

Unitary Synthesis with Fewer T Gates

Xinyu Tan^{*}

*Google Quantum AI, Venice, California, 90291, USA
and Department of Mathematics, Massachusetts Institute of Technology,
Cambridge, Massachusetts, 02139, USA*



(Received 21 February 2026; revised 15 June 2026; accepted 10 August 2026; published 16 September 2026)

We present a simple algorithm that implements an arbitrary n -qubit unitary operator using a Clifford + T circuit with T-count $O(2^{4n/3}n^{2/3})$ and ancilla-count $O(2^{2n/3}n^{1/3})$. The previous best T-count for unitary synthesis was $O(2^{3n/2}n)$ with $O(2^{n/2})$ ancillae by [Low *et al.*, [Quantum 8, 1375 \(2024\)](#)]. There is a fundamental tradeoff between T gates and ancillae, and our algorithm extends the achievable regime of this tradeoff, reducing the exponent in T-count from $3n/2$ to $4n/3$. The best known lower bound remains $\Omega(2^n)$, so the optimal T-count for general unitary synthesis is still open. Our construction is based on a recursive application of the cosine-sine decomposition, together with a generalization of the optimal diagonal unitary synthesis method by [D. Gosset *et al.*, [Quantum 10, 2168 \(2026\)](#)] to multi-controlled k -qubit unitaries.

DOI: [10.1103/pxhd-9s9q](https://doi.org/10.1103/pxhd-9s9q)

I. INTRODUCTION

Decomposing arbitrary unitaries into smaller, structured circuits is a fundamental and well-known problem in quantum computation. This question is not only mathematically interesting—aiming to understand how complex unitary transformations arise from simple building blocks—but also of practical significance, as it forms the basis of quantum circuit compilation. There is a long history: early work studied exact synthesis using continuous gate sets, such as single-qubit rotations together with CNOT gates [3,4]. Researchers have also investigated discrete universal gate sets as they can approximate any unitary to arbitrary precision, with the Solovay-Kitaev theorem providing the first general efficiency guarantee for single-qubit approximation [5].

Among discrete universal gate sets, particular attention has been given to decompositions into *Clifford + T* circuits. This focus is motivated by fault-tolerant quantum computing: Clifford operations can typically be implemented at low cost, while T gates are expensive due to the overhead of magic state distillation and injection. It is therefore natural to ask *how arbitrary unitaries can be compiled using as few T gates as possible, and what the optimal T-count is.*

Although this general question remains open, progress has been made in certain special cases. One example is the decomposition of single-qubit unitaries into Clifford + T circuits [6–8], restated in Lemma II.4 and used as a subroutine in our work. In this setting, the focus has been on optimizing constant factors, which are crucial for practical implementations.

Another important case is quantum state preparation, which corresponds to implementing the first column of an n -qubit unitary. With only a constant number of ancillae, the T-count for quantum state preparation can be shown to have a lower bound of $\Omega(2^n/n)$ [9] using a counting argument [1]. Surprisingly, it was shown that if exponentially many ancillae are allowed, the number of T gates can be substantially smaller than the number of the Clifford gates [1], and an optimal T-count of $\Theta(2^{n/2})$ was recently established in Ref. [2].

For general unitary synthesis, when only a constant number of ancillae is available, a lower bound of $\Omega(2^{2n}/n)$ can be derived in a similar manner. However, [1] showed that one can achieve a T-count of $O(2^{3n/2} \cdot n)$ using $O(2^{n/2})$ ancillae, by reducing the task to preparing 2^n quantum states. This was the first result demonstrating an asymptotic saving in T gates for unitary synthesis, and it appeared in 2018. Subsequently, in 2021, Rosenthal proved a query upper bound (with queries to classical Boolean functions) of $O(2^{n/2})$, which translates into a T-count of roughly the same order, $O(2^{3n/2})$, while using $O(2^n)$ ancillae [10]. For a summary of these results and other related scalings, see Table I. Since then, no further improvements in synthesis algorithms have been obtained.

^{*}Contact author: norahtan@mit.edu

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.

TABLE I. Comparison of related T-count and ancilla-count scalings for constant approximation error.

Task	Result	T-count	Ancillae
State preparation	Low-Kliuchnikov-Schaeffer [1]	$O(2^{n/2} \sqrt{n \log n})$	$O(2^{n/2} \sqrt{\log n/n})$
	Gosset-Kothari-Wu [2]	$\Theta(2^{n/2})$	$O(2^{n/2})$
Unitary synthesis	Counting lower bound	$\Omega(2^{2n}/(n + \lambda))$	λ
	Low-Kliuchnikov-Schaeffer [1]	$O(2^{3n/2} n)$	$O(2^{n/2})$
	Rosenthal [10]	$O(2^{3n/2})$	$O(2^n)$
	This work, the naive version (Sec. III A)	$O(2^{3n/2} n^{1/2})$	$O(2^{n/2} n^{1/2})$
	This work (Theorem I.1)	$O(2^{4n/3} n^{2/3})$	$O(2^{2n/3} n^{1/3})$

Given arbitrarily many ancillae, the current best lower bound is $\Omega(2^n)$, given in Ref. [2] (restated in Theorem I.3 for completeness). Determining the optimal T-count for unitary synthesis remains an open problem. Two natural candidate scalings have been discussed informally: $2^{3n/2}$, suggested by the best known upper bound from two distinct approaches, and 2^n , motivated by the analogy with quantum state preparation, which exhibits a quadratic saving.

In this paper, we give a unitary synthesis algorithm that reduces the T-count to $O(2^{4n/3} \cdot n^{2/3})$. This provides further evidence that the optimal scaling could be as low as 2^n .

Theorem I.1. (Main result). Let $\epsilon > 0$ and set $L = n + \log(1/\epsilon)$. Then any $U \in U(2^n)$ can be ϵ -approximated by a Clifford + T circuit using

$$O(2^{4n/3} \cdot L^{2/3} + 2^n \cdot L) \quad \text{T gates}$$

and

$$O(2^{2n/3} \cdot L^{1/3} + L) \quad \text{ancillae.}$$

In particular, for any positive integer $k \leq (n - \log_2 L)/3$, U can be ϵ -approximated by a Clifford + T circuit using

$$O(2^{(3n-k)/2} \cdot \sqrt{L}) \quad \text{T gates}$$

and

$$O(2^{(n+k)/2} \cdot \sqrt{L}) \quad \text{ancillae.}$$

When $k = O(1)$, Theorem I.1 gives a T-count of $O(2^{3n/2} \sqrt{L})$ and an ancilla-count of $O(2^{n/2} \sqrt{L})$. This corresponds to the naive recursive CS-decomposition approach described in Sec. III A, and is a minor refinement of the $O(2^{3n/2} n)$ scaling of [1] within the same $2^{3n/2}$ regime. The main improvement of the present work is obtained when k grows with n , which moves further along the ancilla–T-count tradeoff and yields the $O(2^{4n/3} L^{2/3})$ T-count.

The notion of approximation by a Clifford + T circuit in Theorem I.1 is defined as follows.

Definition I.2. (Clifford + T approximation). Let $U \in U(2^n)$. We say that U admits an exact Clifford + T implementation using ℓ T gates and m ancillae if there exists $C \in U(2^{n+m})$ such that C can be written as a product of ℓ T gates and arbitrarily many Clifford gates, and

$$U \otimes |0^m\rangle = C \cdot (I_{2^n} \otimes |0^m\rangle).$$

Given $V \in U(2^{n+m})$ and $\epsilon \geq 0$, we say that V implements U to error ϵ if

$$\|U \otimes |0^m\rangle - V \cdot (I_{2^n} \otimes |0^m\rangle)\| \leq \epsilon.$$

In particular, if V also admits an exact Clifford + T implementation, then we say that U can be ϵ -approximated by a Clifford + T circuit.

