Fast exact synthesis of two-qubit unitaries using a near-minimum number of T gates

Longcheng Li [1,2,*] Cheng Guo [1,†] Qian Li [1,‡] and Xiaoming Sun [1,3,§]

[1] *Institute of Computing Technology, Chinese Academy of Sciences, 100190 Beijing, China*

[2] *School of Computer Science and Technology, University of Chinese Academy of Sciences, 100049 Beijing, China*

[3] *CAS Center for Excellence in Topological Quantum Computation, University of Chinese Academy of Sciences, 100049 Beijing, China*

(Received 18 January 2023; accepted 16 March 2023; published 18 April 2023)

This paper focuses on exact synthesis of two-qubit unitaries using Clifford and T gates. We propose an ancilla-free synthesis algorithm (i) which uses T gates no more than ten times the minimum possible number of T gates, also known as the T count, and (ii) whose time complexity is linear with the T count and thus instance optimal. Our synthesis algorithm relies on a characterization of the T count of two-qubit unitaries based on Lie group homomorphism, which may be interest of its own. Precisely, we show that for any two-qubit unitary generated by Clifford and T gates, its T count is equivalent to the least denominator exponent of its SO(6) representation up to a factor of at most 10.

DOI: 10.1103/PhysRevA.107.042424

I. INTRODUCTION

The quantum circuit model is a universal language for the description of quantum algorithms, in which many famous quantum algorithms, including Shor's algorithm [1] and Grover's search [2], are presented. As quantum computing has entered the era of noisy intermediate-scale quantum systems, where there are inherent limitations on the size, depth, and number of ancillas of quantum circuits that can be supported by physical experimental hardware, it becomes extremely important to optimize the size, depth, and number of ancillas of quantum circuits, and the degree of optimization directly affects the scope of application of quantum computers [3].

As present quantum computing devices are extremely vulnerable to quantum noises, we need techniques for fault-tolerant computation. A well-studied approach for fault-tolerant computation is based on the observation that many quantum error-correcting codes can natively implement Clifford gates. However, Clifford gates are not approximately universal, i.e., they cannot approximate an arbitrary unitary with arbitrary precision. To achieve the approximate universality, one and only one type of non-Clifford gate is needed [4]. The most common choice of this gate is the T gate, which is a single-qubit gate and can be fault-tolerantly implemented by using the magic state distillation technique [5]. The total resource consumption of a quantum circuit is usually dominated by T gates. To implement a T gate using magic state distillation at a physical error rate of 10^{-3} in a surface-code-based architecture, the space-time cost is estimated to be roughly (225 logical qubits) $\times$ (10 Clifford depths) [6], which is much more expensive than Clifford gates. Thus the number of T gates used is usually treated as a measure of the resource consumption of a quantum circuit [7].

There has been a great deal of research studying how to *approximately* synthesize an arbitrary unitary using as few T gates as possible [8–12]. Any n-qubit unitary can be approximated within ϵ precision using $O(4^n[\log(1/\epsilon)+n])$, T gates. This can be achieved by first decomposing the unitary into $O(4^n)$ controlled-NOT (CNOT) and single-qubit gates [13] and then approximating single-qubit gates with Clifford+T circuits [10]. We remark that the bound $O(4^n[\log(1/\epsilon)+n])$ on the number of T gates is asymptotically optimal when n is fixed, since there exist unitaries whose synthesis requires $\Omega(4^n \log(1/\epsilon))$ T gates [8]. In addition, Gheorghiu *et al.* [12] provided another synthesis algorithm where the number of T gates is m_ϵ and thus instance optimal, but the synthesis time can be as large as $O(2^{2nm_\epsilon})$. Here m_ϵ denotes the minimum number of T gates needed to approximate the unitary within a precision ϵ. We remark that both of the above two synthesis algorithms are ancilla-free.

In this paper we focus on exact synthesis rather than approximate synthesis. The benefit of exact synthesis is that it does not introduce errors arising from approximations, which is crucial since minimizing the effect of errors has direct implications on the resources needed to implement an algorithm and sometimes determines the very ability to implement a quantum algorithm and demonstrate it experimentally on available hardware of a specific size [14].

There are two questions related to exact synthesis: first, what kind of unitaries can be exactly synthesized by using Clifford and T gates, and second, how to find an exact synthesis of a given unitary (if exists) as fast as possible and use as few T gates as possible. Giles and Selinger [15] provided a complete answer to the first question: A unitary can be exactly synthesized by using Clifford and T gates if and only if its entries belong to the ring $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$. In addition, the synthesis can be ancilla-free if the determinant of the unitary takes some certain values (see Lemma 10).

*lilongcheng18@mails.ucas.ac.cn

†cheng323232@163.com

‡liqian@ict.ac.cn

§sunxiaoming@ict.ac.cn

TABLE I. Summary of exact synthesis algorithms. Here n is the number of qubits, $\mathscr{T}(U)$ is the T count, and k is the denominator exponent of the input unitary.

n	No. of T gates	Time complexity	No. of ancilla	Ref.
1	$\mathscr{T}(U)$ (instance optimal)	$O(\mathscr{T}(U))$	0	[7]
2	$\leqslant 10\mathscr{T}(U)$	$O(\mathscr{T}(U))$	0	this paper
$\geqslant 1$	$O(3^{2^n}nk)$	$O(\mathrm{poly}(2^n, 3^{2^n}nk))$	1	[15]
$\geqslant 1$	$O(4^n nk)$	$O(\mathrm{poly}(2^n, 4^n nk))$	2	[11]
$\geqslant 1$	$\mathscr{T}(U)$	$O(2^{n\mathscr{T}(U)/2})$	0	[7,16]

The second question has not been completely answered. Table I summarizes some related results. Given a unitary U, we use $\mathscr{T}(U)$ to represent the T count of U. That is, $\mathscr{T}(U)$ is the minimum number of T gates used among all ancilla-free Clifford+T circuits implementing U. For single-qubit unitaries, Kliuchnikov *et al.* [14] provided a perfect answer: They developed an ancilla-free synthesis algorithm (i) which uses $\mathscr{T}(U)$ T gates and (ii) whose time complexity is $O(\mathscr{T}(U))$ and thus instance optimal. For general n-qubit unitaries U, Giles and Selinger [15] proposed a synthesis algorithm (i) using $O(3^{2^n}nk)$ T gates, which is far from optimal, and (ii) whose time complexity is $O(\mathrm{poly}(2^n, 3^{2^n}nk))$, where k is the denominator exponent of U. We remark that k may be much larger than $\mathscr{T}(U)$. For example, $H^{\otimes n}$ has a denominator exponent $2n$, but $\mathscr{T}(H^{\otimes n}) = 0$. The number of T gates used was then improved by Kliuchnikov [11] to $O(4^n nk)$. Gosset *et al.* [7] developed an ancilla-free synthesis algorithm where (i) the number of T gates is $\mathscr{T}(U)$ and thus instance optimal, but (ii) the time complexity is as large as $O(2^{n\mathscr{T}(U)/2})$. There are also some heuristic synthesis algorithms achieving good empirical performance, but they still need synthesis time exponential in $\mathscr{T}(U)$ in the worst case [16,17].

In summary, for the second question, only the case $n = 1$ has been answered [14]. For $n > 1$, the known synthesis algorithms either use a number of T gates which is far from optimal [11,15] or need synthesis time exponential in $\mathscr{T}(U)$ [7,16].

In this paper, we answer the second question for the case $n = 2$, precisely, as follows.

Theorem 1. Algorithm 1 is an ancilla-free exact synthesis algorithm for two-qubit unitaries (i) which uses no more than $10\mathscr{T}(U)$, T gates and (ii) whose time complexity is $O(\mathscr{T}(U))$ and thus instance optimal.

Our synthesis algorithm is based on the following characterization of the T count of two-qubit unitaries, which may be interest of its own.

Theorem 2. For all $U \in \mathscr{J}_2$, $\tau(\widehat{U}) \leqslant \mathscr{T}(U) \leqslant 10\tau(\widehat{U})$.

Here $\mathscr{J}_2$ denotes the set of two-qubit unitaries that are implemented by Clifford+T circuits without ancilla qubits, $\widehat{\cdot}$ denotes the SO(6) representation, and $\tau(\cdot)$ denotes the least denominator exponent. Their definitions can be seen in Definition 7 of Appendix A and Definitions 2 and 3 of Sec. II.

As an application of our exact synthesis algorithm, we obtain an ancilla-free approximate synthesis algorithm (Algorithm 2) for two-qubit unitaries where the number of T gates used is $O(\log(1/\epsilon))$, which is asymptotically optimal. We remark that although it does not yield a better bound than existing approximate synthesis algorithms, it shows the

Algorithm 1. The exact synthesis algorithm for $\mathscr{J}_2$.

Input: A matrix $U \in \mathscr{J}_2$
Output: A synthesis $C\prod_{i=1}^{m}\mathrm{rot}(P_i)$ where $C \in \mathscr{C}_2$ and $P_i \in \pm\mathscr{P}_2$
$l \leftarrow 0$
$V_0 \leftarrow \widehat{U}$
while $\tau(V_l) > 0$ **do**
  $l \leftarrow l + 1$
  $M_{[p_1^{(l)},q_1^{(l)}]}, \ldots, M_{[p_{t_l}^{(l)},q_{t_l}^{(l)}]}, \ldots, M_{[p_{m_l}^{(l)},q_{m_l}^{(l)}]} \leftarrow$ the sequence found by Theorem 3 such that $\tau[(\prod_{i=1}^{t_l} M_{[p_i^{(l)},q_i^{(l)}]})V_{l-1}(\prod_{i=t_l+1}^{m_l} M_{[p_i^{(l)},q_i^{(l)}]})] < \tau(V_{l-1})$
  $V_l \leftarrow (\prod_{i=1}^{t_l} M_{[p_i^{(l)},q_i^{(l)}]})V_{l-1}(\prod_{i=t_l+1}^{m_l} M_{[p_i^{(l)},q_i^{(l)}]})$
end while
$k \leftarrow 0$
for i **from** 1 **to** l **do**
  for j **from** 1 **to** t_i **do**
    $k \leftarrow k + 1$
    $M_{[r_k,l_k]} \leftarrow V_l^T M_{[p_j^{(i)},q_j^{(i)}]} V_l$
  end for
end for
for i **from** l **down to** 1 **do**
  for j **from** $t_i + 1$ **to** m_i **do**
    $k \leftarrow k + 1$
    $M_{[r_k,l_k]} \leftarrow M_{[p_j^{(i)},q_j^{(i)}]}$
  end for
end for
Compute the preimage C' of V_l and preimages $\{\mathrm{rot}(P_k)\}$ $(P_i \pm \mathscr{P}_2)$ of $\{M_{[r_k,l_k]}\}$
Let $\ell \in [8]$ be such that $U = e^{i(\pi\ell/4)}C'\prod_{i=1}^{m}\mathrm{rot}(P_i)$
return $(HSHSHS)_{(1)}^{\ell}C'\prod_{i=1}^{m}\mathrm{rot}(P_i)$ as the exact synthesis of U

Algorithm 2. Approximate synthesis algorithm.

