j-1]$ and $q_4 = \{C_1', t\}[j-1]$ for $3 \leq j \leq n_1$, we get the following circuits.



(B5)



(B6)

Each of these circuits is used to replace $n_1 - 2$ copies of circuit (B1). This allows for a reduction of $n_1 - 2$ CNOT gates in depth 1, and an increase of $2(n_1 - 2)$ H gates in depth 1, along with the introduction of $n_1 - 2$ S gates in depth 1.

The total reduction of CNOT gates is $7(n_1 - 2)$, which is equivalent to $3.5n - 14$ if n is even, and $3.5n - 17.5$ if n is odd. The following theorem is achieved from Lemma 5, Lemma B1, and the above CNOT cancellations.

Theorem B1. Any $SU(2)$ operator controlled by $n \geq 6$ qubits can be implemented without ancilla using eight gates from $\{R_{\hat{x}}, R_{\hat{z}}\}$ in addition to

CNOT cost $12.5n - 30$ ($12.5n - 26.5$ for odd n)
and depth $12n - 27$ ($12n - 24$ for odd n),
T cost $16n - 48$ and depth $4n$,
H cost $9n - 36$ ($9n - 37$ for odd n) and depth $4n - 10$,
S cost $0.5n - 2$ ($0.5n - 2.5$ for odd n) and depth 1.

We note that it is possible to use the structure that produces Theorem 1, and use circuit (B1) to replace only the $6(n_1 - 2)$ $[X_\Delta]$ gates that allow CNOT cancellations according to circuit (B3). This approach maintains all of the costs described in Theorem 1, while reducing the T depth by $3n + O(1)$, and increasing the CNOT depth by the same. This provides a T depth of $5n + O(1)$, with no increase to the CNOT cost.

The results mentioned in Theorem B1 can be used, along with Theorems 2 and 3, in order to obtain the reduced T depth implementations of MCX and MCU2 gates.

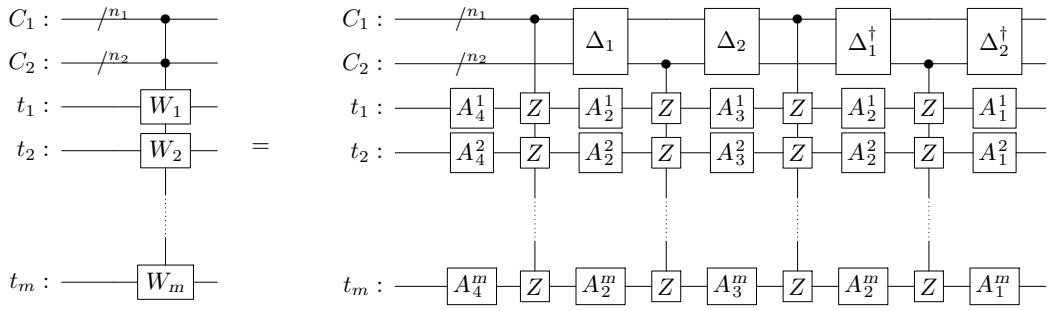
We summarize the results for multicontrolled single-target gates with reduced T depth in Table III. We use the same comparisons as used in Table I. The depth reduction for T gates is now between 75% and 87.5%, while maintaining low costs.

TABLE III. A summary of our results compared to previous methods.

Gate			CNOT		T	
Type	Ancilla	Source	Cost	Depth	Cost	Depth
MCSU2	None	[39]	$\sim 20n$	$\sim 20n$	$\sim 20n$	$\sim 20n$
		Ours	$12.5n - 26$	$12n - 24$	$16n - 48$	$4n$
MCX	One	[7]	$\sim 16n$	$\sim 16n$	$\sim 16n$	$\sim 16n$
	dirty	Ours	$12.5n - 14$	$12n - 12$	$16n - 32$	$4n + 4$
MCU2	One	[3,7]	$\sim 32n$	$\sim 32n$	$\sim 32n$	$\sim 32n$
	clean	Ours	$12.5n - 18$	$12n - 16$	$16n - 48$	$4n$

APPENDIX C: ATA MULTICONTROLLED MULTITARGET $SU(2)$

We present a method to extend the multicontrolled $SU(2)$ structure to implement a multi-control multitarget $SU(2)$ gate (MCMTSU2), in which any $SU(2)$ operator can be applied to each target. We use the following structure.

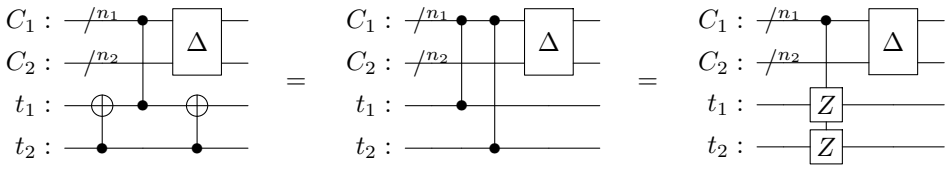


(C1)

Lemma C1. Any $\prod_{j=1}^{m}[R_{\hat{v}_j}(\lambda_j)]_{t_j}^{C}$ gate with $R_{\hat{v}_j}(\lambda_j) \in SU(2)$ can be implemented using $8m$ gates from $\{R_{\hat{x}}, R_{\hat{z}}\}$, two $\prod_{j=1}^{m}[Z]_{t_j}^{C_1}$, and two $\prod_{j=1}^{m}[Z]_{t_j}^{C_2}$ gates as in circuit (C1) such that $A_2^j = R_{\hat{x}}(-\lambda_j/4)$, $A_3^j = R_{\hat{x}}(\lambda_j/4)$, $A_4^j = R_{\hat{z}}(\theta_2^j)R_{\hat{x}}(\theta_1^j)$, $A_1^j = {A_4^j}^\dagger A_3^j$, and $C_1 \cup C_2 = C$. The $[\Delta]_C$ gates apply a relative phase.

Proof. This can be realized by applying the structure described in Lemma 5 for each $[R_{\hat{v}_j}(\lambda_j)]_{t_j}^{C}$ gate separately. The gates can be reordered to achieve circuit (C1) since MCZ gates commute with each other, and single-qubit gates operating on different qubits commute as well. ■

The following identity can be derived from Lemma 6.



(C2)

Lemma C2 describes the structure used to extend any MCZ gates to a MCMTZ gate with m targets, using $2(m-1)$ CNOT gates in depth $2\lceil \log_2(m) \rceil$.

Lemma C2. We have

$$[\Delta']_C \prod_{j=1}^{m}[Z]_{t_j}^{C'} = \left(\prod_{k=1}^{\lceil \log_2(m) \rceil}\prod_{l=1}^{f'}[X]_{t_l}^{t_{l+f}}\right)[\Delta']_C[Z]_{t_1}^{C'}\left(\prod_{k=1}^{\lceil \log_2(m) \rceil}\prod_{l=1}^{f'}[X]_{t_l}^{t_{l+f}}\right)^\dagger$$

with $C' \in C$ and $t_j \notin C$, such that $f = 2^{k-1}$ and $f' = \min(f, m-f)$.

Proof. As demonstrated in circuit (C2), it follows from Lemma 6 that

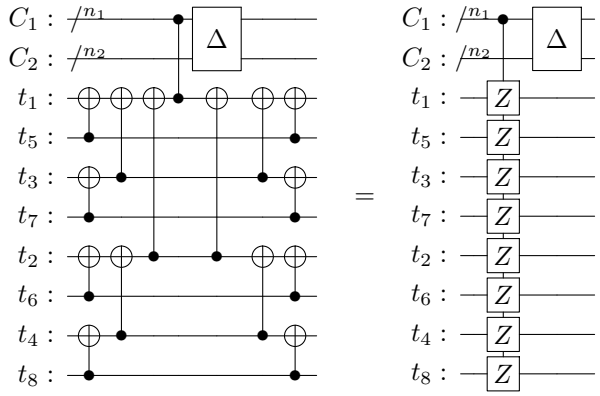
$$[\Delta']_C[Z]_{t_j}^{C'}[Z]_{t_j+f}^{C'} = [X]_{t_j}^{t_j+f}[\Delta']_C[Z]_{t_j}^{C'}[X]_{t_j}^{t_j+f}.$$

The number of targets can therefore be doubled by applying a CNOT transformation in depth 2, as follows:

$$[\Delta']_C\prod_{j=1}^{2f}[Z]_{t_j}^{C'} = \left(\prod_{l=1}^{f}[X]_{t_l}^{t_l+f}\right)[\Delta']_C\prod_{j=1}^{f}[Z]_{t_j}^{C'}\left(\prod_{l=1}^{f}[X]_{t_l}^{t_l+f}\right).$$

This can be repeatedly applied until $\log_2(f) = \lfloor\log_2(m)\rfloor$. The remaining $m-f$ targets can be added similarly by applying $\prod_{l=1}^{m-f}[X]_{t_l}^{t_l+f}$ at each side. ■

Circuit (C3) describes the structure for $m = 8$.