A. Proof overview

We begin by showing that any n -qubit unitary can be decomposed into a product of $2^n - 1$ multi-controlled single-qubit unitaries, obtained via a recursive application of the cosine-sine decomposition. Each step of the recursion halves the dimension and introduces multi-controlled rotations, which ultimately yields $2^n - 1$ such gates (Theorem III.5).

By carefully organizing this recursion, we observe that many of these controlled unitaries share the same target qubits and can therefore be grouped together. More concretely, let $k \in [n - 1]$. Then the product of certain consecutive blocks of $2^k - 1$ controlled unitaries has the same k -qubit target register (say, the first k qubits), and hence simplifies to a single multi-controlled k -qubit unitary. There are in total 2^{n-k} such consecutive blocks.

The remaining main technical step is to determine the T-count for implementing a multi-controlled k -qubit unitary. The inspiration comes from the algorithm of [2] for synthesizing diagonal unitaries, together with a polynomial factoring technique. A diagonal unitary is a special case of a multi-controlled single-qubit unitary. The algorithm of [2] delegates the application of Hadamard and T gates to a specific target controlled by the other $n - 1$

qubits to a Boolean function of $n - 1$ variables. This idea can be made more concrete via the following example. Suppose $U = \sum_{x \in \{0,1\}^{n-1}} |x\rangle\langle x| \otimes V_x$ is a multi-controlled single-qubit unitary where each $V_x \in U(2)$ can be written as a product of Hadamard and T gates:

$$V_x = H^{f_1(x)} \cdot T^{f_2(x)} \dots H^{f_{2L-1}(x)} \cdot T^{f_{2L}(x)},$$

with Boolean functions $f_i: \{0,1\}^{n-1} \rightarrow \{0,1\}$. If we can implement the corresponding Boolean function oracles $|x\rangle \otimes |0\rangle \mapsto |x\rangle \otimes |f_i(x)\rangle$, then U can be realized by applying each Hadamard or T gate controlled by the ancilla register holding $|f_i(x)\rangle$.

In our setting, each multi-controlled single-qubit unitary in the block can similarly be described by a Boolean function $f: \{0,1\}^n \rightarrow \{0,1\}$. Moreover, since their targets lie only on the first k qubits, we find it convenient to factor each Boolean function f as

$$f(x_1, \dots, x_n) = \sum_i g_i(x_1, \dots, x_k) \cdot h_i(x_{k+1}, \dots, x_n),$$

where each g_i is a polynomial in the first k variables and each h_i is a polynomial in the remaining $n - k$ variables. By treating g_i and h_i separately, we obtain a T-count of $O(2^{\frac{n+k}{2}} \sqrt{k} + 4^k \cdot k)$ for implementing a multi-controlled k -qubit unitary (Lemma IV.3). The ancilla overhead comes from this amortization: the workspace stores the shared Boolean-function values $g_{i,j,b}$ on the control variables and the monomial data for the target variables while the controlled Hadamard and T sequences are applied across the entire block. Thus, the workspace grows with k for the same reason the T-count decreases.

By setting $k \approx n/3$, this approach yields a substantial saving over the naïve strategy of decomposing each multi-controlled k -qubit unitary into a product of $2^k - 1$ multi-controlled single-qubit unitaries (Sec. III A).

B. The tradeoff between T gates and ancillae

Let λ denote the number of ancillae and R the number of T gates. There is a notable tradeoff between the space (ancilla-count) and time (T-count) complexity in all related synthesis algorithms.

For the state preparation problem, it was first observed in Ref. [1] and later refined by the algorithm in Ref. [2] that

$$(n + \lambda) \cdot R = \Omega(2^n),$$

and that there exists an algorithm achieving T-count $O(2^n/\lambda)$ when $\lambda = O(2^{n/2})$. This tradeoff is essentially optimal, since even with arbitrarily many ancillae there is a lower bound of $\Omega(2^{n/2})$ on the T-count [2, Theorem 4.1].

For general unitary synthesis, we can derive an analogous lower bound using the counting argument of [1]:

$$(n + \lambda) \cdot R = \Omega(2^{2n}). \quad (1)$$

In the regime $\lambda = O(2^{n/2})$, [1] gave an algorithm that uses λ ancillae and $O(2^{2n} \cdot n/\lambda)$ T gates. However, the regime of this tradeoff is far from optimal, since in principle λ could scale up to $O(2^n)$. Extrapolating this regime would then suggest the possibility of achieving a T-count as small as $O(2^n)$.

Our algorithm essentially extends this tradeoff to the larger regime $\lambda = O(2^{2n/3} \cdot n^{1/3})$. This also indicates that, in order to further reduce the T-count beyond the $2^{4n/3}$ scale, one would need a more sophisticated method that leverages substantially more ancillae, possibly up to $O(2^n)$; indeed, the lower bound in Eq. (1) forces $R = \Omega(2^{4n/3}/\text{poly}(n))$ at this ancilla scale.

A natural question is whether reducing T-count by using additional ancilla qubits is practically meaningful. We emphasize that this depends on the fault-tolerant architecture: in many settings, the dominant non-Clifford cost is the preparation and delivery of high-fidelity T states, whereas clean ancilla qubits and measurements are comparatively cheaper. From this perspective, trading ancillae for T gates is not merely a theoretical artifact, but a meaningful compiler-level degree of freedom.

C. Applications

Implementing arbitrary unitaries on n qubits is a common subroutine in quantum algorithms, appearing for example in first-quantized quantum simulation [11,12] and in the preparation of matrix product states [13–15]. Such unitaries are typically specified by 4^n matrix elements stored classically, and synthesizing them with minimal T-count is crucial for practical implementations. Once fault-tolerant quantum computers are available, unitary synthesis will form part of the standard compilation toolchain for higher-level algorithmic primitives. Our synthesis technique could thus be used to reduce the cost of these primitives, where classical preprocessing can guide quantum circuit construction. Furthermore, since T gates are expected to be relatively expensive on certain hardware platforms, such as neutral-atom architectures, lowering the T-count through our method may be especially impactful.

D. Lower bound

We include, for completeness, the current best known lower bound of $\Omega(2^n)$ on the T-count of unitary synthesis, and conjecture that it is tight when ϵ is constant.

Theorem I.3. ([2, Theorem 4.3]). There exists $U \in U(2^n)$ such that the following is true. For any integer $m \geq 0$, let $\mathcal{U}$ be the associated quantum channel given by $\mathcal{U}(\rho) := U\rho U^\dagger \otimes |0^m\rangle\langle 0^m|$. For any adaptive Clifford + T circuit $\mathcal{A}$ with $\|\mathcal{A} - \mathcal{U}\|_\diamond \leq \epsilon$, $\mathcal{A}$ must use $\Omega(2^n \cdot \sqrt{\log(1/\epsilon)} + \log(1/\epsilon))$ T gates. In particular, this T-count is the

expectation over the randomness in the measurement outcomes in $\mathcal{A}$ with worst-case input.

E. Notations

Throughout this paper, we use $i = \sqrt{-1}$ to denote the imaginary unit, I_N for the $N \times N$ identity matrix, and $[N] = \{1, 2, \dots, N\}$. $U(N)$ denotes the unitary group of all $N \times N$ unitary matrices and $SU(N)$ denotes the special unitary group of all $N \times N$ unitary matrices with determinant 1. We write $\|A\|$ for the operator norm of a matrix A and $\|A\|_\diamond$ for the diamond norm of a quantum channel $\mathcal{A}$.

II. PRELIMINARIES

In this section, we include a few lemmas that will be used frequently in this paper.

The proof of the first lemma below is fairly standard via a telescoping sum argument. We nevertheless include it in the Appendix for completeness.

Lemma II.1. (Composition error bound). Suppose that $V_i \in U(2^{n+m_i})$ implements $U_i \in U(2^n)$ to error ϵ for some integer $m_i \geq 0$. Let $m = \max_{i \in [L]} m_i$. Then $(V_1 \otimes I_{2^{m-m_1}}) \cdots (V_L \otimes I_{2^{m-m_L}}) \in U(2^{n+m})$ implements $U_1 \cdots U_L$ to error $L\epsilon$.