Input: A matrix $U \in \mathrm{SU}(4)$ and $\epsilon > 0$
Output: A Clifford+T circuit
Compute the decomposition $U = [\prod_{i=1}^{m} C_i G_{Z\otimes\mathbb{I}}(\theta_i)C_i^{\dagger}]C_0$
for i **from** 1 **to** m **do**
  Compute $\alpha_i, \beta_i \in \mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ such that (i) $|\alpha_i|^2 + |\beta_i|^2 = 1$, (ii) $\|T(\alpha_i, \beta_i) - G_{Z\otimes\mathbb{I}}(\theta_i)\|_F \leqslant \epsilon/15$, and (iii) $\tau[\widehat{T(\alpha_i, \beta_i)}] = O(\log 1/\epsilon)$
end for
$U^* \leftarrow [\prod_{i=1}^{m} C_i T(\alpha_i, \beta_i)C_i^{\dagger}]C_0$
Execute Algorithm 1 on U^* and get a Clifford+T circuit G
return the above Clifford+T circuit G

possibility of leveraging our techniques to the approximate version of this problem.

The paper is organized as follows. Section II is about preliminaries. Section III presents our exact synthesis algorithm and the proof of Theorem 2. Section IV presents our approximate synthesis algorithm. Section V provides a summary.

II. PRELIMINARIES

We use $[n]$ to denote $\{1, 2, \ldots, n\}$ and $\mathbb{I}_n$ to denote the $n \times n$ identity matrix ($\mathbb{I}$ by default is $\mathbb{I}_2$). Given a matrix U, let U_{ij} denote its (i, j)th entry, $U[i, *]$ its ith row, $U[*, j]$ its jth column, and $U[i : j, k : l]$ the submatrix

$$\begin{bmatrix} U_{ik} & \cdots & U_{il} \\ \vdots & \ddots & \vdots \\ U_{jk} & \cdots & U_{jl} \end{bmatrix}.$$

For a ring or field R, we use $\mathrm{U}_n(R)$, $\mathrm{SU}_n(R)$, and $\mathrm{SO}_n(R)$ to denote the unitary group, special unitary group, and special orthogonal group of order n over R, respectively. For simplicity of notation, we also write $\mathrm{U}_n(\mathbb{C})$ as $\mathrm{U}(n)$, $\mathrm{SU}_n(\mathbb{C})$ as $\mathrm{SU}(n)$, and $\mathrm{SO}_n(\mathbb{R})$ as $\mathrm{SO}(n)$. Here $\mathbb{C}$ is the field of complex numbers and $\mathbb{R}$ is the field of real numbers. In addition, we also use $\mathrm{spin}(n)$ to denote the spin group of order n. We assume familiarity with the above groups. We refer to [18] for an introduction to these groups.

For a single-qubit gate G placed in an n-qubit circuit, we use $G_{(i)}$ to denote the $2^n \times 2^n$ matrix corresponding to applying G on the ith qubit. The n is always clear from the context. For example, $T_{(1)} = T \otimes \mathbb{I}^{\otimes(n-1)}$ is the matrix corresponding to applying the T gate on the first qubit.

A. Normal form of Clifford+T circuits

We use $\mathscr{P}_n$ to denote the set of n-qubit Pauli matrices, $\mathscr{C}_n$ to denote n-qubit Clifford group, and $\mathscr{J}_n$ to denote the group formed by the unitaries of all n-qubit ancilla-free Clifford+T circuits. Their rigorous definitions and some basic facts are presented in Appendix A.

Gosset *et al.* [7] provided a normal form of Clifford+T circuits, which will be used. Note that for a unitary $C \in \mathscr{C}_n$ we have

$$\begin{aligned} C^\dagger T_{(1)} C &= C^\dagger \exp\left(\frac{\pi}{8}(\mathbb{I}^{\otimes n} - Z_{(1)})\right) C \\ &= \exp\left(C^\dagger \frac{\pi}{8}(\mathbb{I}^{\otimes n} - Z_{(1)}) C\right) \\ &= \exp\left(\frac{\pi}{8}(\mathbb{I}^{\otimes n} - P)\right), \end{aligned}$$

where $P = C^\dagger Z_{(1)} C \in \pm\mathscr{P}_n^*$. We call $\exp[\frac{\pi}{8}(\mathbb{I}^{\otimes n} - P)]$ the $\pi/4$ rotation in Pauli angle P. Formally, we use the following definition.

Definition 1 (Pauli $\pi/4$ rotation). For a $P \in \pm\mathscr{P}_n^*$, the $\pi/4$ rotation in Pauli angle P is defined as

$$\mathrm{rot}(P) = \exp\left(\frac{\pi}{8}(\mathbb{I}^{\otimes n} - P)\right).$$

Lemma 1 (a normal form of Clifford+T circuits). A Clifford+T circuit U using m, T gates can be written as

$$U = C \prod_{i=1}^{m} \mathrm{rot}(P_i),$$

where $C \in \mathscr{C}_n$ and $P_1, \ldots, P_m \in \pm\mathscr{P}_n^*$.

Conversely, any unitary $C \prod_{i=1}^m \mathrm{rot}(P_i)$ can be implemented by a Clifford+T circuit using m, T gates.

A self-contained proof of Lemma 1 is presented in Appendix A.

B. SO(6) representation

The SO(6) representation was first introduced into the field of quantum unitary synthesis in Ref. [19]. It was used to synthesize two-qubit Clifford+CS circuits with the optimal number of controlled-phase (CS) gates. This representation reveals the essential role of non-Clifford gates in two-qubit Clifford+T circuits.

Definition 2 [SO(6) representation]. Given a unitary $U \in \mathrm{U}(4)$, let $\widehat{U} \in \mathrm{SO}(6)$ denote the SO(6) representation of U. Precisely, $\widehat{U}$ is defined as the image of $U/|U|^{1/4} \in \mathrm{SU}(4)$ under the 2-to-1 homomorphism $\mathrm{SU}(4) \to \mathrm{SO}(6)$.

The homomorphism $\mathrm{SU}(4) \to \mathrm{SO}(6)$ is derived from the exceptional isomorphism of two Lie groups $\mathrm{SU}(4) \cong \mathrm{spin}(6)$. Since spin(6) is a double cover of SO(6), we can get a 2-to-1 homomorphism from SU(4) to SO(6). The explicit construction of this homomorphism can be found in Definition 2.9 of Ref. [19]. In particular, the SO(6) representations of basic gates are

$$\widehat{S \otimes \mathbb{I}} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \widehat{H \otimes \mathbb{I}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\widehat{\mathrm{CNOT}} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \widehat{T \otimes \mathbb{I}} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The following properties are immediate by the definition.

Proposition 1 (from [19]). Given $A, B \in U(4)$, we have the following.

(a) $\widehat{A} = \widehat{B}$ if and only if there exists a phase $\phi \in \mathbb{R}$ such that $A = e^{i\phi}B$. In other words, the SO(6) representation is a bijection from two-qubit unitaries to 6×6 orthogonal matrices up to a global phase.

(b) $\widehat{AB} = \widehat{A}\widehat{B}$. In other words, the SO(6) representation preserves multiplication.

Via the SO(6) representation, we can clearly distinguish the T gate from Clifford gates: The SO(6) representation of a unitary in the Clifford group is a permutation matrix up to relative phases and that of the T gate is a two-level $\pi/4$ rotation matrix. In fact, we have the following lemma.

Lemma 2 (from [19]). Given $U \in \mathscr{J}_2$, $U \in \mathscr{C}_2$ if and only if $\widehat{U} \in \{\pm 1, 0\}^{6\times 6}$.

Thus the T gate is the unique source of the $\frac{1}{\sqrt{2}}$ factor in the SO(6) representations of two-qubit ancilla-free Clifford+T unitaries. Moreover, we can see that $\widehat{H}, \widehat{S}, \widehat{\text{CNOT}}, \widehat{T} \in \text{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$. Thus $\widehat{\mathscr{J}_2} := \{\widehat{U} \mid U \in \mathscr{J}_2\} \subseteq \text{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$. In fact, it has $\widehat{\mathscr{J}_2} = \text{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$ (see the remark after the proof of Theorem 2 in the next section).

C. Properties of $\mathbb{Z}[\frac{1}{\sqrt{2}}]$

The least denominator exponent (LDE) and parity are two important characters of elements in $\mathbb{Z}[\frac{1}{\sqrt{2}}]$.

Definition 3 (least denominator exponent). For a number $x \in \mathbb{Z}[\frac{1}{\sqrt{2}}]$, the least denominator exponent of x is defined as

$$\tau(x) := \min\{k \in \mathbb{N} \mid \sqrt{2}^k x \in \mathbb{Z}[\sqrt{2}]\}.$$

For a set $S \subset \mathbb{Z}[\frac{1}{\sqrt{2}}]$, define $\tau(S) := \max\{\tau(x) \in S\}$. For a matrix $U \in \mathbb{Z}[\frac{1}{\sqrt{2}}]^{n\times n}$, define $\tau(U) := \tau(\{U_{ij} \mid 1 \leqslant i, j \leqslant n\})$.

Proposition 2. Let U_{ij} be a nonzero entry of a matrix $U \in \text{SO}_n(\mathbb{Z}[\frac{1}{\sqrt{2}}])$. We write $U_{ij} = \frac{a+b\sqrt{2}}{\sqrt{2}^{\tau(U_{ij})}}$, where a and b are integers and then a must be odd.

Proof. When $\tau(U_{ij}) > 0$, a is odd by definition of the LDE. When $\tau(U_{ij}) = 0$, $U_{ij} = a + b\sqrt{2}$. Then $\sum_{k=1}^n U_{kj}^2 = (a^2 + 2b^2 + X) + Y\sqrt{2}$ for some $X, Y \in \mathbb{Z}[\frac{1}{2}]$, $X \geqslant 0$. Since $\sum_{k=1}^n U_{kj}^2 = 1$, $(a^2 + 2b^2 + X) + Y\sqrt{2} = 1$. Thus $b = X = Y = 0$ and $a = \pm 1$ is odd. ∎

Definition 4 (parity of $\mathbb{Z}[\frac{1}{\sqrt{2}}]$). For $x \in \mathbb{Z}[\frac{1}{\sqrt{2}}]$, it can be expressed as $x = \frac{a+b\sqrt{2}}{\sqrt{2}^{\tau(x)}}$, where a and b are integers. Then (a) x is called odd if b is odd and (b) x is called even if b is even.