(C3)

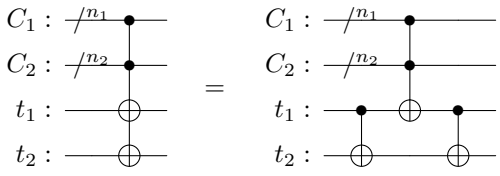
The following can be obtained using Lemmas C1 and C2.

Theorem C1. Any multicontrolled multitarget $SU(2)$ with $n \geq 6$ controls and $m \geq 1$ targets can be implemented without ancilla with the same costs and depths of an MCSU2 gate implemented using Lemma 5, with an increase of $8(m-1)$ to both $\{R_{\hat{x}}/R_{\hat{z}}\}$ and CNOT count, and $8\lceil\log_2(m)\rceil$ to CNOT depth.

The final costs and depths of our implementation can therefore be achieved by applying these adjustments to the costs and depths listed in Theorem 1.

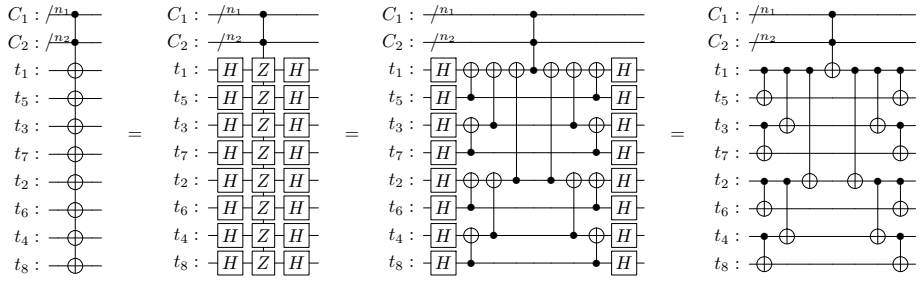
APPENDIX D: ATA MCMTX

We extend the multicontrolled X structure to a multicontrolled multitarget X gate (MCMTX). First we demonstrate that, if the number of target qubits is larger than one, an ancilla qubit is not required. The following can be easily deduced from Lemma 6 (and has been shown in Ref. [75]).



(D1)

Since t_2 is not used while the MCX gate is applied, it can be utilized as the required dirty ancilla previously referred to as qubit a. Similarly, Lemma C2 can be used in order to add any number of targets as follows.



(D2)

The last step is achieved by commuting and cancelling out the H gates, which results in transforming controls into targets and vice versa, as has been mentioned in Ref. [76, Proposition 1]. Theorem D1 simply follows.

Theorem D1. Any multicontrolled multitarget X gate with $n \geq 5$ controls and $m \geq 2$ targets can be implemented without ancilla with the same costs and depths of an MCX gate, with an increase of $2(m-1)$ to CNOT count, and $2\lceil \log_2(m) \rceil$ to CNOT depth.

The costs and depths of our MCMTX structure can be achieved by merging Theorems D1 and 2. As can be seen in circuit (D2), $m-1$ targets can be used as dirty ancilla qubits for the implementation of the MCX gate, while one acts as the required ancilla qubit a, and the remaining $m-2$ can be used in order to reduce the cost of the MCX gate. We discuss these cost reductions in Appendix E.

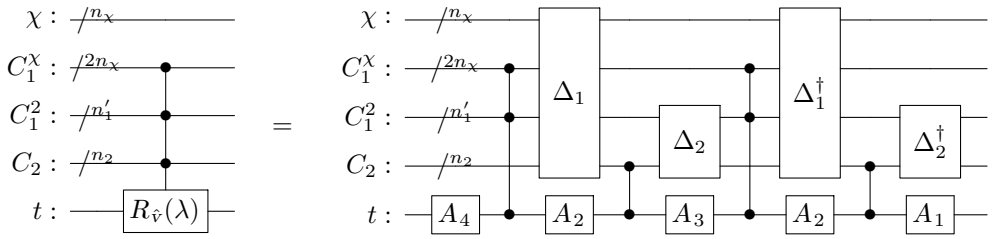
The results of this section can be used to implement any multicontrolled multitarget Π gate as well, simply by using Lemma 2 in order to change the axes of rotations. This requires no more than four arbitrary $\{R_{\hat{x}}, R_{\hat{z}}\}$ gates per target.

APPENDIX E: ATA MCMTSU2 WITH DIRTY ANCILLAE

We provide a method that reduces the cost of each type of Clifford+T gate used in our decompositions if an additional set χ of $n_\chi \leq \lfloor (n-6)/2 \rfloor$ dirty ancilla qubits is available. Since any unused qubit in the circuit can be utilized as a dirty ancilla, this reduction can be applied in any scenario in which the multicontrolled gate does not operate on all available qubits.

We note that the $[O_2]$ gates that commute out of the structure corresponding to C_1 to form Ξ layers, as mentioned in Appendix A, can be cancelled out if those commute with the structure corresponding to C_2. Such commutations can be achieved by using available dirty ancilla qubits as part of C_2' as previously defined. Each available dirty ancilla can be used in that way to reduce the cost by four $[O_2]$ gates, until the number of ancilla reaches $n + O(1)$, and all of these gates have been cancelled out. Here we present a method that allows us to double the number of gates that can be cancelled out due to each added ancilla.

The following structure can be used to implement any MCSU2 gate according to Lemma 5, with $C_1^\chi \cup C_1^2 = C_1$.



(E1)

The subsets are defined as follows:

$$C_1^\chi = C[1 : 2n_\chi],$$

$$C_1^2 = C[2n_\chi + 1 : 2n_\chi + n_1'],$$

$$C_2 = C[2n_\chi + n_1' + 1 : n],$$

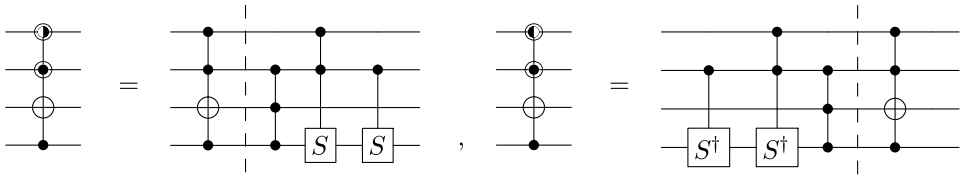
$$C_1' = \begin{cases} C_1^2[3 : n_2] & n_2 = n_1', \\ \{C_1^2[3 : n_1'], C_1^2[2]\} & n_2 \neq n_1', \end{cases}$$

$$C_2' = \{\chi, C_2[3 : n_1']\},$$

$$C_1 = \{C_1^2[1 : 2], C_1^\chi, C_1^2[3 : n_1']\}.$$

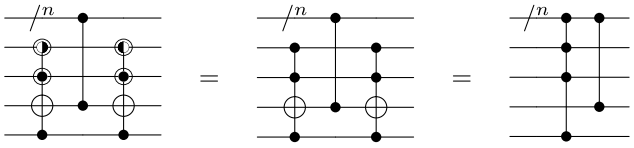
The set marked as C_1^χ holds the first $2n_\chi$ qubits of the control set C. The remaining $n - 2n_\chi$ control qubits are divided between the sets C_1^2 and C_2 of size $n_1' = \lfloor (n-2n_\chi)/2 \rfloor$ and $n_2 = \lceil (n-2n_\chi)/2 \rceil$, respectively. The set C_1 holds all $n_1 = 2n_\chi + \lfloor (n-2n_\chi)/2 \rfloor$ qubits of sets C_1^χ and C_1^2.

The gate $[\Delta_2]_{\{C_2, C_1^2\}}[Z]_t^{C_2} = [S]^{C_2[1:2]}\{Z\}_{C_2,\{C_1',t\}}^{n_2}$ is decomposed using Lemma A1 as shown in the previous sections. The gate $[\Delta_1]_{\{C,\chi\}}[Z]_t^{C_1}$ is also decomposed using Lemma A1, while the qubits in C_1^χ are arranged as n_χ pairs, and each pair is treated as a single control bit. This adjustment is simply achieved by adding a control bit to each $[X_\Delta]$ gate which is controlled by such a pair. We use the following notation for a three-controlled relative-phase Toffoli and its inverse.