Given a bitstring $a \in \{0, 1\}^n$, denote by $\text{wt}(a)$ the number of 1's in a .

Lemma II.2. (Generating all monomials). The monomial-generating unitary of degree n given by

$$\begin{aligned} |x_1, \dots, x_n\rangle \otimes \bigotimes_{a \in \{0,1\}^n, \text{wt}(a) \geq 2} |y_a\rangle &\mapsto \\ |x_1, \dots, x_n\rangle \otimes \bigotimes_{a \in \{0,1\}^n, \text{wt}(a) \geq 2} |y_a \oplus x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}\rangle, \end{aligned}$$

where $x_1, \dots, x_n, y_a \in \{0, 1\}$, can be written exactly as a product of $2^n - n - 1$ Toffoli gates.

Proof. We trivially have all the degree-1 monomials $x_1, \dots, x_n$. The algorithm then generates each degree-2 monomial using one Toffoli gate. More concretely, to generate $x_{i_1} x_{i_2}$ for some $1 \leq i_1 < i_2 \leq n$, apply a Toffoli gate which is controlled on $|x_{i_1}\rangle$ and $|x_{i_2}\rangle$ and acts on $|y_a\rangle$ where $a \in \{0, 1\}^n$ has two 1's at positions i_1 and i_2 and 0's elsewhere, i.e. $a = e_{i_1} + e_{i_2}$. The algorithm can thus work recursively to generate all monomials. For each integer $i \in [2, n]$, the algorithm can generate all degree- i monomials by applying $\binom{n}{i}$ Toffoli gates controlled on the appropriate degree- $(i-1)$ monomials and degree-1 monomials. Overall, the number of Toffoli gates used to generate all degree- n monomials is $\sum_{i=2}^n \binom{n}{i} = 2^n - n - 1$. ■

Lemma II.3. (T-count for Boolean function oracles, [1, 16, Theorem 2]). Let $r \geq 1$ be an integer and $f: \{0, 1\}^n \rightarrow \{0, 1\}^r$ be an arbitrary Boolean function. Define U_f as the unitary mapping $|x\rangle|y\rangle$ to $|x\rangle|y \oplus f(x)\rangle$ for all $x \in \{0, 1\}^n$ and $y \in \{0, 1\}^r$. Then U_f admits an exact

Clifford + T implementation using $O(\sqrt{r \cdot 2^n})$ T gates and ancillae.

We also refer readers to [[2], Remark 2.2] for a nice proof sketch of [16].

Lemma II.4. (Single-qubit Clifford + T approximation, [8]). For any $\epsilon > 0$ and any $U \in SU(2)$, there exists $\tilde{U} \in SU(2)$ such that $\|U - \tilde{U}\| \leq \epsilon$ and $\tilde{U}$ is a product of $O(\log(1/\epsilon))$ Hadamard and T gates.

III. COSINE-SINE DECOMPOSITION

In this section, we recall the cosine-sine (CS) decomposition, and explain how it can be used recursively to factorize any unitary into multi-controlled single-qubit unitaries. This will serve as the structural backbone for our synthesis results.

We begin by clarifying a piece of terminology that will be used frequently throughout the paper.

Definition III.1. (Multi-controlled unitaries). Let k be a positive integer smaller than n . We call $U \in U(2^n)$ an $(n-k)$ -fold controlled k -qubit unitary if, up to a permutation of qubits,

$$U = \sum_{x \in \{0,1\}^{n-k}} |x\rangle\langle x| \otimes V_x, \quad \text{where } V_x \in U(2^k).$$

When the number of control qubits is clear from context, we simply refer to U as a multi-controlled k -qubit unitary. In particular, if $U \in U(2^n)$ is described this way, the number of control qubits is understood to be $n-k$.

The form $\sum_x |x\rangle\langle x| \otimes V_x$ is block-diagonal and naturally connects to the cosine-sine (CS) decomposition [17]. In this paper, we will only use a special case as summarized in Theorem III.2. We refer interested readers to [18, Sec. 2.1] for detailed illustration and a proof of the more general case.

Theorem III.2. (Special case of the CS decomposition). For any $U \in U(2^n)$, there exist $V_1, V_2, W_1, W_2 \in U(2^{n-1})$ and angles $\theta_1, \dots, \theta_{2^{n-1}} \in [0, \pi/2]$ such that

$$U = \underbrace{\begin{pmatrix} V_1 & \\ & V_2 \end{pmatrix}}_{:=V} \cdot \underbrace{\begin{pmatrix} C & S \\ S & -C \end{pmatrix}}_{:=D} \cdot \underbrace{\begin{pmatrix} W_1 & \\ & W_2 \end{pmatrix}}_{:=W}, \quad (2)$$

where $C = \text{diag}(\cos \theta_1, \dots, \cos \theta_{2^{n-1}})$ and $S = \text{diag}(\sin \theta_1, \dots, \sin \theta_{2^{n-1}})$.

We remark that

- (i) $V, W \in U(2^n)$ are block-diagonal and correspond to 1-fold controlled $(n-1)$ -qubit unitaries. They are both controlled by the first qubit and act on the remaining $n-1$ qubits;
- (ii) The middle block $D \in U(2^n)$ is a multi-controlled single-qubit unitary controlled by the last $n-1$ qubits and acts on the first qubit.

By applying the CS decomposition recursively to V and W , we obtain a factorization of any n -qubit unitary into $2^n - 1$ multi-controlled single-qubit unitaries. If many consecutive multi-controlled single-qubit unitaries share the same target, we can simplify the circuit by implementing their product at once, since their product is itself a multi-controlled single-qubit unitary. In our case, we cannot do this exactly, but when the target qubits belong to the same block, there are still opportunities to combine gates.

We note that the order in which the target qubits appear has a specific combinatorial structure. To better describe this order, we introduce the following notation.

Definition III.3. (Position of the least significant 1). For each $i \in [2^n - 1]$, let $t_n(i) \in [n]$ denote the position of the least significant 1 in the binary representation of i in n bits (with leading zeros allowed), where the most significant bit has index 1 and the least significant bit has index n [19].

For example, $t_n(1) = n$ since the binary representation of 1 is $0^{n-1}1$, and $t_n(2^n - 2) = n - 1$ since the binary representation of $2^n - 2$ is $1^{n-1}0$. When $n = 3$,

$$i: 1, 2, 3, 4, 5, 6, 7 \Rightarrow t_3(i): 3, 2, 3, 1, 3, 2, 3.$$

We will use the following simple proposition in the proof of Theorem III.5.

Proposition III.4. For each $i \in [2^{n-1} - 1]$, we have that $t_n(i) = t_n(i + 2^{n-1}) = t_{n-1}(i) + 1 \geq 2$.

Theorem III.5. (Recursive CS decomposition). For any $U \in \text{U}(2^n)$, there exist $U_1, \dots, U_{2^n-1} \in \text{U}(2^n)$ such that $U = U_1 U_2 \cdots U_{2^n-1}$, where each U_i is a multi-controlled single-qubit unitary targeting the qubit indexed by $t_n(i)$.

As an example, the target qubits of $U_1, \dots, U_{2^k}$ are among the last k qubits. So their product $U_1 U_2 \cdots U_{2^k}$ is a multi-controlled k -qubit unitary acting on the last k qubits.

Proof of Theorem III.5. We will prove the claim by induction on n .

The base case $n = 2$ follows directly from Theorem III.2: in Eq. (2), we have $U_1 = V$ and $U_3 = W$, both targeting at the second qubit since $t_2(1) = t_2(3) = 2$, and $U_2 = D$, which targets the first qubit since $t_2(2) = 1$.