A frequently used operation on two elements $x, y \in \mathbb{Z}[\frac{1}{\sqrt{2}}]$ is $(x, y) \to (\frac{x+y}{\sqrt{2}}, \frac{x-y}{\sqrt{2}})$. The following lemma tells us the least denominator exponents of the output.

Lemma 3 (from [7]). For all $x, y \in \mathbb{Z}[\frac{1}{\sqrt{2}}]$, (a) if $\tau(x) \neq \tau(y)$, $\tau(\frac{x\pm y}{\sqrt{2}}) = \max\{\tau(x), \tau(y)\} + 1$, and (b) if $\tau(x) = \tau(y) = k > 0$ and (i) if x and y have different parity, $\tau(\frac{x\pm y}{\sqrt{2}}) = k$, and (ii) if x and y have the same parity, $\tau(\frac{x\pm y}{\sqrt{2}}) < k$; additionally, if $k > 1$, then either $\tau(\frac{x+y}{\sqrt{2}})$ or $\tau(\frac{x-y}{\sqrt{2}})$ is $k - 1$ and the other one is less than $k - 1$.

TABLE II. Correspondence of P and (p, q).

p	q	P	p	q	P	p	q	P	p	q	P
1	2	$+Z\otimes\mathbb{I}$	2	1	$-Z\otimes\mathbb{I}$	1	3	$-Y\otimes\mathbb{I}$	3	1	$+Y\otimes\mathbb{I}$
1	4	$-X\otimes X$	4	1	$+X\otimes X$	1	5	$-X\otimes Y$	5	1	$+X\otimes Y$
1	6	$-X\otimes Z$	6	1	$+X\otimes Z$	2	3	$+X\otimes\mathbb{I}$	3	2	$-X\otimes\mathbb{I}$
2	4	$-Y\otimes X$	4	2	$+Y\otimes X$	2	5	$-Y\otimes Y$	5	2	$+Y\otimes Y$
2	6	$-Y\otimes Z$	6	2	$+Y\otimes Z$	3	4	$-Z\otimes X$	4	3	$+Z\otimes X$
3	5	$-Z\otimes Y$	5	3	$+Z\otimes Y$	3	6	$-Z\otimes Z$	6	3	$+Z\otimes Z$
4	5	$+\mathbb{I}\otimes Z$	5	4	$-\mathbb{I}\otimes Z$	4	6	$-\mathbb{I}\otimes Y$	6	4	$+\mathbb{I}\otimes Y$
5	6	$+\mathbb{I}\otimes X$	6	5	$-\mathbb{I}\otimes X$						

III. EXACT SYNTHESIS OF TWO-QUBIT UNITARIES

By Lemma 1, a unitary $U \in \mathscr{J}_2$ can be synthesized using m, T gates is equivalent to that U can be decomposed as $U = C\prod_{i=1}^m \text{rot}(P_i)$, where $C \in \mathscr{C}_2$ and $P_i \in \pm\mathscr{P}_2$. The basic idea of our synthesis algorithm is to find such a decomposition under the SO(6) representation.

In the SO(6) representation of $\mathscr{C}_2$, given any $U \in \mathscr{C}_2$, by Lemma 2, we have $\widehat{U} \in \{0, \pm 1\}^{6\times 6}$, or equivalently $\tau(\widehat{U}) = 0$. On the other hand, for any $U \in \mathscr{J}_2$, if $\tau(\widehat{U}) = 0$, then $U \in \mathscr{C}_2$ by Lemma 2.

In the SO(6) representation of rot(P), define $M = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix}1 & -1\\ 1 & 1\end{smallmatrix}\right]$. Given two distinct integers $p, q \in [6]$, define $M_{[p,q]}$ as follows: If $p < q$, then

$$M_{[p,q]} = \begin{array}{c} \begin{matrix} \cdots & p & \cdots & q & \cdots \end{matrix} \\ \begin{bmatrix} \mathbb{I}_{p-1} & 0 & 0 & 0 & 0 \\ 0 & M_{11} & 0 & M_{12} & 0 \\ 0 & 0 & \mathbb{I}_{q-p-1} & 0 & 0 \\ 0 & M_{21} & 0 & M_{22} & 0 \\ 0 & 0 & 0 & 0 & \mathbb{I}_{6-q} \end{bmatrix} \end{array} \begin{matrix} \vdots \\ p \\ \vdots \\ q \\ \vdots \end{matrix};$$

if $p > q$, then $M_{[p,q]} = M_{[q,p]}^T$. Trivially, $M_{[p,q]} \in \text{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$ and $\tau(M_{[p,q]}) = 1$.

The point of $M_{[p,q]}$ is that they are the SO(6) representations of rot(P). Precisely, given any $P \in \pm\mathscr{P}_2^*$, there exist two distinct $p, q \in [6]$ such that $\widehat{\text{rot}(P)} = M_{[p,q]}$. The correspondence of P and (p, q) is listed in Table II. For example, $\widehat{\text{rot}(+Z\otimes\mathbb{I})} = M_{[1,2]}$. The table can be verified by direct calculations.

Note that the SO(6) representation preserves multiplication (Proposition 1). So the exact synthesis problem can be reduced to the following problem.

Problem 1. Given a $U \in \mathscr{J}_2$, find a decomposition $\widehat{U} = L\prod_{i=1}^m M_{[p_i,q_i]}$ where $\tau(L) = 0$ such that m is minimized.

Precisely, having a decomposition $\widehat{U} = L\prod_{i=1}^m M_{[p_i,q_i]}$, we can obtain an exact synthesis of U using m, T gates as follows.

(i) Compute the preimage $C' \in \mathscr{C}_2$ of L and preimages $\{\text{rot}(P_i)\}(P_i \pm \mathscr{P}_2)$ of $\{M_{[p_i,q_i]}\}$.

(ii) By Lemma 1, $C'\prod_{i=1}^{m}\mathrm{rot}(P_i)$ and U may differ by a global phase ϕ, i.e., $U = e^{i\phi}C'\prod_{i=1}^{m}\mathrm{rot}(P_i)$. We claim that $\phi = \frac{\ell\pi}{4}$ for some $\ell \in [8]$. Since $e^{i\pi/4}\mathbb{I} = HSHSHS \in \mathscr{C}_1$, we output the concatenation of $(HSHSHS)^{\ell}_{(1)}C'\prod_{i=1}^{m}\mathrm{rot}(P_i)$ as the exact synthesis of U.

Given $U_1, U_2 \in \mathscr{J}_2$ such $U_1 = e^{i\phi}U_2$, entries of U_1 and U_2 belongs to $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ by Lemma 10. Then $e^{i\phi} \in \mathbb{Z}[\frac{1}{\sqrt{2}}, i]$. Thus ϕ must be a multiple of $\pi/4$. The above claim holds.

In the rest of this section, we focus on Problem 1. The following theorem, which we will prove later, is the main technical part.

Theorem 3. Given any $V \in \mathrm{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$ with $\tau(V) > 0$, there exists a sequence $M_{[p_1,q_1]}, M_{[p_1,q_2]}, \ldots, M_{[p_t,q_t]}, \ldots, M_{[p_m,q_m]}$ where $m \leqslant 10$ such that

$$\tau\left[\left(\prod_{i=1}^{t} M_{[p_i,q_i]}\right)V\left(\prod_{i=t+1}^{m} M_{[p_i,q_i]}\right)\right] < \tau(V).$$

In addition, the sequence $M_{[p_1,q_1]}, \ldots, M_{[p_t,q_t]}, \ldots, M_{[p_m,q_m]}$ can be found in constant time.

By Theorem 3 we can conclude Theorem 2 quickly.

Proof of Theorem 2. We first prove $\tau(\widehat{U}) \leqslant \mathscr{T}(U)$. By Lemma 1, U can be decomposed as

$$U = C\prod_{i=1}^{\mathscr{T}(U)}\mathrm{rot}(P_i),$$

where $C \in \mathscr{C}_2$ and $P_i \in \pm\mathscr{P}_2$. Observe that the LDE of matrices is submultiplicative, i.e., $\tau(AB) \leqslant \tau(A) + \tau(B)$ for $A, B \in \mathbb{Z}[\frac{1}{\sqrt{2}}]^{6\times 6}$. So

$$\begin{aligned}\tau(\widehat{U}) &= \tau\left(\widehat{C}\prod_{i=1}^{\mathscr{T}(U)}\widehat{\mathrm{rot}(P_i)}\right) \leqslant \tau(\widehat{C}) + \sum_{i=1}^{\mathscr{T}(U)}\tau[\widehat{\mathrm{rot}(P_i)}]\\ &= 0 + \sum_{i=1}^{\mathscr{T}(U)} 1 = \mathscr{T}(U).\end{aligned}$$

In the following, we show $\mathscr{T}(U) \leqslant 10\tau(\widehat{U})$. Let $V = \widehat{U} \in \mathrm{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$ and $N = \tau(V)$. By applying Theorem 3 N times, we can decompose V as

$$\begin{aligned}V &= \left(\prod_{i=1}^{t_1} M_{[p_1^{(i)},q_1^{(i)}]}\right)V_1\left(\prod_{i=t_1+1}^{m_1} M_{[p_1^{(i)},q_1^{(i)}]}\right)\\ &= \left(\prod_{k=1}^{N}\prod_{i=1}^{t_k} M_{[p_k^{(i)},q_k^{(i)}]}\right)V_N\left(\prod_{k=0}^{N-1}\prod_{i=t_{N-k}+1}^{m_{N-k}} M_{[p_{N-k}^{(i)},q_{N-k}^{(i)}]}\right)\\ &= V_N\left(\prod_{k=1}^{N}\prod_{i=1}^{t_k} V_N^T M_{[p_k^{(i)},q_k^{(i)}]}V_N\right)\left(\prod_{k=0}^{N-1}\prod_{i=t_{N-k}+1}^{m_{N-k}} M_{[p_{N-k}^{(i)},q_{N-k}^{(i)}]}\right)\\ &= V_N\left(\prod_{k=1}^{N}\prod_{i=1}^{t_k} M_{[r_k^{(i)},l_k^{(i)}]}\right)\left(\prod_{k=0}^{N-1}\prod_{i=t_{N-k}+1}^{m_{N-k}} M_{[p_{N-k}^{(i)},q_{N-k}^{(i)}]}\right),\end{aligned}$$

where $\tau(V_k) \leqslant N - k$; $0 \leqslant t_k \leqslant m_k \leqslant 10$; $r_k^{(i)}, l_k^{(i)}, p_k^{(i)}, q_k^{(i)} \in [6]$; and $M_{[r_k^{(i)},l_k^{(i)}]} = V_N^T M_{[p_k^{(i)},q_k^{(i)}]}V_N$.