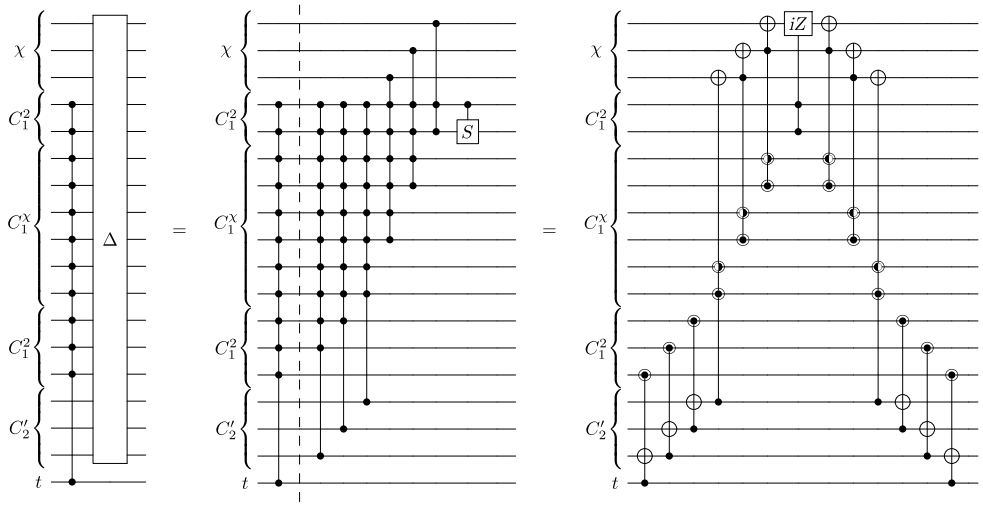
(E2)

Using Lemma 6, we get the following version of circuit (3).



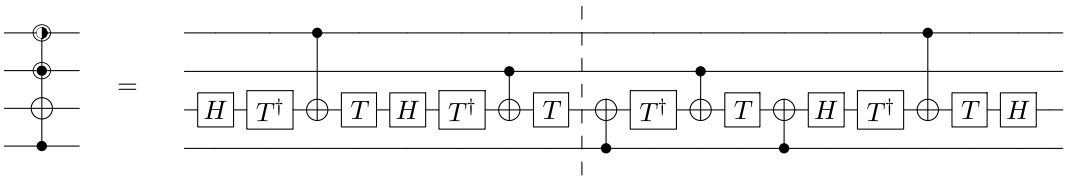
(E3)

The following circuit presents the structure used to decompose $[\Delta_1]_{\{C,\chi\}}[Z]_t^{C_1} = [S]^{C_1[1:2]}\{Z\}^{n_1}_{C_1,\{C_2',t\}}$ according to Lemma 8, while pairs of qubits from C_1^χ are treated as a single qubit.



(E4)

In order to decompose this structure in terms of Clifford+T gates, we use the following implementation, as provided in Ref. [74].



(E5)

As can be seen, when this circuit is used in order to decompose circuit (E4), all gates on the left to the barrier can commute with any gate that is applied on the left-hand side, and, for the inverse gates, a similar set of gates can commute to the right-hand side. Moreover, as can be seen from circuit (E1), these gates commute with the structures controlled by C_2, and thus gates that commute from the right-hand side of the first structure controlled by C_1 cancel out with their counterparts from the left of the second structure controlled by C_1.

The rest of these commuted gates, which commute to the right and to the left of the full MCSU2 structure, cancel out as well, since all of these gates operate on a qubit from χ, either as a single-qubit gate, or as a gate controlled by a qubit from C. Therefore, a similar sequence of these gates and their inverse can be applied [analogously to

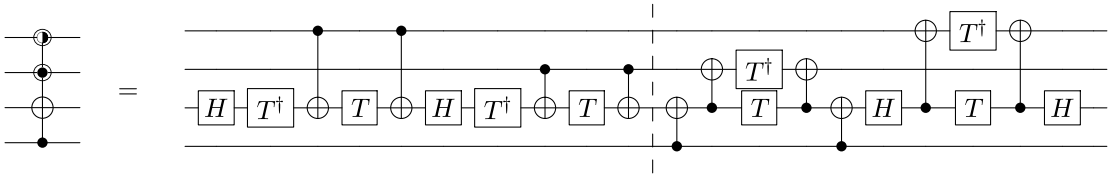
circuit (G1)] on both sides of the MCSU2 gate shown in circuit (E1) without changing the operator, and cancel out the gates that commute toward the edges. The rest of the $[X_\Delta]$ gates in circuit (E4) are decomposed in the same ways as circuit (6), creating two Ξ operators that are applied on $C_2' \cup C_1^2$.

The gate $[\Delta_1]_{\{C,\chi\}}[Z]_t^{C_1}$ can be implemented using one $[iZ]$ gate, $2n_1' - 4$ $[X_\Delta]$ gates, and $2n_\chi$ three-controlled relative-phase Toffoli gates. Note that, due to our choice of C_1, C_2, C_1', and C_2', the same depth reductions or CNOT cancellations between pairs of Ξ operators can be achieved $n_1' - 2$ times per layer for gates applied on $C_2' \cup C_1^2$. The total costs and depths of the MCSU2 structure can be simply achieved from Theorem 1, with n replaced by $n - 2n_\chi$, in addition to the costs and depths of $4n_\chi$ reduced three-controlled Toffoli gates, each contributing cost and depth of four CNOT, four T, and two H gates. We note that the steps used to produce Theorem C1 can be used in this case as well, as those are agnostic to the decomposition of the large relative-phase gates. Theorem E1 simply follows.

Theorem E1. Any multicontrolled multitarget $SU(2)$ with $n \geq 6$ controls and $m \geq 1$ targets can be implemented with $0 \leq n_\chi \leq \lfloor (n-6)/2 \rfloor$ dirty ancilla qubits using $8m$ gates from $\{R_{\hat{x}}, R_{\hat{z}}\}$ in depth 8, in addition to

CNOT cost $12n + 8m - 8n_\chi - 40$
and depth $8n + 8\lceil log_2(m) \rceil - 8$,
T cost $16n - 16n_\chi - 48$
and depth $8n - 6$ ($8n - 3$ for odd n),
H cost $8n - 8n_\chi - 32$ and depth $4n - 11$.

For a minimized T depth decomposition, the same approach is used, with $[X_\Delta]$ and $[iZ]$ gates implemented as in circuits (B1) and (B2), and the three-controlled relative-phase Toffoli implemented as the following circuit.



(E6)

The gates on the left of the barrier are removed in the same manner as described above, and the CNOT cancellations that provide Theorem B1 can be applied for gates operating on $C_2' \cup C_1^2$. The total costs and depths of the MCSU2 structure can be simply achieved from Theorem B1, with n replaced by $n - 2n_\chi$, and in addition to the costs and depths of $4n_\chi$ reduced three-controlled Toffoli gates, each contributing four T gates in depth 2, in addition to cost and depth of six CNOT and two H gates.

Theorem E2. Any multicontrolled multitarget $SU(2)$ with $n \geq 6$ controls, and $m \geq 1$ targets can be implemented with $0 \leq n_\chi \leq \lfloor (n-6)/2 \rfloor$ dirty ancilla qubits using $8m$ gates from $\{R_{\hat{x}}, R_{\hat{z}}\}$ in depth 8, in addition to

CNOT cost $12.5n + 8m - n_\chi - 38$ ($+3.5$ for odd n)
and depth $12n + 8\lceil log_2(m) \rceil - 27$ ($+3$ for odd n),
T cost $16n - 16n_\chi - 48$ and depth $4n$,
H cost $9n - 10n_\chi - 36$ (-1 for odd n) and depth $4n - 10$,
S cost $0.5n - n_\chi - 2$ (-0.5 for odd n) and depth 1.

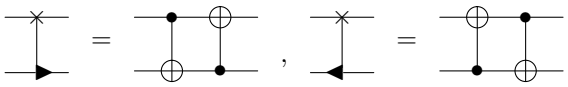
We note that the number of dirty ancilla qubits can be increased up to $\lceil (n-2)/2 \rceil$. This is achieved by allowing the implementation of the second and fourth MCZ and $[\Delta]$ gates in circuit (E1) to be implemented as $[iZ]$ or CZ gates when n_2 is reduced to 2 or 1. Finally, if $n_2 = 0$, these controlled gates are replaced by Z gates, which can commute and cancel out. This final stage might increase the Clifford+T count slightly; however, it reduces the number of single-qubit arbitrary rotations, and halves the cost and depth of CNOT gates that are added for $m > 1$ targets. Note that if this is used, and n is odd, the last control qubit will be added using an $[X_\Delta]$ gate with reduced cost, since the gates that can commute outwards cancel out similarly to the three-controlled Toffoli gates.

The results mentioned in Theorems E1 and E2 can be used, together with Theorems 2 and 3, in order to obtain the implementations of the MCX, MCMTX, MCU2, and MCMTU2 gates with additional dirty ancilla qubits, noting that, for MCMTX, $m - 2$ targets can be used as dirty ancilla for MCMTX with $m > 2$.