For the induction step, assume the claim holds for $n - 1$ qubits. Given $U \in \text{U}(2^n)$, its CS decomposition has the form $U = VDW$ as in Eq. (2). By the induction hypothesis, each $V_j, W_j \in \text{U}(2^{n-1})$ can be decomposed into multi-controlled single-qubit unitaries: for $j = 1, 2$,

$$\begin{aligned} V_j &= V_{j,1} V_{j,2} \cdots V_{j,2^{n-1}-1}, \\ W_j &= W_{j,1} W_{j,2} \cdots W_{j,2^{n-1}-1}. \end{aligned}$$

Here each $V_{j,i}, W_{j,i}$ is an $(n - 1)$ -qubit unitary acting on qubit $t_{n-1}(i) \in [n - 1]$ where $i \in [2^{n-1} - 1]$.

For each $i \in [2^{n-1} - 1]$, define

$$U_i = \begin{pmatrix} V_{1,i} & \\ & V_{2,i} \end{pmatrix}$$

and

$$U_{i+2^{n-1}} = \begin{pmatrix} W_{1,i} & \\ & W_{2,i} \end{pmatrix}.$$

Each U_i (or $U_{i+2^{n-1}}$) inherits the control structure from $V_{j,i}$ (or $W_{j,i}$), with the first qubit acting as an additional control. Then by Proposition III.4, each U_i targets the qubit indexed by $t_{n-1}(i) + 1 = t_n(i)$ and is controlled by the remaining $n - 1$ qubits. Similarly, each $U_{i+2^{n-1}}$ targets the qubit indexed by $t_{n-1}(i) + 1 = t_n(i + 2^{n-1})$ and is controlled by the rest of the qubits.

Finally, let us set $U_{2^n-1} = D$, which is a multi-controlled single-qubit unitary acting on the first qubit and indeed $t_n(2^n - 1) = 1$. This completes the induction. ■

A. A naïve implementation using $O(2^{3n/2} \cdot n^{1/2})$ T gates

It was shown in Ref. [[2], Theorem 1.2] that any diagonal unitary $D \in \text{U}(2^n)$ can be ϵ -approximated by a Clifford + T circuit using $O(\sqrt{2^n \cdot \log(1/\epsilon)} + \log(1/\epsilon))$ T gates and ancillae. In fact, if one takes a closer look at the proof of [[2], Theorem 1.2], it directly works for any unitary in the form of

$$\sum_{x \in \{0,1\}^{n-1}} |x\rangle\langle x| \otimes R_x, \quad \text{as long as } R_x \in \text{SU}(2). \quad (3)$$

Hence, one can relax the theorem to any multi-controlled single-qubit unitary U , not just the diagonal ones. This is because U can be written as a product of a diagonal gate and a gate in the form of Eq. (3). We summarize this generalization as below.

Corollary III.6. (T-count for multi-controlled single-qubit unitaries). For any $\epsilon > 0$ and any multi-controlled single-qubit unitary $U \in \text{U}(2^n)$, there exists $\tilde{U} \in \text{U}(2^{n+1})$ such that $\tilde{U}$ implements U to error ϵ and $\tilde{U}$ admits an exact Clifford + T implementation using $O(\sqrt{2^n \cdot \log(1/\epsilon)} + \log(1/\epsilon))$ T gates and ancillae.

Proof. Any multi-controlled single-qubit unitary $U \in \text{U}(2^n)$, up to a permutation of qubits, can be written as

$$\begin{aligned}
U &= \sum_{x \in \{0,1\}^{n-1}} |x\rangle\langle x| \otimes V_x \quad (\text{each } V_x \in \text{U}(2)) \\
&= \sum_{x \in \{0,1\}^{n-1}} |x\rangle\langle x| \otimes (e^{i\theta_x} \cdot R_x) \quad (\text{each } R_x \in \text{SU}(2), \theta_x \in [0, 2\pi)) \\
&= \underbrace{\left(\sum_{x \in \{0,1\}^{n-1}} |x\rangle\langle x| \otimes (e^{i\theta_x} \cdot I_2) \right)}_{:=D} \cdot \underbrace{\left(\sum_{x \in \{0,1\}^{n-1}} |x\rangle\langle x| \otimes R_x \right)}_{:=R}.
\end{aligned}$$

It follows from the proof of [2], Theorem 1.2] that

- (i) there exists $\tilde{R} \in \text{SU}(2^n)$ such that $\tilde{R}$ implements R to error $\epsilon/2$, i.e. $\|R - \tilde{R}\| \leq \epsilon/2$.
- (ii) there exists $\tilde{D} \in \text{SU}(2^{n+1})$ such that $\tilde{D}$ implements D to error $\epsilon/2$.
- (iii) Both $\tilde{R}$ and $\tilde{D}$ admit exact Clifford + T implementations using $O(\sqrt{2^n \cdot \log(1/\epsilon)} + \log(1/\epsilon))$ T gates and ancillae.

By Lemma II.1, we know that $\tilde{U} = \tilde{D} \cdot (\tilde{R} \otimes I_2)$ implements $U = DR$ to error ϵ , which completes the proof. ■

Following from Theorem III.5, we know that any $U \in \text{U}(2^n)$ can be written as a product of $2^n - 1$ multi-controlled single-qubit unitaries, i.e. $U = U_1 U_2 \cdots U_{2^n-1}$. By Corollary III.6, for each $i \in [2^n - 1]$, there exists $\tilde{U}_i \in \text{U}(2^{n+1})$ such that $\tilde{U}_i$ implements U_i to error $\epsilon \cdot 2^{-n}$ and $\tilde{U}_i$ admits an exact Clifford + T implementation using

$$\begin{aligned}
&O\left(\sqrt{2^n \cdot \log(1/(\epsilon \cdot 2^{-n}))} + \log(1/(\epsilon \cdot 2^{-n}))\right) \\
&= O\left(\sqrt{2^n(n + \log(1/\epsilon))} + \log(1/\epsilon)\right)
\end{aligned}$$

T gates and ancillae. So naively, let $\tilde{U} = \tilde{U}_1 \tilde{U}_2 \cdots \tilde{U}_{2^n-1} \in \text{U}(2^{n+1})$. Then by Lemma II.1, $\tilde{U}$ implements U to error ϵ and $\tilde{U}$ admits an exact Clifford + T implementation with a T-count of

$$O\left(2^n \left(\sqrt{2^n(n + \log(1/\epsilon))} + \log(1/\epsilon)\right)\right),$$

and an ancilla-count of

$$O\left(\sqrt{2^n(n + \log(1/\epsilon))} + \log(1/\epsilon)\right).$$

Thus, our naïve recursive CS-based synthesis achieves a T-count of $O(2^{3n/2} \cdot n^{1/2})$, improving slightly over the previous best $O(2^{3n/2} \cdot n)$ scaling due to [1] while relying on an arguably simpler analysis.

IV. LOWER THE T-COUNT

The key observation that lowers the T-count is that many of the controlled unitaries in the recursive CS decomposition can be grouped together.