Thus $\widehat{U}$ can be decomposed as the product of a matrix V_N with $\tau(V_N) = 0$ and $\sum_{k=1}^{N} m_k \leqslant N \times 10 = 10\tau(\widehat{U})$, $M_{[p,q]}$'s, which implies that U can be synthesized by using at most $10\tau(\widehat{U})$, T gates i.e., $\mathscr{T}(U) \leqslant 10\tau(\widehat{U})$. ■

Based on Theorem 2, we propose our exact synthesis algorithm, namely, Algorithm 1.

Remark. Indeed, by slightly adapting the proof of Theorem 2, we can conclude that any $V \in \mathrm{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$ can also be decomposed as the product of a matrix V_N with $\tau(V_N) = 0$ and at most $10\tau(V)$, $M_{[p,q]}$'s. Thus, for any $V \in \mathrm{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$, there exists a $U \in \mathscr{J}_2$ such that $\widehat{U} = V$, that is, $\mathrm{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}]) \subseteq \widehat{\mathscr{J}_2}$. Combined with the fact that $\widehat{\mathscr{J}_2} \subseteq \mathrm{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$, we have $\widehat{\mathscr{J}_2} = \mathrm{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$.

Proof of Theorem 1. By Theorem 2 and the discussion above Theorem 3, we know that $U = (HSHSHS)^{\ell}_{(1)}C'\prod_{i=1}^{m}\mathrm{rot}(P_i)$ and $m \leqslant 10\tau(\widehat{U})$.

In the following, we analyze the time complexity. Algorithm 1 costs constant time to execute lines 1, 2, 8, and 23 and costs $O(m)$ time to execute lines 21 and 22. The loop of lines 3–6 is repeated at most $\tau(\widehat{U}) \leqslant \mathscr{T}(U)$ times and each time needs constant time to execute due to Theorem 3. In addition, the loop of lines 11 and 12 is repeated $\sum_{i=1}^{l} t_i \leqslant m$ rounds and each round needs constant time to execute. Similarly, it costs $O(m)$ time to execute lines 15–20. Therefore, the total time complexity is $O(\mathscr{T}(U))$. ■

In the rest of this section, we prove Theorem 3. Given a $V \in \mathrm{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$, one can easily check that (i) right multiplying $M_{[p,q]}$, i.e., $V \to VM_{[p,q]}$, is equivalent to a column transformation

$$V[*,p], V[*,q] \to \frac{V[*,p] + V[*,q]}{\sqrt{2}}, \frac{V[*,q] - V[*,p]}{\sqrt{2}}$$

and (ii) left multiplying $M_{[p,q]}$, i.e., $V \to M_{[p,q]}V$, is equivalent to a row transformation

$$V[p,*], V[q,*] \to \frac{V[p,*] - V[q,*]}{\sqrt{2}}, \frac{V[p,*] + V[q,*]}{\sqrt{2}}.$$

Then having the LDEs and parities of all entries in V, Lemma 3 can tells how the LDEs are changed. For example, if $V_{\ell p}$ and $V_{\ell q}$ have the same LDE and the same parity, then right multiplying $M_{[p,q]}$ decreases their LDE.

The following lemma, about the distribution of LDEs in one row or one column, will be useful. Note that V is an orthogonal matrix, so the squares of entries in any row or any column sum up to 1.

Lemma 4. Given $x_1, x_2, \ldots, x_n \in \mathbb{Z}[\frac{1}{\sqrt{2}}]$ such that $\sum_{i=1}^{n} x_i^2 = 1$ and $\tau_0 = \tau(\{x_1, \ldots, x_n\}) \geqslant 1$, let N_ℓ denote the number of x_i with $\tau(x_i) = \ell$, $O_{\max}$ denote the number of odd x_i with $\tau(x_i) = \tau_0$, and $E_{\max}$ denote the number of even x_i with $\tau(x_i) = \tau_0$. Then (a) N_{τ_0}, $O_{\max}$, and $E_{\max}$ are even and (b) moreover, if $\tau_0 \geqslant 2$, then $N_{\tau_0}/2 + N_{\tau_0-1} \equiv 0 (\mathrm{mod}\, 2)$.

Proof. (a) For $i \in [n]$, $\sqrt{2}^{\tau_0}x_i \in \mathbb{Z}[\sqrt{2}]$. Recall that $\mathbb{Z}[\sqrt{2}]$ is a Euclidean domain. By Proposition 2,

$$\left(\sqrt{2}^{\tau_0}x_i\right)^2 \mathrm{mod}\, 2 = \begin{cases}1 & \text{if } \tau(x_i) = \tau_0\\ 0 & \text{if } \tau(x_i) < \tau_0.\end{cases}$$

In addition, note that

$$\sum_{i=1}^{n}\left(\sqrt{2}^{\tau_0}x_i\right)^2 \equiv 2^{\tau_0}\sum_{i=1}^{n}x_i^2 \equiv 2^{\tau_0} \equiv 0(\mathrm{mod}\,2).$$

Thus $N_{\tau_0} \equiv 0(\mathrm{mod}\,2)$, that is, N_{τ_0} is even.

Furthermore, observe that

$$\left(\sqrt{2}^{\tau_0}x_i\right)^2 \mathrm{mod}\,4 = \begin{cases} z_i + 2\sqrt{2} & \text{if } \tau(x_i) = \tau_0 \text{ and } x_i \text{ is odd} \\ z_i & \text{otherwise} \end{cases}$$

for some $z_i \in \mathbb{Z}$. In addition, note that

$$\sum_{i=1}^{n}\left(\sqrt{2}^{\tau_0}x_i\right)^2(\mathrm{mod}\,4) = 2^{\tau_0}(\mathrm{mod}\,4) \in \mathbb{Z}.$$

So we have $(\sum_{i=1}^{n} z_i + 2\sqrt{2}O_{\max})(\mathrm{mod}\,4) \in \mathbb{Z}$, which implies $O_{\max}$ is even. Finally, since $E_{\max} = N_{\tau_0} - O_{\max}$, $E_{\max}$ is also even.

(b) Without loss of generality, assume the entries with the LDE τ_0 are $x_1, x_2, \ldots, x_{N_{\tau_0}}$. By part (a), N_{τ_0} is even and $x_1, x_2, \ldots, x_{N_{\tau_0}}$ can be divided into $N_{\tau_0}/2$ pairs such that two elements in each pair have the same parity. Without loss of generality, assume x_{2i-1} and x_{2i} have the same parity for $i \in [N_{\tau_0}/2]$. Define

$$x'_{2i-1} = \frac{x_{2i-1} + x_{2i}}{\sqrt{2}}, \quad x'_{2i} = \frac{x_{2i-1} - x_{2i}}{\sqrt{2}}.$$

By Lemma 3 we have that $\tau(\{x'_1, \ldots, x'_{N_{\tau_0}}\}) = \tau_0 - 1$ and there are exactly $N_{\tau_0}/2$ elements with the LDE $\tau_0 - 1$ in $x'_1, \ldots, x'_{N_{\tau_0}}$.

Note that

$$\sum_{i=1}^{N_{\tau_0}} x_i'^2 + \sum_{i=N_{\tau_0}+1}^{n} x_i^2 = \sum_{i=1}^{n} x_i^2 = 1.$$

Then in the new sequence $x'_1, x'_2, \ldots, x'_{N_{\tau_0}}, x_{N_{\tau_0}+1}, \ldots, x_n$, the number of elements with the maximal LDE (now $\tau_0 - 1$) is even by Lemma 4. There are $N_{\tau_0}/2$ elements with the LDE $\tau_0 - 1$ in $x'_1, \ldots, x'_{N_{\tau_0}}$ and N_{τ_0-1} elements with the LDE $\tau_0 - 1$ in $x_{N_{\tau_0+1}}, \ldots, x_n$. So we can conclude that $N_{\tau_0}/2 + N_{\tau_0-1} \equiv 0(\mathrm{mod}\,2)$. ■

The following lemma considers the distribution of LDEs across two rows or two columns. Because V is an orthogonal matrix, the inner product of any two distinct rows or columns equals 0.

Lemma 5. Given $x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n \in \mathbb{Z}[\frac{1}{\sqrt{2}}]$ such that $\sum_{i=1}^{n} x_i y_i = 0$, let $\tau_0 = \tau(\{x_1, \ldots, x_n, y_1, \ldots, y_n\}) \geqslant 1$. Also let $N_{\max}$ be the number of indices i such that $\tau(x_i) = \tau(y_i) = \tau_0$, N_{pair} be the number of indices i such that $\{\tau(x_i), \tau(y_i)\} = \{\tau_0, \tau_0 - 1\}$, and $D_{\max}$ be the number of indices i such that $\tau(x_i) = \tau(y_i) = \tau_0$ and x_i and y_i have different parity. Then $N_{\max}$ and $(N_{\mathrm{pair}} + D_{\max})$ are both even.

Proof. Observe that

$$\left(\sqrt{2}^{\tau_0}x_i\right)\left(\sqrt{2}^{\tau_0}y_i\right)\mathrm{mod}\,2 = \begin{cases} 1 & \text{if } \tau(x_i) = \tau(y_i) = \tau_0,\ p(x_i) = p(y_i) \\ 1+\sqrt{2} & \text{if } \tau(x_i) = \tau(y_i) = \tau_0,\ p(x_i) \neq p(y_i) \\ \sqrt{2} & \text{if } \{\tau(x_i), \tau(y_i)\} = \{\tau_0, \tau_0 - 1\} \\ 0 & \text{otherwise,} \end{cases}$$

where $p(n)$ denotes the parity of $n \in \mathbb{Z}[\frac{1}{\sqrt{2}}]$. In addition, we have that

$$\sum_{i=1}^{n}\left(\sqrt{2}^{\tau_0}x_i\right)\left(\sqrt{2}^{\tau_0}y_i\right) \equiv 0(\mathrm{mod}\,2).$$

So $N_{\max} + (N_{\mathrm{pair}} + D_{\max})\sqrt{2} \equiv 0(\mathrm{mod}\,2)$, which directly implies the conclusion. ■

By Lemmas 4 and 5, all $V \in \mathrm{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$ with $\tau(V) \geqslant 1$ can be classified into eight cases, precisely, as follows.