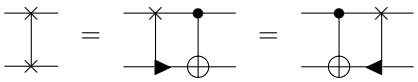
We presented our CNOT and T costs of MCSU2, MCX, and MCU2 as a function of the number of available dirty ancilla qubits in Fig. 1. For comparison, we use the known results from Table I, noting that additional ancilla qubits do not improve the costs in those methods. When about $n/2$ dirty ancilla qubits are available, our results for MCSU2, MCX, and MCU2 are close to those provided by Ref. [74] for MCX with a similar number of ancilla. When the number of ancilla is lower, our methods still benefit from this resource, while other methods use at most one of the available ancilla qubits.

APPENDIX F: LNN COMMUNICATION OVERHEAD

In Sec. III A 2, SWAP chains are used in order to relocate the target qubit t of an LNN MCSU2. We present the method used to reduce the CNOT cost required to implement the SWAP chains in Theorem 4. Each SWAP gate can be replaced with a partial SWAP that can be implemented using two CNOT gates instead of three. We use the following gate notation.

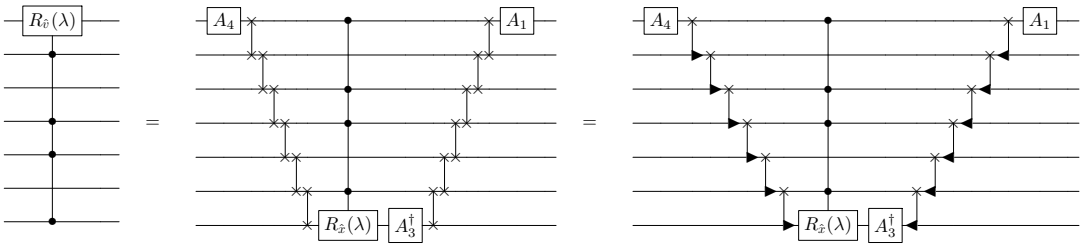


(F1)



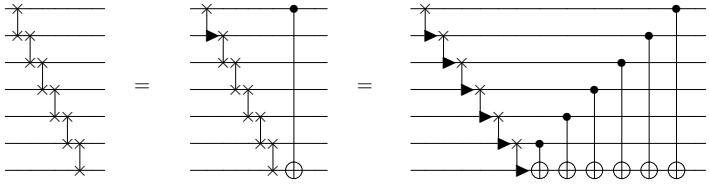
(F2)

Recalling from Lemma 5 that $[R_{\hat{v}}(\lambda)]_t^C = [A_1]_t[A_3^\dagger]_t[R_{\hat{x}}(\lambda)]_t^C[A_4]_t$, the SWAP chains can be applied as follows.



(F3)

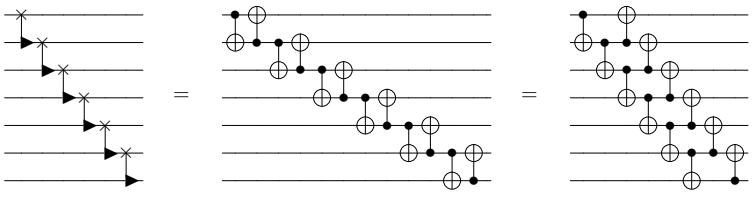
The final circuit is achieved by applying CNOT cancellations between the SWAP chains. CNOT gates are first "extracted" from each side as follows.



(F4)

The extracted CNOT gates can be removed since each such gate commutes with $[A_3^\dagger]_t[R_{\hat{x}}(\lambda)]_t^C$, as both operators apply a rotation about the $\hat{x}$ axis on the bottom qubit. In addition, the CNOT gates are controlled by qubits that are either controls of the MCSU2 gate, or are not affected by it.

Since each partial SWAP is decomposed as two CNOT gates, the maximal number of CNOT gates required for this reordering is $4d$, with d as the "distance" of the swap, i.e., the number of qubits below the location of t. The CNOT depth is no larger than $2(d+2)$, as can be realized from the following.

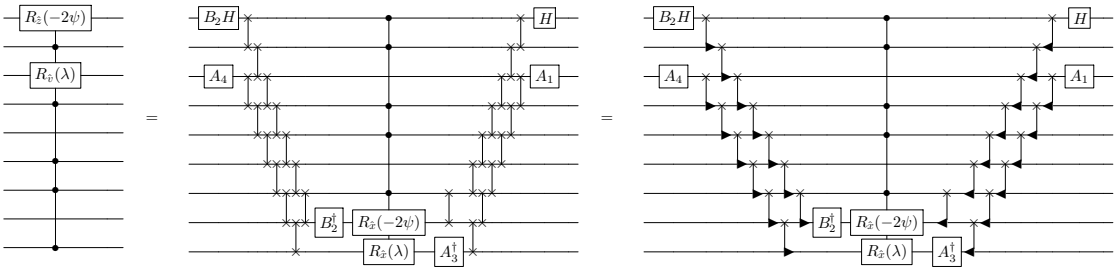


(F5)

In the MCSU2 structure, two additional CNOT gates can be cancelled out, noting that the structure between the partial SWAP chains in circuit (F3) must either begin or end with a $[X]_{q_{k-1}}^{q_k}$ gate. This is since q_{k-1} cannot be in both C_1 and C_2.

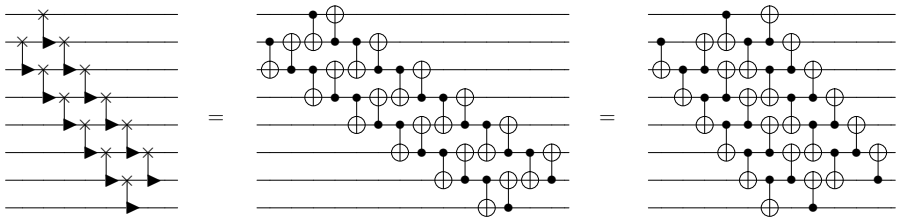
A similar analysis can be used for MCMT gates in order to place all of the target qubits at the bottom. However, the method described in Appendix H can be used instead if the number of targets is large. The MCU2 is implemented as an

MCMTSU2 with two target qubits. In this case it is beneficial to place the targets below all control qubits, as can be realized from the cost analysis in Sec. III C. The SWAP gates can be replaced by partial SWAP gates in this case as well, by first applying the swap chains for the target qubit that is nearest to the bottom as in circuit (F3), and then repeat a similar process for the second qubit. This can be realized from the following example.



(F6)

The depth of the swap chains varies slightly, depending on the distance between the target qubits. From circuit (F7), it can be noted that the CNOT depth is no larger than $2(k+4)$.

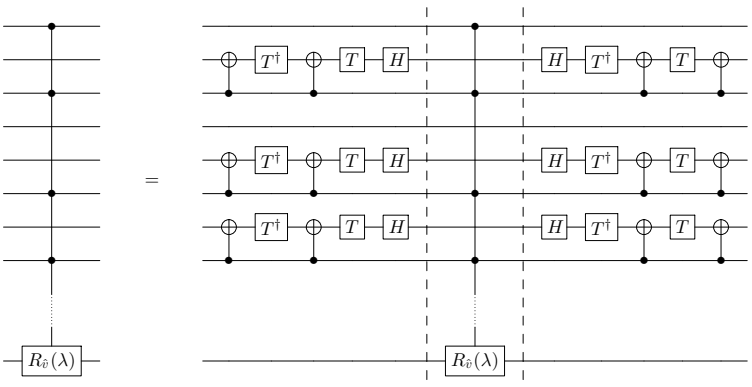


(F7)

APPENDIX G: LNN SPECIFIC CASES

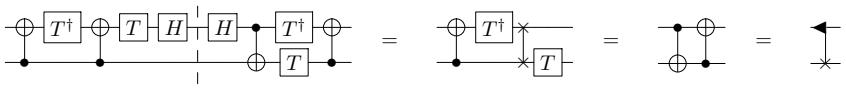
We recall that, in the ATA case, additional ancilla qubits are a useful resource, which may assist in reducing gate counts. In the LNN case, ancilla qubits may be seen as an obstacle, as these increase the value of k and therefore the number of CNOT gates as shown in Sec. III A 2. However, in certain cases, the ancilla qubits can be used in order to reduce the gate count in LNN connectivity.

As can be seen in our example in circuit (11), some of the control qubits of C_1 are located above all C_2 qubits. In this case, the gates that commute out of the C_1 structures to form Ξ layers [the gates on the left of the barrier in circuit (14)] may be cancelled out, if those commute with the structure between them, which corresponds to C_2. The identities described in circuits (H2) and (H3) may be used in some cases to increase the number of gates that can be commuted and cancelled out in this way. In addition, similarly to the ATA case, some of the gates commuting toward the edges of the full MCSU2 structure can be removed by applying gates that commute with the MCSU2 on both sides as follows.



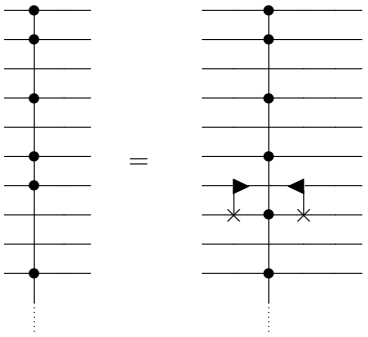
(G1)

This allows for the following cancellations between the added gates, and the ones commuted to the left from circuit (14) (similarly for the right side of the MCSU2).