Throughout this section, we will write $t_n(i)$ simply as $t(i)$, since the subscript is always n . Let $k \in [n - 1]$ be a parameter that we will optimize later. To begin with, note that each of $U_1, \dots, U_{2^k-1}$ acts on one of the last k qubits, since $t(1), \dots, t(2^k - 1) \in \{n - k + 1, \dots, n\}$. Their product $U_1 \cdots U_{2^k-1}$ is thus a unitary acting on qubits in $[n] \setminus [n - k]$ and controlled by qubits in $[n - k]$, i.e.

$$W_0 := U_1 \cdots U_{2^k-1} = \sum_{x \in \{0,1\}^{n-k}} |x\rangle\langle x| \otimes V_{0,x},$$

where $V_{0,x} \in \text{U}(2^k)$. Similarly, since $t(2^k + i) = t(i)$ for any $i \in [2^k - 1]$, each of $U_{2^k+1}, \dots, U_{2^{k+2}-1}$ also acts on one of the last k qubits. So their product $W_1 := U_{2^k+1} \cdots U_{2^{k+2}-1}$ is also a multi-controlled k -qubit unitary. Let us fully generalize this. For each $j \in \{0, 1, \dots, 2^{n-k} - 1\}$, let

$$\begin{aligned}
W_j &:= U_{j \cdot 2^k+1} \cdot U_{j \cdot 2^k+2} \cdots U_{j \cdot 2^k+2^k-1} \\
&= \sum_{x \in \{0,1\}^{n-k}} |x\rangle\langle x| \otimes V_{j,x},
\end{aligned} \tag{4}$$

where $V_{j,x} \in \text{U}(2^k)$. Overall,

$$U = W_0 \cdot \prod_{j=1}^{2^{n-k}-1} U_{j \cdot 2^k} \cdot W_j. \tag{5}$$

If we implement each W_j , a product of $2^k - 1$ controlled unitaries, to error $\epsilon \cdot 2^{-(n-k)}$ naively as in Sec. III A, the T-count is $O(2^k(\sqrt{2^n(n + \log(1/\epsilon))} + \log(1/\epsilon)))$. If we ignore the polynomial dependence on n and take ϵ to be constant, this T-count is on the order of $2^{k+n/2}$. However, we will show in Lemma IV.3 that there is a cheaper way to implement W_j , with cost roughly on the order of $2^{(n+k)/2} + 2^{2k}$.

a. Notation Let U_i be a multi-controlled single-qubit unitary targeting qubit $t(i)$, i.e.

$$U_i = \sum_{x \in \{0,1\}^{n-1}} |x_1, \dots, x_{t(i)-1}\rangle \langle x_1, \dots, x_{t(i)-1}| \\ \otimes V_{i,x} \otimes |x_{t(i)}, \dots, x_{n-1}\rangle \langle x_{t(i)}, \dots, x_{n-1}|,$$

where $V_{i,x} \in \text{U}(2)$. We adopt the following notation to simplify the writing of U_i as

$$U_i = \sum_{x \in \{0,1\}^{n-1}} |x\rangle \langle x| \otimes [V_{i,x}]_{t(i)},$$

where the subscript $t(i)$ in $[A]_{t(i)}$ indicates that A actually acts on the qubit indexed by $t(i)$ and the control qubits are indexed by $[n] \setminus \{t(i)\}$.

A. Implementing multi-controlled k -qubit unitaries

In Lemma IV.1, we first show a special case of implementing a product of m multi-controlled single-qubit unitaries, each acting on one of k designated target qubits. Then we plug in $m = O(2^k)$ and bound the cost of implementing a general multi-controlled k -qubit unitary in Lemma IV.3.

Lemma IV.1. (T-count for a product of multi-controlled single-qubit unitaries). Let n, m, k be positive integers and $k < n$. For each $i \in [m]$, let

$$U_i = \sum_{x \in \{0,1\}^{n-1}} |x\rangle \langle x| \otimes [R_{i,x}]_{h_i},$$

where $R_{i,x} \in \text{SU}(2)$ and $h_i \in [k]$. Let $U = U_1 U_2 \cdots U_m$. Then for any $\epsilon > 0$, there exists $\tilde{U} \in \text{U}(2^n)$ such that $\|U - \tilde{U}\| \leq \epsilon$ and $\tilde{U}$ admits an exact Clifford + T implementation using $O(2^{n/2} \cdot \sqrt{m \cdot \log(m/\epsilon)} + 2^k \cdot m \cdot \log(m/\epsilon))$ T gates and ancillae.

Proof. We first describe the construction of $\tilde{U} \in \text{U}(2^n)$. For each $i \in [m]$ and $x \in \{0,1\}^{n-1}$, since $R_{i,x} \in \text{SU}(2)$, it follows from Lemma II.4 that there exists $\tilde{R}_{i,x} \in \text{SU}(2)$ such that $\|R_{i,x} - \tilde{R}_{i,x}\| \leq \epsilon/m$ and $\tilde{R}_{i,x}$ can be written exactly as a product of L Hadamard and L T gates where $L = \lceil c \cdot \log(m/\epsilon) \rceil$ for some constant $c > 0$. More concretely, let us write

$$\tilde{R}_{i,x} = H^{f_{i,1}(x)} \cdot T^{f_{i,2}(x)} \cdots H^{f_{i,2L-1}(x)} \cdot T^{f_{i,2L}(x)},$$

where $f_{i,j}: \{0,1\}^{n-1} \mapsto \{0,1\}$ for each $j \in [2L]$. Then $\tilde{U}_i := \sum_{x \in \{0,1\}^{n-1}} |x\rangle \langle x| \otimes [\tilde{R}_{i,x}]_{h_i}$ implements U_i to error ϵ/m . Overall, by Lemma II.1, $\tilde{U} := \tilde{U}_1 \tilde{U}_2 \cdots \tilde{U}_m$ implements U to error ϵ .

It remains to calculate the T-count and ancilla-count for implementing $\tilde{U}$.

Remark IV.2. Let us take a detour and recall the high-level idea for implementing each $\tilde{U}_i$. The Boolean function $f_{i,j}$ can be written as a polynomial in $n-1$ variables $x_1, \dots, x_{h_i-1}, x_{h_i+1}, \dots, x_n$ over $\mathbb{F}_2$. Let us write $x = (x_1, \dots, x_{h_i-1}, x_{h_i+1}, \dots, x_n) \in \{0,1\}^{n-1}$. Suppose that we can implement

$$|x\rangle \otimes \left(\bigotimes_{j \in [2L]} |y_j\rangle \right) \mapsto |x\rangle \otimes \left(\bigotimes_{j \in [2L]} |y_j \oplus f_{i,j}(x)\rangle \right). \quad (6)$$

Then we can implement $\tilde{U}_i$ as follows:

- (1) prepare $2L$ ancillae in $|0^{2L}\rangle$.
- (2) run Eq. (6) once on all qubits except the one indexed by h_i (whose state is $|x_{h_i}\rangle$).
- (3) sequentially apply L controlled-Hadamard and L controlled-T gates, each of which is controlled on the corresponding qubit with state $|f_{i,j}(x)\rangle$ and acts on the same qubit indexed by h_i .
- (4) run Eq. (6) once to uncompute each ancilla register with state $|f_{i,j}(x)\rangle$ back to $|0\rangle$.

In summary,

$$\begin{aligned} & |x\rangle \otimes |x_{h_i}\rangle \otimes |0^{2L}\rangle \\ & \mapsto |x\rangle \otimes |x_{h_i}\rangle \otimes \left(\bigotimes_{j \in [2L]} |f_{i,j}(x)\rangle \right) \quad \text{step 2} \\ & \mapsto |x\rangle \otimes (\tilde{R}_{i,x} \cdot |x_{h_i}\rangle) \otimes \left(\bigotimes_{j \in [2L]} |f_{i,j}(x)\rangle \right) \quad \text{step 3} \\ & \mapsto |x\rangle \otimes (\tilde{R}_{i,x} \cdot |x_{h_i}\rangle) \otimes |0^{2L}\rangle \quad \text{step 4.} \end{aligned}$$

So to implement $\tilde{U}$, it boils down to the Clifford + T implementation of Eq. (6) for each $i \in [m]$ and how their product can be optimized altogether.

We now continue the proof of Lemma IV.1. From now on we write $x = (x_1, \dots, x_n)$, and each $f_{i,j}$ is extended to a polynomial in n variables (i.e. its dependence on x_{h_i} is trivial).