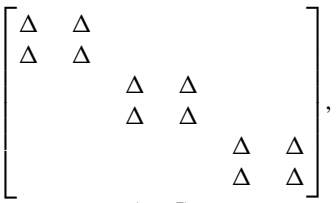
Lemma 6. For any $V \in \mathrm{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$ with $\tau(V) \geqslant 1$, by swapping rows or columns and transposing the matrix, V must satisfy one of eight patterns

$$\begin{bmatrix} \Delta & \Delta \\ \Delta & \Delta \\ & \\ & \\ & \\ & \end{bmatrix}, \quad \begin{bmatrix} \Delta & \Delta \\ \Delta & \Delta \\ \Delta & \Delta \\ \Delta & \Delta \\ & \\ & \end{bmatrix}, \quad \begin{bmatrix} \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta \\ & & & \\ & & & \end{bmatrix},$$

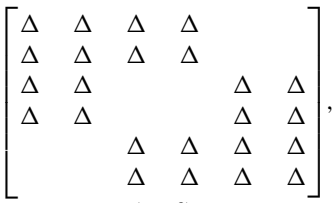
(case 1) (case 2) (case 3)

$$\begin{bmatrix} \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & & \\ \Delta & \Delta & & \\ & & & \\ & & & \end{bmatrix}, \quad \begin{bmatrix} \Delta & \Delta & & \\ \Delta & \Delta & & \\ & & \Delta & \Delta \\ & & \Delta & \Delta \\ & & & \\ & & & \end{bmatrix}, \quad \begin{bmatrix} \Delta & \Delta & \Delta & \Delta & & \\ \Delta & \Delta & \Delta & \Delta & & \\ \Delta & \Delta & & & \Delta & \Delta \\ \Delta & \Delta & & & \Delta & \Delta \\ & & & & & \\ & & & & & \end{bmatrix},$$

(case 4) (case 5) (case 6)



(case 7)



(case 8)

where the Δ's represent entries with a maximal LDE $\tau(V)$ and blanks are entries with an LDE less than $\tau(V)$.

Proof. By Lemmas 4(a) and 5, the number of Δ's in one row (or one column) and the number of common Δ's in two rows (or two columns) are both even. So positions of Δ's will form several 2×2 blocks. Additionally, the following two cases are impossible.

(i) First is one row (column) with pattern $P = [\Delta\ \ \Delta\ \ \Delta\ \ \Delta\ \ \Delta\ \ \Delta]$. Assume P exists. Then $N_{\tau(P)}/2 + N_{\tau(P)-1} = 3+0 \equiv 1 (\mathrm{mod}\, 2)$, contradicting Lemma 4(b).

(ii) The second case is two rows (columns) with pattern $Q = [^{\Delta\ \ \Delta\ \ \Delta\ \ \Delta}_{\ \ \ \ \ \ \Delta\ \ \Delta}]$. Assume Q exists. By Lemma 4(b), the first row of Q has an even number of elements with the LDE $\tau(Q) - 1$ and the second row has an odd number of elements with the LDE $\tau(Q) - 1$. Define N_{pair} and D_{max} for these two rows as in Lemma 5. Then we have N_{pair} is odd while D_{max} is 0. Thus $N_{\mathrm{pair}} + D_{\mathrm{max}}$ is odd, contradicting Lemma 5.

After ruling out all impossible cases, one can verify that there are only eight valid cases listed above. ■

Now we know exactly how entries with the maximal LDE are distributed. We will show that Theorem 3 holds for each case. We first introduce a useful lemma.

Lemma 7. Given $V \in \mathrm{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$, let $V' = VM_{[j,k]}$. If V_{ij} and V_{ik} are the only two entries with the LDE $\tau(V)$ in the ith row, then $\tau(V'[i,*]) < \tau(V)$. Moreover, if the ℓth row satisfies $\tau(V[\ell,*]) < \tau(V)$, then $\tau(V'[\ell,*]) < \tau(V)$.

By symmetry, let $V' = M_{[j,k]}V$. If V_{ji} and V_{ki} are the only two entries with the LDE $\tau(V)$ in the ith column, then $\tau(V'[*,i]) < \tau(V)$. Moreover, if the ℓth column satisfies $\tau(V[*,\ell]) < \tau(V)$, then $\tau(V'[*,\ell]) < \tau(V)$.

In other words, the two entries with the LDE $\tau(V)$ in the ith row (column) can be eliminated without creating entries with the LDE $\tau(V)$ in the ℓth row (column), if the ℓth row (column) has none.

Proof. $V[i,*]$ has only two entries V_{ij} and V_{ik} with the LDE $\tau(V)$. By Lemma 4(a), V_{ij} and V_{ik} have the same parity. Right multiplying $M_{[j,k]}$ will send V_{ij} and V_{ik} to $\frac{V_{ij}+V_{ik}}{2}$ and $\frac{V_{ik}-V_{ij}}{2}$, respectively. By Lemma 3, the LDEs of these two entries will both decrease.

Right multiplying $M_{[j,k]}$ will send $V_{\ell j}$ and $V_{\ell k}$ to $\frac{V_{\ell j}+V_{\ell k}}{2}$ and $\frac{V_{\ell k}-V_{\ell j}}{2}$, respectively. Since $\tau(V[\ell,*]) < \tau(V)$, $\tau(V[\ell,*])$ would increase to $\tau(V)$ only when $\{\tau(V_{\ell j}), \tau(V_{\ell k})\} = \{\tau(V) - 1, \tau_1\}$ for some $\tau_1 < \tau(V) - 1$. Assuming this case occurs, there will be exactly one $h \in [6]$ such that $\{\tau(V_{ih}), \tau(V_{\ell h})\} = \{\tau(V), \tau(V) - 1\}$. Thus $N_{\mathrm{pair}} = 1$. In addition, $D_{\mathrm{max}} = 0$ because $\tau(V[\ell,*]) < \tau(V[i,*])$. Thus $N_{\mathrm{pair}} + D_{\mathrm{max}} = 1$, contradicting Lemma 5.

The case for $V' = M_{[j,k]}V$ can be shown similarly and is omitted here. ■

Now we are ready to prove Theorem 3.

Proof of Theorem 3. For each of the eight cases in Lemma 6, we describe case by case how to transform one case to another case or to a matrix with a strictly lower LDE (for clarity, the transition between cases is summarized in Fig. 1).

Case 1. Right multiply $M_{[1,2]}$. By Lemma 7, the LDE of the upper left block $U[1:2,1:2]$ will decrease without creating entries with the LDE $\tau(V)$ in the other four rows. Then $\tau(VM_{[1,2]}) < \tau(V)$.

Case 2. Right multiply $M_{[1,2]}$. By Lemma 7, the LDE of the upper left block $U[1:4,1:2]$ will decrease without creating entries with the LDE $\tau(V)$ in the other two rows. Then $\tau(VM_{[1,2]}) < \tau(V)$.

Case 3. We claim that by at most five multiplications, the LDE of V will decrease. The proof is left to Appendix B.

Case 4. Right multiply $M_{[1,2]}$. By Lemma 7, the LDE of the 2×2 block $V[3:4,1:2]$ will decrease without creating entries with the LDE $\tau(V)$ in the last two rows. Then $VM_{[1,2]}$ will turn into case 1 or 2.

Case 5. Right multiply $M_{[1,2]}$. By Lemma 7, the LDE of the block $V[1:2,1:2]$ will decrease without creating entries with the LDE $\tau(V)$ in the last two rows, but the LDE of the block $V[3:4,1:2]$ may increase. Thus $VM_{[1,2]}$ will turn into case 1 or 2.

Case 6. Left multiply $M_{[3,4]}$. By Lemma 7, the LDE of the block $V[3:4,5:6]$ will decrease, but the LDE of the block $V[3:4,3:4]$ may increase. Thus $M_{[3,4]}V$ will turn into case 2, 3, or 4.

Case 7. Right multiply $M_{[1,2]}$. By Lemma 7, the LDE of the block $V[1:2,1:2]$ will decrease. Let $X = V[3:4,1:2]$ and $Y = V[5:6,1:2]$. It is impossible that the LDE of exactly one of X and Y increases, because otherwise the pattern $[^{\Delta\ \ \Delta\ \ \Delta\ \ \Delta}_{\ \ \ \ \ \ \Delta\ \ \Delta}]$ will occur in $VM_{[1,2]}$, which is proven impossible in the proof of Lemma 6. If neither X nor Y increases, it will turn into case 5. If both X and Y increase, it will turn into case 6.

Case 8. This case has two subcases.

Case (a). If V_{11} and V_{12} have the same parity, then right multiply $M_{[1,2]}M_{[3,4]}M_{[5,6]}$. We claim that $\tau(VM_{[1,2]}M_{[3,4]}M_{[5,6]}) < \tau(V)$. The proof is left to Appendix B.

Case (b). If V_{11} and V_{12} have different parities, then we claim that by at most four multiplications, V will turn into one of the cases from 1 to 6. The proof is left to Appendix B.

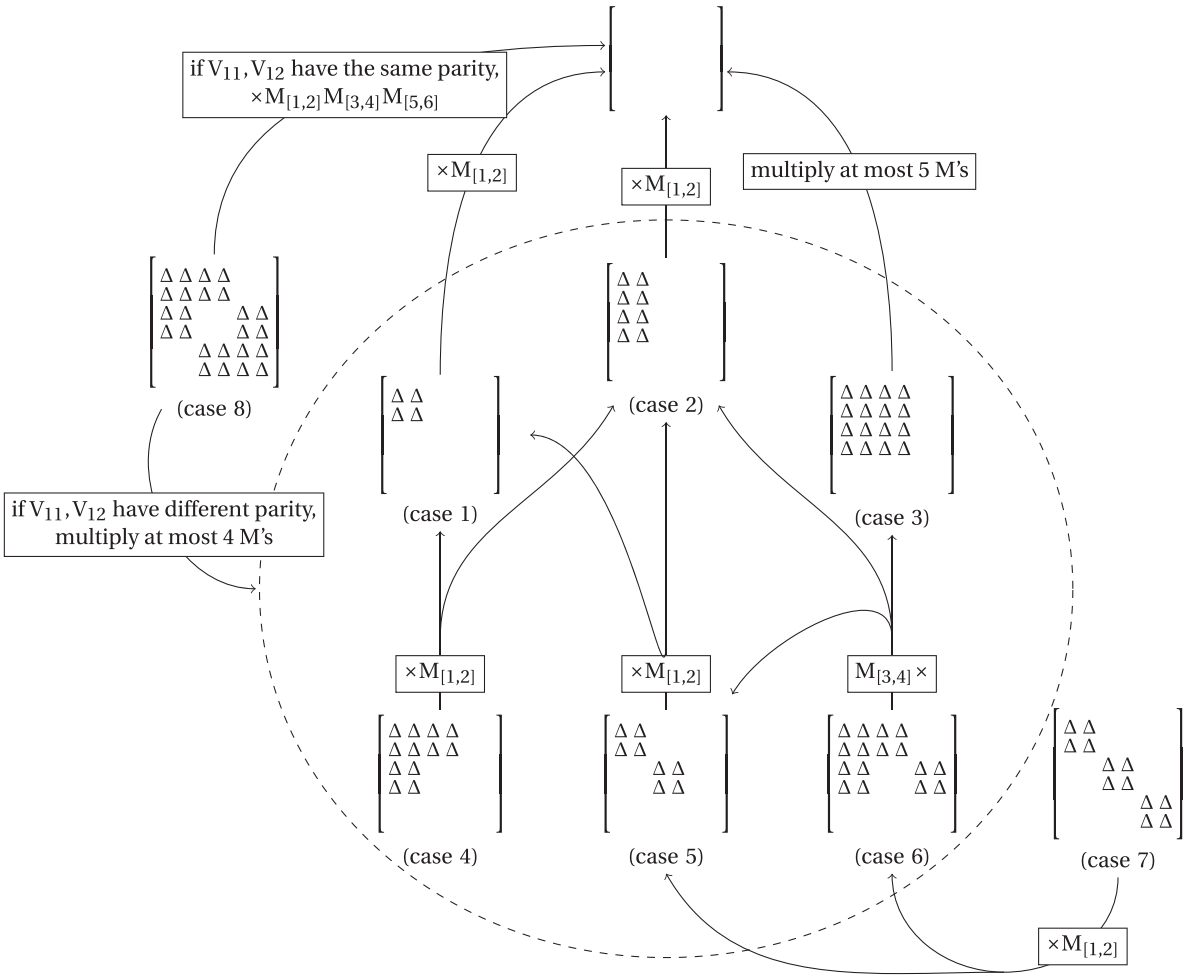


FIG. 1. Transition between cases.