(G2)

Therefore, each "useful" ancilla allows the removal of eight T gates, four CNOT gates, and four H gates. These cancellations apply when the ancilla qubit is located directly above the control qubit. However, if the ancilla was originally located directly below the control, it can be swapped as follows.



(G3)

This has two benefits. First, as can be seen, the two remaining CNOT gates in circuit (G2) can be cancelled out, thus reducing the CNOT cost by additional 4. Moreover, this swap may increase the number of C_1 qubits located above all C_2 ones, thus allowing for more such cancellations by making more ancilla qubits "useful," and at the same time reduce k_2, thus reducing the number of CNOT gates even further.

From this arises an important question: In which cases is it beneficial to apply SWAP gates? As can be seen, if one wishes to minimize the number of T gates, and is willing to "pay" with a quadratic increase of CNOT gates, then many SWAP chains can be applied in order to choose the qubit ordering that maximizes the number of useful ancilla. Otherwise, one might compare the costs of each possible qubit reordering and select the most efficient result, considering all Clifford+T gates. As this might be expensive in terms of classical computation, we identify some cases in which it is always beneficial to reorder the qubits.

Initially, both k_1 and k_2 may be reduced by relocating the bottom $\{C, t\}$ qubit upward toward the nearest $\{C, t\}$ one above it, using two partial SWAP chains. Similarly, k_1 can be reduced by relocating the top qubit downwards, unless this results in an increase of k_2.

Finally, one can easily identify the first pair of neighboring control qubits that do not include $C[1]$, and the nearest ancilla qubit below this pair. As mentioned above, if the ancilla is directly below this pair, it is always beneficial to apply a swap, and iteratively continue to identify the next neighboring pair of controls. In the case that the ancilla is not directly below, it is beneficial to relocate it upward if the number of added CNOT gates applied for this relocation is no larger than the number of CNOT gates which can be removed due to this relocation.

We consider the specific control permutation $\Omega := [1, 1, 1, 0, 1, 0, 1, \ldots, 0, 1, 0, 0]$ such that the target qubit t is at the end. After one iteration of the mentioned procedure, a SWAP(q_3, q_4) will be applied, and the new order will be $\Omega' = [1, 1, 0, 1, 1, 0, 1, \ldots, 0, 1, 0, 0]$, then SWAP$(q_5, q_6)$ will be applied, and so on. Finally, the permutation will be $\Omega^{(n-2)} = [1, 1, 0, 1, 0, 1, 0, \ldots, 1, 0, 1, 0]$. This case assumes $k = 2n - 1$, i.e., $n - 2$ available dirty ancilla qubits, and allows for all $[X_\Delta]$ gates to be implemented as the gates to the right of the barrier in circuit (14), or their inverse.

Furthermore, all applied SWAP gates are automatically cancelled during the process. In this case, $k_2 = n_2 = 0$ and $l'_{0,1} = 1$. The complete structure will require two SWAP-$[iZ]$ gates and $4(n-2)$ reduced $[X_\Delta]$ gates in addition to five $\{R_{\hat{x}}, R_{\hat{z}}\}$ gates. We note that, since $n_2 = 0$, in this case the following applies for $n \geq 3$.

Theorem G1. Any multicontrolled $SU(2)$ gate with $n \geq 3$ controls can be implemented over $k = 2n - 1$ qubits, ordered as Ω or its reversed string, in LNN connectivity such that the target qubit t is at the top or bottom, using five gates from $\{R_{\hat{x}}, R_{\hat{z}}\}$, in addition to

CNOT cost $12n - 12$ and depth $12n - 12$,
T cost $8n - 8$ and depth $4n - 4$,
H cost $4n - 8$ and depth $4n - 8$.

This can be considered as the best-case scenario; however, slightly different permutations may result in a reduction of some costs and/or depths while increasing others.

We cover an additional specific case in which all qubits in the circuit are used, i.e., $k = n + 1$, and the target is located at the top or bottom. For this case, $C = [1, 1, 1, \ldots, 1, 1, 0]$. It is clear that $l'_{0,1} = l'_{0,2} = 1$, $k_1 = k$, and $k_2 = k - 2$. The following result can be achieved by following the steps presented in Sec. III A 2.

Theorem G2. Any multicontrolled $SU(2)$ gate with $n \geq 6$ controls can be implemented over $k = n + 1$ qubits in LNN connectivity such that the target qubit t is at the top or bottom, using eight gates from $\{R_{\hat{x}}, R_{\hat{z}}\}$, in addition to

CNOT cost $20n - 48$ and depth $12n$,
T cost $16n - 48$ and depth $4n$,
H cost $8n - 32$ and depth $4n - 11$.

The costs and depths of MCX and MCU2 with one dirty or clean ancilla can be obtained in a similar way. As the structures used for these gates are based on the MCSU2 implementation, it is clear that in these cases the leading terms are the same as mentioned above.

APPENDIX H: LNN MULTICONTROLLED MULTITARGET *SU*(2)

We provide an implementation for any LNN restricted MCMTSU2 gate with a set $C = \{c_1, c_2, \ldots, c_n\}$ of n controls and $\tau = \{t_1, t_2, \ldots, t_m\}$ of m targets implemented over a set $Q' = \{q_1, q_2, \ldots, q_{k'}\}$ of LNN qubits. The set $Q \in Q'$ of size k is defined in this case as the smallest set of LNN qubits that satisfies $C \cup \tau \in Q$ and $q_k \notin C$.

At first we assume that each target qubit in τ is located below all qubits in C, i.e., $C^j = C$ for any $q_j \in \tau$. In this case, recalling the ATA MCMTSU2 case in Lemma C1, the structure provided for a single target LNN MCSU2 can be directly used, along with $8(m-1)$ additional single-qubit rotations from $\{R_{\hat{x}}, R_{\hat{z}}\}$ in order to implement the LNN MCMTSU2. This is achieved since $\{\overline{Z}\}^{k_1}_{C_1,Q_1}$ and $\{\overline{Z}\}^{k_2}_{C_2,Q_2}$ include $\prod_{t\in\tau}[Z]^{C_1}_t$ and $\prod_{t\in\tau}[Z]^{C_2}_t$, respectively, and no other MCZ gate is applied on qubits from τ.

However, in order to guarantee that all target qubits are at the bottom of the circuit, many SWAP chains must be applied, and, in a random qubit permutation, this may result in a quadratic overhead of $O(mk)$ CNOT gates. In order to maintain a linear ($O(k)$) cost for each basic gate type, we provide a method that can be applied for any permutation in which $q_k \notin C$. In the case that $q_k \in C$, two partial SWAP chains can be used in order to relocate one qubit as before. Since, in this case, any noncontrol qubit can be chosen, the maximal distance from the nearest edge is $\lfloor n/2 \rfloor$ qubits. The CNOT cost overhead in the worst case is $4\lfloor n/2 \rfloor$ in depth $2\lfloor n/2 \rfloor + 4$, as shown in Appendix F.

The macrostructure in this case is similar to circuit (C1), with $[\Delta]$ gates applied on subsets of $Q \backslash \tau$, and the subsets C_1 and C_2 are defined similarly to the LNN MCSU2 case with one additional condition: for each of the subsets C_1 and C_2 it must hold that any consecutive subset of target qubits from τ may not have two neighboring control qubits.

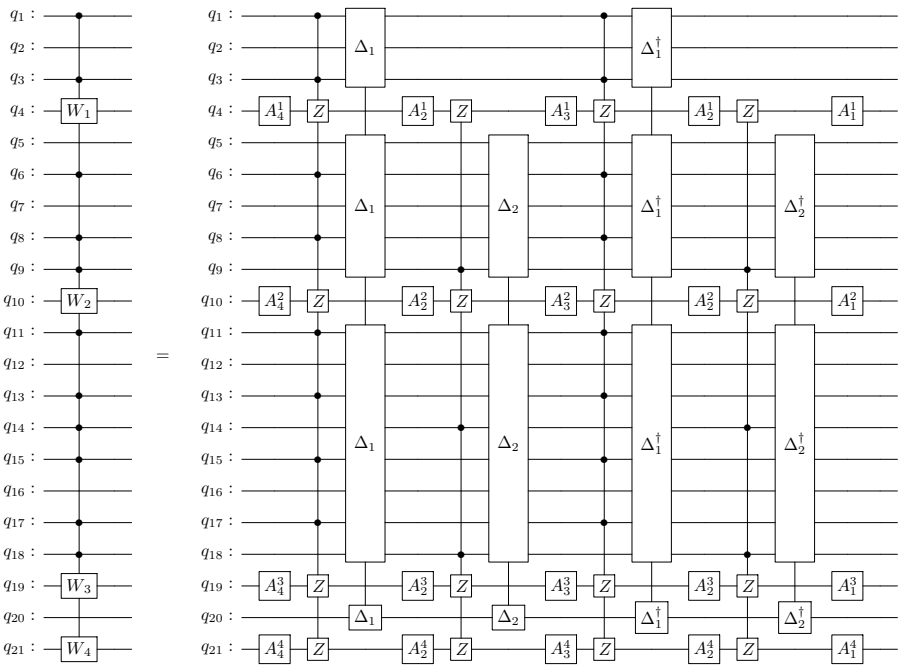
A similar procedure to the one defined for MCSU2 can be used while treating the qubits above and below consecutive targets as nearest neighbors. In addition, in order to simplify this case, we are not allowing for any nearest-neighboring controls in either subset, which in turn removes the $[iZ]$ gate from this structure.