Since all the targeting qubits h_i can only be in the first k qubits, we can decompose $f_{i,j}$ in a way that separates the variables $x_1, \dots, x_k$ from the rest: for each $i \in [m]$ and $j \in [2L]$,

$$f_{i,j}(x) = \sum_{b \in \{0,1\}^k} (x_1^{b_1} \cdots x_k^{b_k}) \cdot g_{i,j,b}(x_{k+1}, \dots, x_n), \quad (7)$$

where $g_{i,j,b}$ is a polynomial in $x_{k+1}, \dots, x_n$ over $\mathbb{F}_2$. We denote all the $g_{i,j,b}$ polynomials together by

$$g: \{0,1\}^{n-k} \rightarrow \{0,1\}^{m \cdot (2L) \cdot 2^k}, \quad (8)$$

where $g(x_{k+1}, \dots, x_n)_{i,j,b} = g_{i,j,b}(x_{k+1}, \dots, x_n)$.

Now we are ready to describe the final algorithm:

- (1) Prepare M ancillae in $|0^M\rangle$. Denote the input state as $|x_1, \dots, x_n\rangle \otimes |0^M\rangle$. We will specify M later.
- (2) Recall $g: \{0, 1\}^{n-k} \rightarrow \{0, 1\}^{m \cdot (2L) \cdot 2^k}$ defined in Eq. (8). Using [16] with $r = m \cdot (2L) \cdot 2^k$, the unitary U_g given by

$$|x_{k+1}, \dots, x_n\rangle \otimes \left(\bigotimes_{\substack{i \in [m], j \in [2L], \\ b \in \{0,1\}^k}} |y_{i,j,b}\rangle \right) \mapsto |x_{k+1}, \dots, x_n\rangle \otimes \left(\bigotimes_{\substack{i \in [m], j \in [2L], \\ b \in \{0,1\}^k}} |y_{i,j,b} \oplus g_{i,j,b}(x_{k+1}, \dots, x_n)\rangle \right)$$

admits an exact Clifford + T implementation using $O(\sqrt{2^{n-k} \cdot r}) = O(2^{n/2} \sqrt{mL})$ T gates and ancillae. Applying U_g gives

$$|x_1, \dots, x_n\rangle \otimes \underbrace{\left(\bigotimes_{\substack{i \in [m], j \in [2L], \\ b \in \{0,1\}^k}} |g_{i,j,b}(x_{k+1}, \dots, x_n)\rangle \right)}_{:=|A\rangle} \otimes |0^{M_A}\rangle.$$

Here, the ancilla count must satisfy $M = \Omega(mL2^k + M_A)$ and $M_A = \Omega(2^{n/2} \sqrt{mL})$.

- (3) For each $i \in [m]$, we now describe how to implement $\tilde{U}_i := \sum_{x \in \{0,1\}^{n-1}} |x\rangle\langle x| \otimes [\tilde{R}_{i,x}]_{h_i}$:
 - (a) Denote the input state by $|x_1, \dots, x_n\rangle \otimes |A\rangle \otimes |0^{M_A}\rangle$.
 - (b) Apply the monomial-generating unitary in Lemma II.2 to produce the state that encodes all monomials in $x_1, \dots, x_{h_i-1}, x_{h_i+1}, \dots, x_k$. With a permutation of registers, the resulting state is given by

$$|x_{h_i}\rangle \otimes |A\rangle \otimes \underbrace{\left(\bigotimes_{a \in \{0,1\}^{k-1}} |x_1^{a_1} \dots x_{h_i-1}^{a_{h_i-1}} x_{h_i+1}^{a_{h_i}} \dots x_k^{a_{k-1}}\rangle \right)}_{:=|B\rangle} \otimes |0^{M_B}\rangle,$$

using $O(2^k)$ Toffoli gates and ancillae. Here, the ancilla counts must satisfy $M_A = \Omega(2^k + M_B)$.

- (c) Set $M_B = 2L \cdot 2^k$. By Eq. (7), one can use $2L \cdot 2^k$ Toffoli gates to produce

$$|x_{h_i}\rangle \otimes |A\rangle \otimes |B\rangle \otimes \left(\bigotimes_{j \in [2L]} |f_{i,j}(x)\rangle \right).$$

- (d) Use L controlled-Hadamard and L controlled-T gates to produce

$$\left(\tilde{R}_{i,x} \cdot |x_{h_i}\rangle \right) \otimes |A\rangle \otimes |B\rangle \otimes \left(\bigotimes_{j \in [2L]} |f_{i,j}(x)\rangle \right).$$

- (e) Uncompute the state to

$$|x_1, \dots, x_{h_i-1}\rangle \otimes \left(\tilde{R}_{i,x} \cdot |x_{h_i}\rangle \right) \otimes |x_{h_i+1}, \dots, x_k\rangle \otimes |A\rangle \otimes |0^{M_A}\rangle.$$

- (4) Apply U_g to uncompute $|A\rangle$.

Overall, both the number of ancillae and T-count are $O(2^{n/2} \cdot \sqrt{mL} + 2^k mL)$. ■

We now bound the T-count for implementing a general multi-controlled k -qubit unitary.

Lemma IV.3. (T-count for multi-controlled k -qubit unitaries). Let k be a positive integer smaller than n . For any $\epsilon > 0$ and any multi-controlled k -qubit unitary $W \in U(2^n)$ targeting the last k qubits, i.e.

$$W = \sum_{x \in \{0,1\}^{n-k}} |x\rangle\langle x| \otimes V_x,$$

where $V_x \in U(2^k)$, there exists $\tilde{W} \in U(2^{n+1})$ such that $\tilde{W}$ implements W to error ϵ and $\tilde{W}$ admits an exact Clifford + T implementation using $O(2^{(n+k)/2} \sqrt{k + \log(1/\epsilon)} + 4^k(k + \log(1/\epsilon)))$ T gates and ancillae.

Proof. It follows from the recursive CS decomposition in Theorem III.5 [see also Eq. (4)] that W can be written as

$$W = U_1 U_2 \cdots U_{2^k-1},$$

where for each $i \in [2^k - 1]$, U_i is a multi-controlled single-qubit unitary targeting qubit $t(i) \in [n] \setminus [n - k]$, i.e.

$$U_i = \sum_{x \in \{0,1\}^{n-1}} |x\rangle\langle x| \otimes [e^{i\theta_{i,x}} \cdot R_{i,x}]_{t(i)},$$

where $\theta_{i,x} \in [0, 2\pi)$, $R_{i,x} \in \text{SU}(2)$. Here we single out the phases $e^{i\theta_{i,x}}$ so that each $R_{i,x}$ admits a Hadamard+T approximation following from Lemma II.4. Let us write

$$R_i := \sum_{x \in \{0,1\}^{n-1}} |x\rangle\langle x| \otimes [R_{i,x}]_{t(i)}$$

and

$$\Phi_i := \sum_{x \in \{0,1\}^{n-1}} |x\rangle\langle x| \otimes \left[\text{diag}(e^{i\theta_{i,x}}, e^{-i\theta_{i,x}}) \right]_{t(i)}.$$

So $U_i = R_i \cdot \Phi_i$. To relate Φ_i to a unitary operator with determinant 1, we use a similar trick as in the proof of [2], Theorem 1.2] and consider

$$\begin{aligned} D_i &:= \Phi_i \otimes |0\rangle\langle 0|_{\text{anc}} + \Phi_i^\dagger \otimes |1\rangle\langle 1|_{\text{anc}} \\ &= \sum_{x \in \{0,1\}^{n-1}} |x\rangle\langle x| \otimes [I_2]_{t(i)} \otimes \underbrace{\text{diag}(e^{i\theta_{i,x}}, e^{-i\theta_{i,x}})}_{=: D_{i,x} \in \text{SU}(2)}_{\text{anc}}, \end{aligned}$$

where the last equality is because

$$\begin{aligned} &\text{diag}(e^{i\theta_{i,x}}, e^{i\theta_{i,x}}) \otimes |0\rangle\langle 0| + \text{diag}(e^{-i\theta_{i,x}}, e^{-i\theta_{i,x}}) |1\rangle\langle 1| \\ &= \text{diag}(e^{i\theta_{i,x}}, e^{-i\theta_{i,x}}, e^{i\theta_{i,x}}, e^{-i\theta_{i,x}}) \\ &= I_2 \otimes \text{diag}(e^{i\theta_{i,x}}, e^{-i\theta_{i,x}}). \end{aligned}$$

This additional ancilla register **anc** does not affect the overall implementation of U_i because for any $|\psi\rangle \in \mathbb{C}^{2^n}$,

$$\begin{aligned} (U_i |\psi\rangle) \otimes |0\rangle &= (R_i \cdot \Phi_i \cdot |\psi\rangle) \otimes |0\rangle \\ &= (R_i \otimes I_2) \cdot D_i \cdot (|\psi\rangle \otimes |0\rangle). \end{aligned}$$

In other words, $(R_i \otimes I_2) D_i$ implements U_i to error 0.