In Table III we sum up the overall steps needed for each case to decrease the LDE of the whole matrix. The worst case is case 8(b), which needs ten steps. Thus for any $V \in \mathrm{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$, the LDE of V can be decreased by multiplying at most ten matrices from $\{M_{[p,q]}\}$. ■

TABLE III. Transition between cases. Dashes in column 2 mean the LDE will decrease. The number of steps is the number of steps needed to turn into the next case. The total number of steps is the total number of steps needed to decrease the LDE.

Case	Next case	No. of steps	Total no. of steps
1	–	1	1
2	–	1	1
3	–	5	5
4	1 or 2	1	2
5	1 or 2	1	2
6	2, 3, or 5	1	6
7	5 or 6	1	7
8a	–	3	3
8b	1–6	4	10

IV. APPROXIMATE SYNTHESIS OF TWO-QUBIT UNITARIES

In this section we apply our exact synthesis algorithm to directly obtain an ancilla-free approximate synthesis algorithm (Algorithm 2) for two-qubit unitaries where the number of T gates used is $O(\log(1/\epsilon))$, which turns out to be asymptotically optimal.

The approximate synthesis problem for two-qubit unitaries can be formally described as follows.

Problem 2. Given a $U \in \mathrm{SU}(4)$ and $\epsilon > 0$, find a Clifford+T circuit $C_0 \prod_{i=1}^{m} T_{(j_i)} C_i$, where $C_i \in \mathscr{C}_2$ and $j_i \in [n]$, satisfying

$$\left\| U - C_0 \prod_{i=1}^{m} T_{(j_i)} C_i \right\|_F \leqslant \epsilon$$

such that m is minimized. Here $\|\cdot\|_F$ is the Frobenius norm of matrices.

The idea of Algorithm 2 is as follows. Fix a $U \in \mathrm{SU}(4)$. First, we decompose U as $U = [\prod_{i=1}^{m} C_i G_{Z\otimes\mathbb{I}}(\theta_i) C_i^\dagger] C_0$, where each $C_i \in \mathscr{C}_2$ and $m \leqslant 15$ (Lemma 8). The definition of $G_{Z\otimes\mathbb{I}}(\theta_i)$ can be found above Lemma 8. Then we show that $G_{Z\otimes\mathbb{I}}(\theta)$ can be approximated with precision ϵ by a unitary in $\mathscr{J}_2$, namely, $T(\alpha, \beta)$, with $\tau[\widehat{T(\alpha, \beta)}] = O(\log 1/\epsilon)$

(Lemma 9). By Theorem 1, $T(\alpha,\beta)$ can be exactly synthesized using $O(\log 1/\epsilon)$ T gates.

Define $G_{Z\otimes\mathbb{I}}(\theta) := \exp(-\frac{\theta}{2}Z\otimes\mathbb{I})$, which is

$$G_{Z\otimes\mathbb{I}}(\theta) = \begin{pmatrix} e^{-i\theta} & 0 & 0 & 0 \\ 0 & e^{-i\theta} & 0 & 0 \\ 0 & 0 & e^{i\theta} & 0 \\ 0 & 0 & 0 & e^{i\theta} \end{pmatrix}.$$

We have the following lemma.

Lemma 8. Any $U \in \mathrm{SU}(4)$ can be decomposed as

$$U = \left(\prod_{i=1}^{m} C_i G_{Z\otimes\mathbb{I}}(\theta_i) C_i^\dagger\right) C_0,$$

where $m \leqslant 15$ and each $C_i \in \mathscr{C}_2$. Moreover, such a decomposition can be found in constant time.

The following lemma show that $G_{Z\otimes\mathbb{I}}(\theta)$ can be well approximated by a matrix

$$T(\alpha,\beta) = \begin{pmatrix} \alpha & 0 & -\beta^\dagger & 0 \\ 0 & \alpha & 0 & -\beta \\ \beta & 0 & \alpha^\dagger & 0 \\ 0 & \beta^\dagger & 0 & \alpha^\dagger \end{pmatrix},$$

where $\alpha,\beta \in \mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ satisfying $|\alpha|^2 + |\beta|^2 = 1$. By Lemma 10, $T(\alpha,\beta) \in \mathscr{J}_2$, i.e., it can be exactly synthesized into an ancilla-free Clifford+T circuit.

Lemma 9. Given any $\epsilon > 0$ and θ, there exist $\alpha,\beta \in \mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ satisfying $|\alpha|^2 + |\beta|^2 = 1$ such that $\|T(\alpha,\beta) - G_{Z\otimes\mathbb{I}}(\theta)\|_F \leqslant \epsilon$ and $\tau[\widehat{T(\alpha,\beta)}] = O(\log 1/\epsilon)$. Moreover, such an (α,β) can be found by a randomized algorithm with expected time complexity polylog$(1/\epsilon)$.

The proofs of Lemma 8 and 9 are left to Appendix C.

Now we are ready to present our approximate synthesis algorithm (Algorithm 2).

Theorem 4. We propose the following.

(a) The circuit G satisfies $\|G - U\|_F \leqslant \epsilon$.

(b) The expected time complexity of Algorithm 2 is polylog$(1/\epsilon)$.

(c) The number of T gates in G is $O(\log(1/\epsilon))$, which is asymptotically optimal.

Proof. Part (a) is obvious by noticing that $\|A_1A_2 - B_1B_2\|_F \leqslant \|A_1 - B_1\|_F + \|A_2 - B_2\|_F$.

Part (b) is proved as follows. Algorithm 2 costs constant time to execute line 1 by Lemma 8. The loop of lines 2–4 is repeated $m \leqslant 15$ times due to Lemma 8 and each time needs polylog$(1/\epsilon)$ to execute in expectation. In addition, line 5 needs constant time to execute and line 6 needs $O(\log(1/\epsilon))$ time due to Theorem 1 by noting that $\tau(\widehat{U^*}) \leqslant \sum_{i=1}^{m} \tau[\widehat{T(\alpha_i,\beta_i)}] = O(\log(1/\epsilon))$. So the total expected time complexity is polylog$(1/\epsilon)$.

Part (c) is proved as follows. From the proof of part (b) we know that $\tau(\widehat{U^*}) = O(\log(1/\epsilon))$. Then by Theorem 1, the number of T gates in G is $O(\log(1/\epsilon))$. Moreover, by Ref. [8], there exists a two-qubit unitary that needs a number $\Omega(\log(1/\epsilon))$ of T gates to approximate within precision ϵ, which implies asymptotic optimality. ■

V. CONCLUSION

In this paper we presented an ancilla-free exact synthesis algorithm for two-qubit Clifford+T unitaries where (i) the number of T gates used is at most $10\mathscr{T}(U)$ and (ii) the synthesis time is instance optimal. Our exact synthesis algorithm is based on the 2-to-1 homomorphism from SU(4) to SO(6). The SO(6) representation provides a clear characterization of $\mathscr{T}(U)$: For any two-qubit ancilla-free Clifford+T unitary, $\mathscr{T}(U)$ is equivalent to the least denominator exponent of its SO(6) representation up to a factor of 10. In addition, we also derive an approximate synthesis algorithm for two-qubit unitaries from our exact algorithm.

It remains open whether our algorithm can be generalized to more qubits. The main obstacle is to find a representation for $\mathrm{SU}(2^n)$ that has properties similar to the SO(6) representation. Unfortunately, even for three-qubit case, we are not aware of such a representation. We have tried a homomorphism $\mathrm{SU}(2^n) \to \mathrm{SO}(4^n)$ called channel representation [16], under which Clifford gates are also mapped to matrices with a zero LDE. However, the channel representation of a rot(P) contains more than one submatrix M, which becomes messy when we try applying the decomposing techniques in Theorem 3.

Another open problem is how to approximate a general unitary in $\mathrm{SU}(2^n)$ with precision ϵ by a unitary in $\mathscr{J}_n$ of minimum possible LDE. If this was solved, any exact synthesis algorithm could be directly translated into an approximate synthesis algorithm using near-optimal number of T gates.

ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China Grants No. 61832003, No. 62272441, and the Strategic Priority Research Program of Chinese Academy of Sciences Grant No. XDB28000000. Q.L. was additionally supported by the National Natural Science Foundation of China Grants No. 62002229.

APPENDIX A: ADDITIONAL PRELIMINARIES OF CLIFFORD+T CIRCUITS

Definition 5 (Pauli matrix). A matrix is called a Pauli matrix if it is the tensor product of the four basic Pauli matrices $\mathbb{I} = \left[\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right]$, $X = \left[\begin{smallmatrix}0 & 1\\ 1 & 0\end{smallmatrix}\right]$, $Y = \left[\begin{smallmatrix}0 & -i\\ i & 0\end{smallmatrix}\right]$, and $Z = \left[\begin{smallmatrix}1 & 0\\ 0 & -1\end{smallmatrix}\right]$.

Let $\mathscr{P}_n := \{\mathbb{I}, X, Y, Z\}^{\otimes n}$ denote the set of Pauli matrices acting on n qubits. In addition, we also use $\mathscr{P}_n^* := \mathscr{P}_n \setminus \{\mathbb{I}^{\otimes n}\}$ to denote the set of nontrivial Pauli matrices.

Definition 6 (Clifford group). The n-qubit Clifford group is defined as

$$\mathscr{C}_n := \{C \in U(2^n) \mid C\mathscr{P}_n C^\dagger \subseteq \pm\mathscr{P}_n\},$$

that is, Clifford matrices map Pauli matrices to Pauli matrices up to a global phase ± 1.