Start with $C_1 = \emptyset$, $C_2 = \emptyset$, the iteration index $l = 1$, and an additional index $l' = 1$. At each iteration, q_l is added to C_1 if

$$q_l \in C \wedge q_{l'} \notin C_1,$$

such that the qubit $q_{l'}$ holds the previous nontarget qubit if $l > 1$. If this condition is not met and $q_l \in C$, the qubit q_l is added to C_2. If $q_l \notin \tau$, the index l' is then set to the value of l. Finally, l is increased by 1 for the next iteration, stopping when $C_1 \cup C_2$ holds all of the control qubits in C, i.e., $n_1 + n_2 = n$.

The following circuit provides an example of the macrostructure achieved for an arbitrary choice of C, τ and Q.

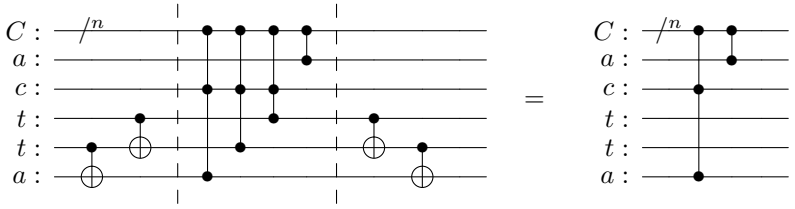


(H1)

We now wish to decompose the $[\Delta_1]_{\mathcal{Q}_1\setminus\tau}\prod_{j=1}^{m}[Z]_{t_j}^{C_1}$ and $[\Delta_2]_{\mathcal{Q}_2\setminus\tau}\prod_{j=1}^{m}[Z]_{t_j}^{C_2}$ gates. We use the $\{\overline{Z}\}_{C_1,\mathcal{Q}_1}^{k_1}$ and $\{\overline{Z}\}_{C_2,\mathcal{Q}_2}^{k_2}$ decompositions as in Lemma 12, and apply CNOT transformations in two steps.

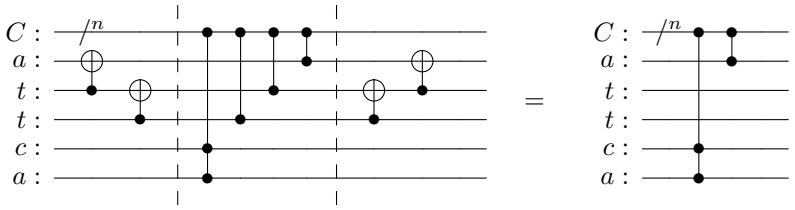
First we remove any $[Z]_{q_j}^{C_1^j}$ and $[Z]_{q_j}^{C_2^j}$ for $q_j \in \tau$ if $C_1^j \neq C_1$ or $C_2^j \neq C_2$, respectively. This can be achieved due to our choice of C_1 and C_2 such that sequential target qubits cannot have more than one neighboring control qubit in either control subset.

In the case that the qubit below the target sequence *is not* a control, we can use the following.



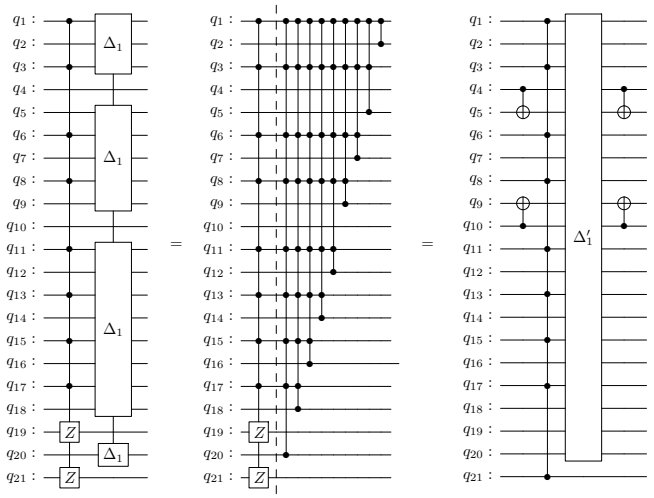
(H2)

And if the qubit below *is* a control (and the qubit above is not a control), we use the following.



(H3)

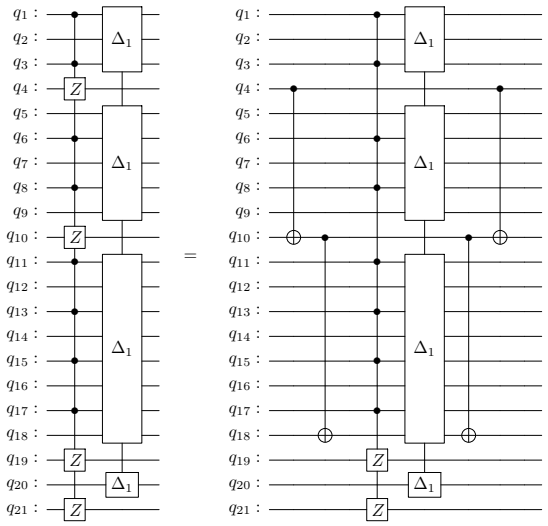
In our example, $\{\overline{Z}\}_{C_1,\mathcal{Q}_1}^{k_1} = [\Delta_1']_{\mathcal{Q}_1\setminus t_m}[Z]_{t_m}^{C_1}$ is converted to $[\Delta_1]_{\mathcal{Q}_1\setminus\tau}[Z]_{t_m}^{C_1}[Z]_{t_{m-1}}^{C_1}$ as follows.



(H4)

This process adds two CNOT gates per target which we "disconnect", and therefore increases the CNOT cost and depth by no more than $2m$.

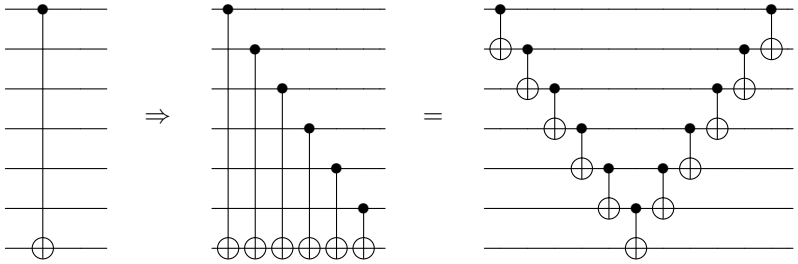
Once these target qubits are not affected by any gate, we wish to add $[Z]_{q_j}^{C_1}$ or $[Z]_{q_j}^{C_2}$ gates for each $q_j \in \tau$ qubit that is located above the bottom control qubit. Using the identity described in circuit (C2), a pair of CNOT gates can be applied on both sides of the structure in order to "add a target" to the MCMTZ gate. For each added target, we utilize the nearest qubit on which an MCZ gate controlled by the entire control subset is already applied. The first pair of CNOT gates will therefore be applied on the qubit neighboring below the bottom control, and the nearest target qubit above it. The second pair will be applied on this added target and the nearest one above it, and so on. The following demonstrates this on the structure controlled by C_1 in our example.



(H5)

This requires long-range CNOT gates, which can be achieved using SWAP gates. However, in order to lower the cost, we can allow for additional CNOT gates to be implemented, if those only result in relative-phase gates applied on $Q\backslash\tau$.

If a pair of CNOT gates controlled by a qubit from $Q\backslash\{C_1,\tau\}$ (the same holds for C_2) is applied on a qubit from $Q\backslash C_1$, this effectively adds an MCZ gate controlled by C_1, and applied on the $Q\backslash\{C_1,\tau\}$ qubit, and therefore it can be included in the $[\Delta]$ gate. Moreover, if a CNOT gate controlled by a qubit from C_1 is applied on a qubit from $Q\backslash C_1$, this effectively adds an MCZ gate, which is only applied on C_1, and thus can be included in the $[\Delta]$ gate as well. The following can therefore be used to replace long-range CNOT gates in this case.