Now, $R_i \otimes I_2$ and $D_i \in \text{SU}(2^{n+1})$ are n -fold controlled single-qubit unitaries, targeting qubit $t(i) \in [n] \setminus [n - k]$ and qubit $n + 1$ (i.e. register **anc**) respectively. Hence, we can further simplify to write them as

$$R_i \otimes I_2 = \sum_{x \in \{0,1\}^n} |x\rangle\langle x| \otimes [R_{i,x}]_{t(i)}$$

and

$$D_i = \sum_{x \in \{0,1\}^n} |x\rangle\langle x| \otimes D_{i,x},$$

where $R_{i,x}, D_{i,x} \in \text{SU}(2)$. Overall,

$$\begin{aligned} (W|\psi\rangle) \otimes |0\rangle &= (U_1 U_2 \cdots U_{2^k-1} |\psi\rangle) \otimes |0\rangle \\ &= \left(\prod_{i=1}^{2^k-1} \left(\sum_{x \in \{0,1\}^n} |x\rangle\langle x| \otimes [R_{i,x}]_{t(i)} \right) \cdot \left(\sum_{x \in \{0,1\}^n} |x\rangle\langle x| \otimes D_{i,x} \right) \right) \cdot (|\psi\rangle \otimes |0\rangle). \end{aligned}$$

Applying Lemma IV.1 with $m = 2 \cdot (2^k - 1)$, we know that there exists $\tilde{W} \in \text{U}(2^{n+1})$ such that $\tilde{W}$ implements W to error ϵ and $\tilde{W}$ admits an exact Clifford+T implementation using $O(2^{(n+k)/2} \sqrt{k + \log(1/\epsilon)} + 4^k(k + \log(1/\epsilon)))$ T gates and ancillae. ■

B. Proof of Theorem I.1

Theorem I.1. (Main result). Let $\epsilon > 0$ and set $L = n + \log(1/\epsilon)$. Then any $U \in \text{U}(2^n)$ can be ϵ -approximated by a Clifford+T circuit using

$$O(2^{4n/3} \cdot L^{2/3} + 2^n \cdot L) \quad \text{T gates}$$

and

$$O(2^{2n/3} \cdot L^{1/3} + L) \quad \text{ancillae.}$$

In particular, for any positive integer $k \leq (n - \log_2 L)/3$, U can be ϵ -approximated by a Clifford+T circuit using

$$O(2^{(3n-k)/2} \cdot \sqrt{L}) \quad \text{T gates}$$

and

$$O(2^{(n+k)/2} \cdot \sqrt{L}) \quad \text{ancillae.}$$

Proof. It follows from the recursive CS decomposition in Theorem III.5 [see also Eq. (5)] that

$$U = W_0 \cdot \prod_{j=1}^{2^{n-k}-1} U_{j \cdot 2^k} \cdot W_j,$$

where each W_j is a multi-controlled k -qubit unitary acting on the last k qubits. We have already calculated the cost for implementing each $U_{j \cdot 2^k}$ and W_j :

- (i) By Corollary III.6, for each $j \in \{1, \dots, 2^{n-k} - 1\}$, there exists $\tilde{U}_{j \cdot 2^k} \in U(2^{n+1})$ such that $\tilde{U}_{j \cdot 2^k}$ implements $U_{j \cdot 2^k}$ to error δ and $\tilde{U}_{j \cdot 2^k}$ admits

an exact Clifford + T implementation using $O(\sqrt{2^n \cdot \log(1/\delta)} + \log(1/\delta))$ T gates and ancillae.

- (ii) By Lemma IV.3, for each $j \in \{0, 1, \dots, 2^{n-k} - 1\}$, there exists $\tilde{W}_j$ such that $\tilde{W}_j$ implements W_j to error δ and $\tilde{W}_j$ admits an exact Clifford + T implementation using $O(2^{(n+k)/2} \sqrt{k + \log(1/\delta)} + 4^k(k + \log(1/\delta)))$ T gates and ancillae.

Let $\delta = \epsilon \cdot 2^{-(n-k)}/2$ and $L = k + \log(1/\delta) = n + \log(1/\epsilon)$. Then the total ancilla-count is

$$O\left(\sqrt{2^n \cdot \log(1/\delta)} + \log(1/\delta) + 2^{\frac{n+k}{2}} \sqrt{k + \log(1/\delta)} + 4^k(k + \log(1/\delta))\right) = O\left(2^{\frac{n+k}{2}} \sqrt{L} + 4^k L\right), \quad (9)$$

and hence the total T-count is

$$O(2^{n-k} \cdot (2^{\frac{n+k}{2}} \sqrt{L} + 4^k L)). \quad (10)$$

For any positive integer k satisfying $2^{\frac{n+k}{2}} \sqrt{L} \geq 4^k L$, i.e. $2^k \leq 2^{n/3} \cdot L^{-1/3}$ or $k \leq (n - \log_2 L)/3$, we have that

$$(9) = O(2^{\frac{n+k}{2}} \sqrt{L}) \quad \text{and} \quad (10) = O(2^{\frac{3n-k}{2}} \sqrt{L}). \quad (11)$$

The ancillae and T gates tradeoff follows from the fact that $2^{\frac{n+k}{2}} \cdot 2^{\frac{3n-k}{2}} = 2^{2n}$. This proves the second part of Theorem I.1. To prove the first part, we distinguish two cases and choose k to minimize the T-count in each.

- (i) When $L \geq 2^n$, we have that $4^k L \geq 2^{\frac{n+k}{2}} \sqrt{L}$ for any k . Then

$$(9) = O(4^k \cdot L) \quad \text{and} \quad (10) = O(2^{n-k} \cdot 4^k \cdot L) = O(2^{n+k} \cdot L).$$

So we should set k to be the smallest possible value, i.e. $k = 1$.

- (ii) When $L \leq 2^n$, let us set $k = \lfloor (n - \log_2 L)/3 \rfloor$. Then by Eq. (11), we have

$$(9) = O(2^{2n/3} \cdot L^{1/3}) \quad \text{and} \quad (10) = O(2^{4n/3} \cdot L^{2/3}).$$

Combining the above two cases gives

$$\text{ancilla-count: } O(L + 2^{2n/3} \cdot L^{1/3}),$$

and

$$\text{T-count: } O(2^n \cdot L + 2^{4n/3} \cdot L^{2/3}).$$

ACKNOWLEDGMENTS

The author would like to thank Bill Huggins and Robin Kothari for many insightful discussions throughout the course of this project, Kewen Wu for comments on an early-stage note that developed into this paper, and Aram Harrow, Gregory Rosenthal, Nathan Wiebe, and John Wright for helpful discussions. This work was done when the author was a Student Researcher at Google.

DATA AVAILABILITY

There are no publicly available research data or software supporting this manuscript. Requests for further information or data should be sent to the authors.

APPENDIX: PROOF OF LEMMA II.1

Lemma A.1. (Composition error bound). Suppose that $V_i \in U(2^{n+m_i})$ implements $U_i \in U(2^n)$ to error ϵ for some integer $m_i \geq 0$. Let $m = \max_{i \in [L]} m_i$. Then $(V_1 \otimes I_{2^{m-m_1}}) \cdots (V_L \otimes I_{2^{m-m_L}}) \in U(2^{n+m})$ implements $U_1 \cdots U_L$ to error $L\epsilon$.