It is a basic fact that the Clifford group can be generated by the three basic gates

$$H = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix},$$

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

A Clifford+T circuit is a circuit consisting of only H gate, S gate, CNOT gate, and the T gate, which is a single-qubit gate defined as

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}.$$

A unitary is called Clifford+T unitary if it can be exactly synthesized into a Clifford+T circuit. Furthermore, if a Clifford+T unitary can be synthesized into Clifford+T circuits without using ancilla, we call the unitary ancilla-free. All n-qubit ancilla-free Clifford+T unitaries form a group $\mathscr{J}_n$.

Definition 7. Let $\mathscr{J}_n$ denote the group of n-qubit unitaries that can be exactly implemented by Clifford+T circuits without ancillas:

$$\mathscr{J}_n := \langle \mathscr{C}_n, T_{(1)} \rangle.$$

Giles and Selinger [15] provided a complete characterization of $\mathscr{J}_n$.

Lemma 10 (from [15]). Given a $U \in U(2^n)$, $U \in \mathscr{J}_n$ if and only if (i) all entries of U belong to $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ and (ii) $\det(U) = \exp(i\frac{\pi}{8}2^n r)$ for some $r \in [8]$.

In particular, a two-qubit unitary is an ancilla-free Clifford+T unitary if and only if its entries lie in ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ and its determinant is i^k for some $k \in [4]$.

We remark that there exist Clifford+T unitaries that are not ancilla-free. For example, the controlled-T gate is in $U_4(\mathbb{Z}[\frac{1}{\sqrt{2}}, i])$, so it is a Clifford+$T$ unitary [15], but it is not in $\mathscr{J}_2$ since its determinant is $e^{i\pi/4}$.

To close this section, we prove Lemma 1, which gives Clifford+T circuits a normal form.

Proof of Lemma 1. Suppose the circuit U implements $C_1, T_{(i_1)}, C_2, \ldots, T_{(i_m)}, C_{m+1}$ in order. Here $C_j \in \mathscr{C}_n$ and $T_{(i_j)}$ is applying the T gate on the i_jth qubit. Then

$$\begin{aligned} U &= C_{m+1}T_{(i_m)}C_m \cdots C_3 T_{(i_2)} C_2 T_{(i_1)} C_1 \\ &= C_{m+1}T_{(i_m)}C_m \cdots C_3 T_{(i_2)} C_2 C_1 (C_1^\dagger T_{(i_1)} C_1) \\ &= \left(\prod_{k=0}^{m} C_{m+1-k}\right) \cdots (C_1^\dagger C_2^\dagger T_{(i_2)} C_2 C_1)(C_1^\dagger T_{(i_1)} C_1) \\ &= C'_{m+1}\text{rot}(P_m)\cdots\text{rot}(P_1), \end{aligned}$$

where $C'_j = \prod_{k=0}^{j-1} C_{j-k}$ and $P_j = C_j'^\dagger Z_{(i_j)} C'_j \in \pm\mathscr{P}_n^*$.

Conversely, for any $P \in \pm\mathscr{P}_n^*$, there exists a Clifford $C \in \mathscr{C}_n$ such that $\text{rot}(P) = C^\dagger T_{(1)} C$. The explicit circuit construction can be found in Appendix A of Ref. [7]. Then the conclusion follows. ∎

APPENDIX B: OMITTED DETAILS IN THE PROOF OF THEOREM 3

In this Appendix we present the technical details of how to decrease $\tau(V)$ for cases 3 and 8. Recall that a matrix $V \in \text{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$ is in case 3 or 8 means that V is of the following pattern (possibly after swapping rows or columns and transposing the matrix):

$$\underset{\text{(case 3)}}{\begin{bmatrix} \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta \\ & & & \end{bmatrix}} \text{ or } \underset{\text{(case 8)}}{\begin{bmatrix} \Delta & \Delta & \Delta & \Delta & & \\ \Delta & \Delta & \Delta & \Delta & & \\ \Delta & \Delta & & & \Delta & \Delta \\ \Delta & \Delta & & & \Delta & \Delta \\ & & \Delta & \Delta & \Delta & \Delta \\ & & \Delta & \Delta & \Delta & \Delta \end{bmatrix}}.$$

Here Δ denotes an entry with an LDE $\tau(V)$ and a blank denotes an entry with an LDE less than $\tau(V)$. Moreover, the following denotations will also be used: δ is an entry with the LDE $\tau(V) - 1$; κ is an entry with the LDE less than or equal to $\tau(V) - 2$; $\star_E$ for $\star \in \{\Delta, \delta, \kappa\}$ is a $\star$ entry with even parity, for example, δ_E denotes an entry with the LDE $\tau(V) - 1$ and even parity; and $\star_O$ for $\star \in \{\Delta, \delta, \kappa\}$ is a $\star$ entry with odd parity.

The following lemma will be used.

Lemma 11. Given a matrix $V \in \text{SO}_6(\mathbb{Z}[\frac{1}{\sqrt{2}}])$ in case 3 or 8 and two rows $V[i, *]$ and $V[j, *]$, let (i) $D_{\max}$ be the number of indices $k \in [6]$ such that $\tau(V_{ik}) = \tau(V_{jk}) = \tau(V)$ and V_{ik} and V_{jk} have different parity and (ii) $S_{\max}$ be the number of indices $k \in [6]$ such that $\tau(V_{ik}) = \tau(V_{jk}) = \tau(V)$ and V_{ik} and V_{jk} have the same parity. Then $D_{\max}$ and $S_{\max}$ are both even. This lemma also holds for two columns $V[*, i]$ and $V[*, j]$.

Proof. Let N_{pair} denote the number of indices $k \in [6]$ such that $\{\tau(V_{ik}), \tau(V_{jk})\} = \{\tau(V), \tau(V) - 1\}$. By Lemma 5, $N_{\text{pair}} + D_{\max}$ is even. To show that $D_{\max}$ and $S_{\max}$ are both even, it suffices to show N_{pair} and $S_{\max} + D_{\max}$ are both even.

Case 3. Suppose V is in case 3. If i or $j \in \{5, 6\}$, we have $D_{\max} = S_{\max} = 0$ because $\tau(V[5:6, *]) < \tau(V)$. If $\{i, j\} \subset [4]$, then one can easily check that $N_{\text{pair}} = 0$ and $S_{\max} + D_{\max} = 4$.

Case 8. By symmetry, we only need to show the lemma when $(i, j) = (1, 2)$ or (1,3). If $(i, j) = (1, 2)$, then one can easily check that $N_{\text{pair}} = 0$ and $S_{\max} + D_{\max} = 4$. In the following, we assume $(i, j) = (1, 3)$. By Lemma 4(b), there is an even number of entries with the LDE $\tau(V) - 1$ in $V[1, *]$ and $V[3, *]$ each. Such an entry could be V_{15}, V_{16}, V_{33}, or V_{34}. Moreover, every such entry would contribute to N_{pair}. Thus N_{pair} is even. Finally, one can easily check that $S_{\max} + D_{\max} = 2$. ∎

1. Case 3

By discussing the parity of entries in $V[1:4, 1:4]$, we show that we can decrease the LDE of a matrix V in case 3 by at least 1 by using just at most five $M_{[i,j]}$'s.

Case 1. There exists a 2×2 block with the same parity. Without loss of generality, we assume $V[1:2, 1:2] = \begin{bmatrix} \Delta_E & \Delta_E \\ \Delta_E & \Delta_E \end{bmatrix}$.

(i) When V_{31} and V_{32} have the same parity, by Lemma 4 we know that the number of Δ_E's in each column is even. By

Lemma 11 we know that the number of pairs (Δ_E, Δ_E) and that of (Δ_O, Δ_O) between any two rows are both even. So we have that V must be in one of the following two patterns:

$$\begin{bmatrix} \Delta_E & \Delta_E & \Delta & \Delta \\ \Delta_E & \Delta_E & \Delta & \Delta \\ \Delta_E & \Delta_E & \Delta & \Delta \\ \Delta_E & \Delta_E & \Delta & \Delta \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \Delta_E & \Delta_E & \Delta & \Delta \\ \Delta_E & \Delta_E & \Delta & \Delta \\ \Delta_O & \Delta_O & \Delta & \Delta \\ \Delta_O & \Delta_O & \Delta & \Delta \end{bmatrix}.$$

Right multiplying $M_{[1,2]}$ would transform both of the above two patterns into case 2. Note that in case 2 we only need one multiplication to decrease $\tau(V)$, namely, right multiplying $M_{[3,4]}$. Thus, right multiplying $M_{[1,2]}M_{[3,4]}$ would decrease $\tau(V)$ in this case.

(ii) When V_{31} and V_{32} have different parity, without loss of generality, assume $V_{31} = \Delta_E$ and $V_{32} = \Delta_O$. By Lemma 4, the number Δ_E in any row or column is even. Thus we can conclude that $V_{41} = \Delta_E$, $V_{42} = \Delta_O$, and V_{43} and V_{44} have different parity. Without loss of generality, we assume $V_{43} = \Delta_E$ and $V_{44} = \Delta_O$. Thus V is in the following pattern:

$$\begin{bmatrix} \Delta_E & \Delta_E & \Delta & \Delta \\ \Delta_E & \Delta_E & \Delta & \Delta \\ \Delta_E & \Delta_O & \Delta & \Delta \\ \Delta_E & \Delta_O & \Delta_E & \Delta_O \end{bmatrix}.$$

(a) When V_{33} and V_{43} have the same parity, then V is in the pattern

$$\begin{bmatrix} \Delta_E & \Delta_E & \Delta & \Delta \\ \Delta_E & \Delta_E & \Delta & \Delta \\ \Delta_E & \Delta_O & \Delta_E & \Delta_O \\ \Delta_E & \Delta_O & \Delta_E & \Delta_O \end{bmatrix}.$$

Left multiplying $M_{[3,4]}$ will turn this case into case 2. Note that in case 2 we only need one multiplication to decrease $\tau(V)$, namely, left multiplying $M_{[1,2]}$. Thus, left multiplying $M_{[1,2]}M_{[3,4]}$ would decrease $\tau(V)$.

(b) When V_{33} and V_{43} have different parities, by Lemma 4, the parities of the other Δ's can be determined:

$$\begin{bmatrix} \Delta_E & \Delta_E & \Delta_E & \Delta_E \\ \Delta_E & \Delta_E & \Delta_O & \Delta_O \\ \Delta_E & \Delta_O & \Delta_O & \Delta_E \\ \Delta_E & \Delta_O & \Delta_E & \Delta_O \end{bmatrix}.$$

By pigeonhole principle, there must exist distinct $i, j \in [4]$ such that either $\tau(V_{5i}) = \tau(V_{5j}) = \tau(V) - 1$ or both $\tau(V_{5i})$ and $\tau(V_{5j})$ are less than $\tau(V) - 1$. Fix such a pair (i, j). In addition, by Lemma 4(b), for $k \in [4]$, either $\tau(V_{5k}) = \tau(V_{6k}) = \tau(V) - 1$ or both $\tau(V_{5k})$ and $\tau(V_{6k})$ are less than $\tau(V) - 1$. So right multiplying $M_{[i,j]}$ would not create entries with the LDE $\tau(V)$ in the fifth and sixth rows. Thus, right multiplying $M_{[i,j]}$ turns V into case 4, which needs two multiplications to decrease $\tau(V)$. Thus the $\tau(V)$ can be decreased by three multiplications.