(H6)

Defining d as the distance from the control qubit to the target qubit of the long-range CNOT gate, the cost and depth of this structure is $2d-1$. In an arbitrary permutation, this structure can be applied up to $2m$ times for the implementation for each MCMTZ gate, at a total cost and depth of $2\sum_{j=1}^{m}(2d_j-1)$, with d_j as the distance of each long-range CNOT pair, such that $\sum_{j=1}^{m}(d_j)\leq k$. The added cost and depth of CNOT gates for this step is therefore no larger than $4k-2m$.

Adding all of these steps together, the MCMTSU2 costs and depths are equivalent to the MCSU2 ones, with $l'_{0,1}=l'_{0,2}=0$, such that CNOT cost and depth are increased by no more than $16k$, and $8(m-1)$ gates from $\{R_{\hat{x}},R_{\hat{z}}\}$ are added. Since neighboring controls are not allowed in C_1, we can only guarantee $q_1\notin C_2$, and we can write $k_2\leq k-1$. Adding the CNOT overhead of $4\lfloor n/2\rfloor$ in depth $2\lfloor n/2\rfloor+4$, the following is achieved similarly to Theorem 4.

Theorem H1. Any multicontrolled multitarget $SU(2)$ gate with $n\geq 6$ controls and $m\geq 1$ targets can be implemented over $k\geq n+m$ qubits in any permutation in LNN connectivity using $8m$ gates from $\{R_{\hat{x}},R_{\hat{z}}\}$, in addition to no more than

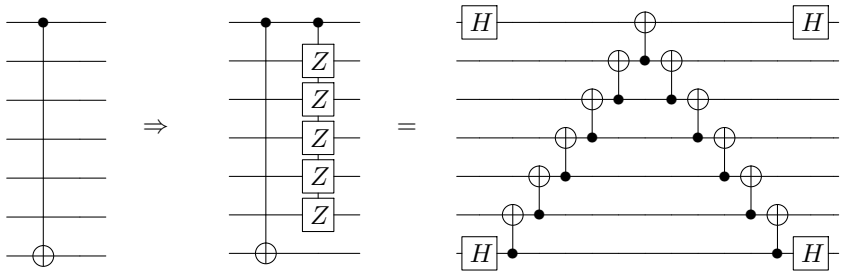
CNOT cost $24k+14n-40$ and depth $24k+5n-4$,
T cost $16n-32$ and depth $4n$,
H cost $8n-8$ and depth $4n$.

We note that some depth reductions are achieved in many cases, considering the orientation of sequential "V"-shaped structures such as circuits (H6) and (12). The structure in circuit (I1) can be used instead of circuit (H6) in some cases in order to reduce the CNOT depth, while increasing the count of H gates.

APPENDIX I: LNN MCMTX

Similarly to the ATA case, if the number of targets is larger than one, an ancilla qubit is not required. We assume that a noncontrol qubit is located at the bottom and treat it as an ancilla qubit in order to implement an MCZ gate applied on the set $\{C,q_j\}$ such that $q_j\in\tau$ is the lowest target qubit that satisfies $j<k$. The MCZ gate is implemented using Theorem 5. Each additional target qubit is added similarly to the ATA case by applying a CNOT gate on each side. In the LNN case we will add a new target using the closest noncontrol qubit to it, on which an MCZ

controlled by C is already implemented. It is clear that, in this case, the long-range CNOT gates can be implemented up to a relative phase as follows.



(I1)

Finally, a pair of H gates is applied on each target qubit in order to transform the MCMTZ to MCMTX. Each long-distance CNOT gate is implemented using four H gates. However, many of these gates cancel out, resulting in an addition of no more than $2m-2$ H gates. Noticing the similarities to the process described in Appendix H, it can be realized that, in addition to the costs described in Theorem 5, this process requires no more than $4k-2m$ CNOT gates.

Theorem I1. Any multicontrolled multitarget X gate with $n \geq 5$ controls and $m \geq 2$ can be implemented over $k \geq n+m$ qubits in any permutation in LNN connectivity using no more than

CNOT cost $12k+14n-2m-34$
and depth $12k+5n-2m+1$,
T cost $16n-16$ and depth $4n+4$,
H cost $8n+2m+2$ and depth $4n+2m+1$.

[1] M. Lewis, S. Soudjani, and P. Zuliani, Matrix representation of arbitrarily controlled quantum gates, ArXiv:2205.02525.

[2] B. Ömer, Classical concepts in quantum programming, Int. J. Theor. Phys. **44**, 943 (2005).

[3] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, Phys. Rev. A **52**, 3457 (1995).

[4] J. M. Arrazola, O. D. Matteo, N. Quesada, S. Jahangiri, A. Delgado, and N. Killoran, Universal quantum circuits for quantum chemistry, Quantum **6**, 742 (2022).

[5] J. H. A. De Carvalho, and F. M. D. P. Neto, Parametrized constant-depth quantum neuron, IEEE Trans. Neural Networks Learn. Syst. , 1 (2024).

[6] X.-H. Ni, L.-X. Li, Y.-Q. Song, Z.-P. Jin, S.-J. Qin, and F. Gao, Progressive quantum algorithm for maximum independent set with quantum alternating operator ansatz, Chin. Phys. B **34**, 070304 (2025).

[7] R. Iten, R. Colbeck, I. Kukuljan, J. Home, and M. Christandl, Quantum circuits for isometries, Phys. Rev. A **93**, 032318 (2016).

[8] E. Malvetti, R. Iten, and R. Colbeck, Quantum circuits for sparse isometries, Quantum **5**, 412 (2021).

[9] D. Grinko, A. Burchardt, and M. Ozols, Efficient quantum circuits for port-based teleportation, ArXiv:2312.03188.

[10] A. Tănăsescu, D. Constantinescu, and P. G. Popescu, Distribution of controlled unitary quantum gates towards factoring large numbers on today's small-register devices, Sci. Rep. **12**, 21310 (2022).

[11] D. K. Park, F. Petruccione, and J.-K. K. Rhee, Circuit-based quantum random access memory for classical data, Sci. Rep. **9**, 3949 (2019).

[12] A. Peres, Reversible logic and quantum computers, Phys. Rev. A **32**, 3266 (1985).

[13] L. K. Grover, Quantum mechanics helps in searching for a needle in a haystack, Phys. Rev. Lett. **79**, 325 (1997).

[14] M. B. Ali, T. Hirayama, K. Yamanaka, and Y. Nishitani, Function design for minimum multiple-control Toffoli circuits of reversible adder/subtractor blocks and arithmetic logic units, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E101.A**, 2231 (2018).

[15] A. Shafaei, M. Saeedi, and M. Pedram, Cofactor sharing for reversible logic synthesis, ACM J. Emerg. Technol. Comput. Syst. **11**, 1 (2014).

[16] M. A. Nielsen, and I. L. Chuang, *Quantum Computation and Quantum Information, 10th Anniversary Edition* (Cambridge, England, 2010).

[17] V. V. Shende, S. S. Bullock, and I. L. Markov, in *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, ASP-DAC '05 (Association for Computing Machinery, New York, NY, USA, 2005), p. 272.

[18] L. Isenhower, M. Saffman, and K. Mølmer, Multibit CkNOT quantum gates via Rydberg blockade, Quantum Inf. Process. **10**, 755 (2011).

[19] D. Petrosyan, S. Norrell, C. Poole, and M. Saffman, Fast measurements and multiqubit gates in dual-species atomic arrays, Phys. Rev. A **110**, 042404 (2024).

[20] S. Jandura, and G. Pupillo, Time-optimal two- and three-qubit gates for Rydberg atoms, Quantum **6**, 712 (2022).

[21] M. AbuGhanem, IBM quantum computers: Evolution, performance, and future directions, J. Supercomput. **81**, 687 (2025).

[22] M. AbuGhanem, Hardware-aware Toffoli gate decomposition via echoed cross-resonance gates, Quantum Studies: Math. Found. **12**, 24 (2025).

[23] X. Xue, M. Russ, N. Samkharadze, B. Undseth, A. Sammak, G. Scappucci, and L. M. K. Vandersypen, Quantum logic with spin qubits crossing the surface code threshold, Nature **601**, 343 (2022).

[24] S. Bravyi, and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, Phys. Rev. A **71**, 022316 (2005).

[25] M. A. Nielsen, and I. Chuang, Quantum computation and quantum information, Am. J. Phys. **70**, 558 (2002).

[26] N. C. Jones, Ph.D. Thesis, Stanford University, California, 2013.

[27] M. Vasmer, and D. E. Browne, Three-dimensional surface codes: Transversal gates and fault-tolerant architectures, Phys. Rev. A **100**, 012312 (2019).

[28] F. Butt, S. Heußen, M. Rispler, and M. Müller, Fault-tolerant code-switching protocols for near-term quantum processors, PRX Quantum **5**, 020345 (2024).