Proof. We will prove the lemma by induction on L .

The base case of $L = 1$ is trivial. For the induction step, assume the claim holds for a product of $L - 1$ terms for some $L \geq 2$, that is $W = (V_1 \otimes I_{2^{m-m_1}}) \cdots (V_{L-1} \otimes I_{2^{m-m_{L-1}}}) \in U(2^{n+m})$ implements $U = U_1 \cdots U_{L-1}$ to error $(L - 1)\epsilon$, i.e.

$$\|W \cdot (I_{2^n} \otimes |0^m\rangle) - U_1 \cdots U_{L-1} \otimes |0^m\rangle\| \leq (L - 1)\epsilon. \quad (A1)$$

Without loss of generality, we can set $m = \max_{i \in [L]} m_i$. The goal is to show that when we add one more term of $V_L \otimes I_{2^{m-m_L}}$ to W , for any n -qubit state $|\psi\rangle$, we have that

$$\|W \cdot (V_L \otimes I_{2^{m-m_L}}) \cdot (|\psi\rangle \otimes |0^m\rangle) - U_1 \cdots U_L |\psi\rangle \otimes |0^m\rangle\| \leq L\epsilon. \quad (\text{A2})$$

Let us adopt a telescoping sum argument to the left-hand side of Eq. (A2).

$$\begin{aligned} & \|W(V_L \otimes I_{2^{m-m_L}}) \cdot (|\psi\rangle \otimes |0^m\rangle) - U_1 \cdots U_L |\psi\rangle \otimes |0^m\rangle\| \\ &= \|W(V_L \otimes I_{2^{m-m_L}}) \cdot (|\psi\rangle \otimes |0^m\rangle) - W(U_L |\psi\rangle \otimes |0^m\rangle) + W(U_L |\psi\rangle \otimes |0^m\rangle) - U_1 \cdots U_L |\psi\rangle \otimes |0^m\rangle\| \\ &\leq \underbrace{\|W(V_L \otimes I_{2^{m-m_L}}) \cdot (|\psi\rangle \otimes |0^m\rangle) - W(U_L |\psi\rangle \otimes |0^m\rangle)\|}_{:=S_1} + \underbrace{\|W(U_L |\psi\rangle \otimes |0^m\rangle) - U_1 \cdots U_L |\psi\rangle \otimes |0^m\rangle\|}_{:=S_2}. \end{aligned}$$

(triangle inequality)

We now bound S_1 and S_2 separately.

$$\begin{aligned} S_1 &\leq \|W\| \cdot \|(V_L \otimes I_{2^{m-m_L}}) \cdot (|\psi\rangle \otimes |0^m\rangle) - U_L |\psi\rangle \otimes |0^m\rangle\| && (\|AB\| \leq \|A\| \cdot \|B\|) \\ &\leq \|(V_L \otimes I_{2^{m-m_L}}) \cdot (|\psi\rangle \otimes |0^m\rangle) - U_L |\psi\rangle \otimes |0^m\rangle\| && (\|U\| = 1 \text{ for any unitary } U) \\ &= \|V_L \cdot (|\psi\rangle \otimes |0^{m_L}) - U_L |\psi\rangle \otimes |0^{m_L}\| \\ &\leq \epsilon, && (V_L \text{ implements } U_L \text{ to error } \epsilon \text{ and Definition I.2}) \end{aligned}$$

and

$$\begin{aligned} S_2 &= \|W(|\psi'\rangle \otimes |0^m\rangle) - U_1 \cdots U_{L-1} |\psi'\rangle \otimes |0^m\rangle\| && (\text{set } |\psi'\rangle := U_L |\psi\rangle) \\ &\leq (L-1)\epsilon. && (\text{induction hypothesis in Equation (A1)}) \end{aligned}$$

Hence $S_1 + S_2 \leq L\epsilon$ which proves Eq. (A2) and thus completes the induction. ■

-
- [1] G. H. Low, V. Kliuchnikov, and L. Schaeffer, *Trading T gates for dirty qubits in state preparation and unitary synthesis*, [Quantum](#) **8**, 1375 (2024).
 - [2] D. Gosset, R. Kothari, and K. Wu, *Quantum state preparation with optimal T-count*, [Quantum](#) **10**, 2168 (2026).
 - [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] M. Möttönen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, *Quantum circuits for general multiqubit gates*, [Phys. Rev. Lett.](#) **93**, 130502 (2004).
 - [5] C. M. Dawson and M. A. Nielsen, *The Solovay-Kitaev algorithm*, [Quantum Inf. Comput.](#) **6**, 81 (2006).
 - [6] V. Kliuchnikov, D. Maslov, and M. Mosca, *Fast and efficient exact synthesis of single-qubit unitaries generated by clifford and T gates*, [Quantum Inf. Comput.](#) **13**, 607 (2013).
 - [7] P. Selinger, *Efficient Clifford+T approximation of single-qubit operators*, [Quantum Inf. Comput.](#) **15**, 159 (2015).
 - [8] N. J. Ross and P. Selinger, *Optimal ancilla-free Clifford+T approximation of z-rotations*, [Quantum Inf. Comput.](#) **16**, 901 (2016).
 - [9] In the introduction, we treat ϵ as a constant in order to focus on the scaling in n . Our main result, Theorem I.1, is stated in full generality with explicit ϵ -dependence.
 - [10] G. Rosenthal, *Query and depth upper bounds for quantum unitaries via Grover search*, [arXiv:2111.07992](#).
 - [11] D. W. Berry, N. C. Rubin, A. O. Elnabawy, G. Ahlers, A. E. DePrince III, J. Lee, C. Gogolin, and R. Babbush, *Quantum simulation of realistic materials in first quantization using non-local pseudopotentials*, [npj Quantum Inf.](#) **10**, 130 (2024).
 - [12] Y. Su, D. W. Berry, N. Wiebe, N. Rubin, and R. Babbush, *Fault-tolerant quantum simulations of chemistry in first quantization*, [PRX Quantum](#) **2**, 040332 (2021).
 - [13] W. J. Huggins, O. Leimkuhler, T. F. Stetina, and K. B. Whaley, *Efficient state preparation for the quantum simulation of molecules in first quantization*, [PRX Quantum](#) **6**, 020319 (2025).
 - [14] S. Fomichev, K. Hejazi, M. S. Zini, M. Kiser, J. Fraxanet, P. A. M. Casares, A. Delgado, J. Huh, A.-C. Voigt, J. E. Mueller, and J. M. Arrazola, *Initial state preparation for quantum chemistry on quantum computers*, [PRX Quantum](#) **5**, 040339 (2024).
 - [15] D. W. Berry, Y. Tong, T. Khattar, A. White, T. I. Kim, G. H. Low, S. Boixo, Z. Ding, L. Lin, S. Lee, G. K.-L. Chan, R. Babbush, and N. C. Rubin, *Rapid initial-state preparation*

- for the quantum simulation of strongly correlated molecules*, [PRX Quantum 6, 020327 \(2025\)](#).
- [16] We note that there is a typo in the statement of [\[\[1\]](#), Theorem 2]: The T-count in both cases should be $\sqrt{r \cdot 2^n}$. The tradeoff established there is only between T-depth and ancilla-count and does not affect the T-count.
- [17] C.C. Paige and M. Wei, *History and generality of the CS decomposition*, [Linear Algebra Appl. 208–209, 303 \(1994\)](#).
- [18] E. Tang and K. Tian, *A CS guide to the quantum singular value transformation*, [arXiv:2302.14324](#).
- [19] If the bits are indexed reversely (least significant bit indexed by 1), then this function is known as the ruler function (OEIS A001511) [\[17\]](#).
- [20] OEIS Foundation Inc., *OEIS sequence a001511: The ruler function (2026)*. *the On-Line Encyclopedia of Integer Sequences*, accessed 2026-02-21.