[29] X. Cheng, Z. Guan, and W. Ding, Mapping from multiple-control Toffoli circuits to linear nearest neighbor quantum circuits, Quantum Inf. Process. **17**, 169 (2018).

[30] D. Schoenberger, S. Hillmich, M. Brandl, and R. Wille, Shuttling for scalable trapped-ion quantum computers, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **44**, 2144 (2025).

[31] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, J. P. Bonilla Ataides, N. Maskara, I. Cong, X. Gao, P. Sales Rodriguez, T. Karolyshyn, G. Semeghini, M. J. Gullans, M. Greiner, V. Vuletić, and M. D. Lukin, Logical quantum processor based on reconfigurable atom arrays, Nature **626**, 58 (2024).

[32] M. Pedram, and A. Shafaei, Layout optimization for quantum circuits with linear nearest neighbor architectures, IEEE Circuits Syst. Mag. **16**, 62 (2016).

[33] M. Saeedi, R. Wille, and R. Drechsler, Synthesis of quantum circuits for linear nearest neighbor architectures, Quantum Inf. Process. **10**, 355 (2011).

[34] X. He, Z. Guan, and F. Ding, The mapping and optimization method of quantum circuits for Clifford + T gate, J. Appl. Math. Phys. **07**, 2796 (2019).

[35] M. H. A. Khan, Cost reduction in nearest neighbour based synthesis of quantum Boolean circuits, Eng. Lett. **16**, 1 (2008).

[36] M. Lukac, P. Kerntopf, and M. Kameyama, Optimization of LNN reversible circuits using an analytic sifting method, J. Circ. Syst. Comput. **30**, 2150166 (2021).

[37] F. Ding, Z. Guau, and F. Ren, in *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)* (IEEE, Singapore, 2019), p. 73.

[38] C. Zhang, Z. Guan, Y. Qian, and S. Feng, in *2022 International Conference on Computing, Communication, Perception and Quantum Technology (CCPQT)* (IEEE, Xiamen, China, 2022), p. 402.

[39] R. Vale, T. M. D. Azevedo, I. C. S. Araújo, I. F. Araujo, and A. J. da Silva, Circuit decomposition of multicontrolled special unitary single-qubit gates, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **43**, 802 (2024).

[40] 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. **56**, 2350 (2017).

[41] S. Balauca, and A. Arusoaie, in *Computational Science – ICCS 2022*, Lecture Notes in Computer Science, edited by D. Groen, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot (Springer International Publishing, Cham, 2022), p. 179.

[42] D. Maslov, and M. Saeedi, Reversible circuit optimization via leaving the Boolean domain, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **30**, 806 (2011).

[43] A. Kole, and K. Datta, in *2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID)* (IEEE, Hyderabad, India, 2017), p. 289.

[44] Z. Sasanian, and D. M. Miller, in *Proceedings of 2011 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing* (IEEE, Victoria, BC, Canada, 2011), p. 567.

[45] L. Biswal, C. Bandyopadhyay, R. Wille, R. Drechsler, and H. Rahaman, in *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)* (IEEE, Kolkata, India, 2016), p. 573.

[46] M. B. Ali, T. Hirayama, K. Yamanaka, and Y. Nishitani, in *2015 International Conference on Computational Science and Computational Intelligence (CSCI)* (IEEE, Las Vegas, NV, USA, 2015), p. 59.

[47] P. Niemann, A. Gupta, and R. Drechsler, in *2019 IEEE 49th International Symposium on Multiple-Valued Logic (ISMVL)* (IEEE, Fredericton, NB, Canada, 2019), p. 108.

[48] J. Leng, F. Yang, and X.-B. Wang, Decomposing $(n+1)$-qubit Toffoli gate with shallow circuit depth and no ancilla, Adv. Quantum Technol. **7**, 2300370 (2024).

[49] N. Abdessaied, M. Amy, M. Soeken, and R. Drechsler, in *2016 IEEE 46th International Symposium on Multiple-Valued Logic (ISMVL)* (IEEE, Sapporo, Japan, 2016), p. 150.

[50] V. V. Arsoski, Implementing multicontrolled X gates using the quantum Fourier transform, Quantum Inf. Process. **23**, 305 (2024).

[51] V. V. Arsoski, Multi-controlled single-qubit unitary gates based on the quantum Fourier transform and deep decomposition, J. Supercomput. **81**, 1202 (2025).

[52] A. Chakrabarti, and S. Sur, Nearest neighbour based synthesis of quantum Boolean circuits, Eng. Lett. **15**, 356 (2007).

[53] D. M. Miller, R. Wille, and Z. Sasanian, in *2011 41st IEEE International Symposium on Multiple-Valued Logic* (IEEE, Tuusula, Finland, 2011), p. 288.

[54] Y. Li, W. Liu, M. Li, and Y. Li, Quantum circuit compilation for nearest-neighbor architecture based on reinforcement learning, Quantum Inf. Process. **22**, 295 (2023).

[55] Y.-y. Tan, X.-y. Cheng, Z.-j. Guan, Y. Liu, and H. Ma, Multi-strategy based quantum cost reduction of linear nearest-neighbor quantum circuit, Quantum Inf. Process. **17**, 61 (2018).

[56] J. D. S. Silva, T. M. D. Azevedo, I. F. Araujo, and A. J. da Silva, Linear decomposition of approximate multicontrolled single qubit gates, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **44**, 1304 (2025).

[57] A. J. da Silva, and D. K. Park, Linear-depth quantum circuits for multiqubit controlled gates, Phys. Rev. A **106**, 042602 (2022).
[58] M. Saeedi, and M. Pedram, Linear-depth quantum circuits for n-qubit Toffoli gates with no ancilla, Phys. Rev. A **87**, 062318 (2013).
[59] C. Gidney, Constructing large controlled Nots, https://algassert.com/circuits/2015/06/05/Constructing-Large-Controlled-Nots.html.
[60] P. M. Q. Cruz, and B. Murta, Shallow unitary decompositions of quantum Fredkin and Toffoli gates for connectivity-aware equivalent circuit averaging, APL Quantum **1**, 016105 (2024).
[61] J. Gwinner, M. Briański, W. Burkot, L. Czerwiński, and V. Hlembotskyi, Benchmarking 16-element quantum search algorithms on superconducting quantum processors, ArXiv:2007.06539.
[62] K. M. Nakanishi, T. Satoh, and S. Todo, Decompositions of multiple controlled-Z gates on various qubit-coupling graphs, Phys. Rev. A **110**, 012604 (2024).
[63] K. M. Nakanishi, T. Satoh, and S. Todo, Quantum-gate decomposer, ArXiv:2109.13223.
[64] N. A. Nemkov, E. O. Kiktenko, I. A. Luchnikov, and A. K. Fedorov, Efficient variational synthesis of quantum circuits with coherent multi-start optimization, Quantum **7**, 993 (2023).
[65] G. Song, and A. Klappenecker, Optimal realizations of controlled unitary gates, Quantum Inf. Comput. **3**, 139 (2003).
[66] B. Zindorf, and S. Bose, All you need is pi: Quantum computing with Hermitian gates, ArXiv:2402.12356.
[67] S. Heußen, D. F. Locher, and M. Müller, Measurement-free fault-tolerant quantum error correction in near-term devices, PRX Quantum **5**, 010333 (2024).
[68] B. Claudon, J. Zylberman, C. Feniou, F. Debbasch, A. Peruzzo, and J.-P. Piquemal, Polylogarithmic-depth controlled-NOT gates without ancilla qubits, Nat. Commun. **15**, 5886 (2024).
[69] J. Nie, W. Zi, and X. Sun, Quantum circuit for multi-qubit Toffoli gate with optimal resource, ArXiv:2402.05053.
[70] C. Figgatt, A. Ostrander, N. M. Linke, K. A. Landsman, D. Zhu, D. Maslov, and C. Monroe, Parallel entangling operations on a universal ion-trap quantum computer, Nature **572**, 368 (2019).
[71] J. Liu, M. Bowman, P. Gokhale, S. Dangwal, J. Larson, F. T. Chong, and P. D. Hovland, in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)* (IEEE, Monterey, CA, USA, 2023), p. 1.
[72] F. d. Meijer, D. Gijswijt, and R. Sotirov, Exploiting symmetries in optimal quantum circuit design, ArXiv:2401.08262.
[73] B. Giles, and P. Selinger, Exact synthesis of multiqubit Clifford+T circuits, Phys. Rev. A **87**, 032332 (2013).
[74] D. Maslov, Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization, Phys. Rev. A **93**, 022311 (2016).
[75] D. Maslov, G. Dueck, and D. Miller, Toffoli network synthesis with templates, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **24**, 807 (2005).
[76] D. Maslov, and B. Zindorf, Depth optimization of CZ, CNOT, and Clifford circuits, IEEE Trans. Quantum Eng. **3**, 1 (2022).