Case II. There exists no 2×2 block with the same parity. Assume $V[1:2, 1:2] = \left[\begin{smallmatrix} \Delta_E & \Delta_E \\ \Delta_E & \Delta_O \end{smallmatrix}\right]$. By Lemma 4, the number Δ_E in any row or column is even. One can check that V must be in the following pattern (possibly after swapping rows or columns):

$$\begin{bmatrix} \Delta_E & \Delta_E & \Delta_O & \Delta_O \\ \Delta_E & \Delta_O & \Delta_E & \Delta_O \\ \Delta_O & \Delta_E & \Delta_O & \Delta_E \\ \Delta_O & \Delta_O & \Delta_E & \Delta_E \end{bmatrix}.$$

By Lemma 4(b), for $k \in [4]$, either $\tau(V_{5k}) = \tau(V_{6k}) = \tau(V) - 1$ (i.e., both V_{5k} and V_{6k} are δ) or both $\tau(V_{5k})$ and $\tau(V_{6k})$ are no more than $\tau(V) - 1$ (i.e., both V_{5k} and V_{6k} are κ). Similarly, for $k \in [4]$, either both V_{k5} and V_{k6} are δ or both are κ.

(i) If there exists $(i, j) \in \{(1, 2), (1, 3), (2, 4)\}$ such that both V_{5i} and V_{5j} are δ or both are κ, right multiplying $M_{[i,j]}$ turns this case to case 4 and then only two additional multiplication are needed to decrease $\tau(V)$.

(ii) If there exists $(i, j) \in \{(1, 2), (1, 3), (2, 4)\}$ such that both V_{i5} and V_{j5} are δ or both are κ, left multiplying $M_{[i,j]}$ turns this case to case 4 and then only two additional multiplication are needed to decrease $\tau(V)$.

(iii) Otherwise, V is in the pattern (possibly after swapping rows or columns)

$$\begin{bmatrix} \Delta_E & \Delta_E & \Delta_O & \Delta_O & \delta & \delta \\ \Delta_E & \Delta_O & \Delta_E & \Delta_O & \kappa & \kappa \\ \Delta_O & \Delta_E & \Delta_O & \Delta_E & \kappa & \kappa \\ \Delta_O & \Delta_O & \Delta_E & \Delta_E & \delta & \delta \\ \delta & \kappa & \kappa & \delta & & \\ \delta & \kappa & \kappa & \delta & & \end{bmatrix}. \tag{B1}$$

Furthermore, by Lemma 4(a), V must be in one of the following two patterns:

$$\begin{bmatrix} \Delta_E & \Delta_E & \Delta_O & \Delta_O & \delta & \delta \\ \Delta_E & \Delta_O & \Delta_E & \Delta_O & \kappa & \kappa \\ \Delta_O & \Delta_E & \Delta_O & \Delta_E & \kappa & \kappa \\ \Delta_O & \Delta_O & \Delta_E & \Delta_E & \delta & \delta \\ \delta & \kappa & \kappa & \delta & \kappa & \kappa \\ \delta & \kappa & \kappa & \delta & \kappa & \kappa \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \Delta_E & \Delta_E & \Delta_O & \Delta_O & \delta & \delta \\ \Delta_E & \Delta_O & \Delta_E & \Delta_O & \kappa & \kappa \\ \Delta_O & \Delta_E & \Delta_O & \Delta_E & \kappa & \kappa \\ \Delta_O & \Delta_O & \Delta_E & \Delta_E & \delta & \delta \\ \delta & \kappa & \kappa & \delta & \delta & \delta \\ \delta & \kappa & \kappa & \delta & \delta & \delta \end{bmatrix}. \tag{B2}$$

By right multiplying $M_{[1,2]}M_{[3,4]}$ and $M_{[1,3]}M_{[2,4]}$, respectively, pattern (B2) would be transformed into

$$\begin{bmatrix} & & & & \delta & \delta \\ \Delta & \Delta & \Delta & \Delta & \kappa & \kappa \\ \Delta & \Delta & \Delta & \Delta & \kappa & \kappa \\ & & & & \delta & \delta \\ \Delta & \Delta & \Delta & \Delta & \kappa & \kappa \\ \Delta & \Delta & \Delta & \Delta & \kappa & \kappa \end{bmatrix} \text{ or } \begin{bmatrix} \Delta & \Delta & \Delta & \Delta & \delta & \delta \\ & & & & \kappa & \kappa \\ & & & & \kappa & \kappa \\ \Delta & \Delta & \Delta & \Delta & \delta & \delta \\ \Delta & \Delta & \Delta & \Delta & \delta & \delta \\ \Delta & \Delta & \Delta & \Delta & \delta & \delta \end{bmatrix}.$$

We rearrange rows to make it more clear:

$$\begin{bmatrix} \Delta & \Delta & \Delta & \Delta & \kappa & \kappa \\ \Delta & \Delta & \Delta & \Delta & \kappa & \kappa \\ \Delta & \Delta & \Delta & \Delta & \kappa & \kappa \\ \Delta & \Delta & \Delta & \Delta & \kappa & \kappa \\ & & & & \delta & \delta \\ & & & & \delta & \delta \end{bmatrix} \text{ or } \begin{bmatrix} \Delta & \Delta & \Delta & \Delta & \delta & \delta \\ \Delta & \Delta & \Delta & \Delta & \delta & \delta \\ \Delta & \Delta & \Delta & \Delta & \delta & \delta \\ \Delta & \Delta & \Delta & \Delta & \delta & \delta \\ & & & & \kappa & \kappa \\ & & & & \kappa & \kappa \end{bmatrix}.$$

Note that the block entries in $V[1:4,5:6]$ are either all δ's or all κ's. Thus we get a matrix not of the pattern (B1). Finally, recalling that any other subcases of case 3 need at most three multiplications to decrease $\tau(V)$, we need at most five multiplications here.

2. Case 8

The following lemma, describing the changes of parity, will be used.

Lemma 12. Given $x, y \in \mathbb{Z}[\frac{1}{\sqrt{2}}]$, where $\tau(x) = k$ and $\tau(y) < k$, (i) if $\tau(y) = k-1$, the parities of $\frac{x\pm y}{\sqrt{2}}$ are both different from x, and (ii) if $\tau(y) < k-1$, the parities of $\frac{x\pm y}{\sqrt{2}}$ are both the same as x.

Proof. We write x as $\frac{a+b\sqrt{2}}{\sqrt{2}^k}$, where $a, b \in \mathbb{Z}$ and $a \bmod 2 = 1$. If $\tau(y) = k-1$, write y as $\frac{c+d\sqrt{2}}{\sqrt{2}^{k-1}}$, where $c, d \in \mathbb{Z}$ and $c \bmod 2 = 1$. Then

$$\frac{x \pm y}{\sqrt{2}} = \frac{(a \pm 2d) + (b \pm c)\sqrt{2}}{\sqrt{2}^{k+1}}.$$

Note that $(a \pm 2d) \bmod 2 = 1$ and $b \pm c \not\equiv b \pmod 2$. Therefore, the parities of $\frac{x\pm y}{\sqrt{2}}$ are both different from x.

If $\tau(y) < k-1$, write y as $\frac{c+d\sqrt{2}}{\sqrt{2}^{k-p}}$, where $c, d, p \in \mathbb{Z}$, $c \bmod 2 = 1$, and $p \geqslant 2$. Then

$$\frac{x \pm y}{\sqrt{2}} = \frac{a + b\sqrt{2} \pm 2 \times \sqrt{2}^{p-2}(c + d\sqrt{2})}{\sqrt{2}^{k+1}}.$$

So $\sqrt{2}^{k+1}\frac{x\pm y}{\sqrt{2}} \equiv \sqrt{2}^k x \pmod 2$ and then the parties of $\frac{x\pm y}{\sqrt{2}}$ are both the same as x. ■

Given a V in case 8, we show that we can decrease $\tau(V)$ by at least 1 by using just at most ten multiplications.

Case 8(a). Here V_{11} and V_{12} have the same parity. By Lemma 11 there is an even number of indices i such that (i) V_{1i} and V_{3i} have the same parity and (ii) $\tau(V_{1i}) = \tau(V_{3i}) = \tau(V)$. Thus V_{31} and V_{32} have the same parity. Similarly, V_{41} and V_{42} have the same parity. By applying Lemma 11 to the second and third rows, we have that V_{21} and V_{22} have the same parity. Thus, for $i \in [4]$, V_{i1} and V_{i2} have the same parity. Right multiplying $M_{[1,2]}$ would eliminate all entries with the LDE $\tau(V)$ in $V[1:4,1:2]$.

In addition, we claim that no entries with the LDE $\tau(V)$ are created in $V[5:6,1:2]$: Without loss of generality, suppose $\tau(V_{51})$ is changed to $\tau(V)$. Then, by applying Lemma 4(a) to the fifth row, we know $\tau(V_{52})$ is also changed to $\tau(V)$, so $\tau(V_{5j}) = \tau(V)$ for any $j \in [6]$, which contradicts Lemma 4(b).

By Lemma 4(a), the number of entries of the maximal LDE and a certain parity is even. Then V_{13} and V_{14} have the same parity and V_{35} and V_{36} have the same parity. Repeating the argument above twice, it can be shown that right multiplying $M_{[3,4]}$ and $M_{[5,6]}$ would eliminate all entries with the LDE $\tau(V)$ in $V[*,3:4]$ and $V[*,5:6]$. Thus, right multiplying $M_{[1,2]}M_{[3,4]}M_{[5,6]}$ decreases $\tau(V)$.

Case 8(b). Here V_{11} and V_{12} have different parities. If V_{11} and V_{21} have the same parity, then we transpose the matrix and go to case 8(a). In the following, we assume V_{11} and V_{21} have different parities.

Without loss of generality, we assume $V[1:2,1:2] = \begin{bmatrix} \Delta_E & \Delta_O \\ \Delta_O & \Delta_E \end{bmatrix}$. By Lemmas 4 and 11, the first and second rows each need another entry Δ_E and these two Δ_E's cannot be in the same column. By repeating this argument for each 2×2 block, one can check that V must satisfy one of the following patterns (possibly after swapping rows or columns):

$$\begin{bmatrix} \Delta_E & \Delta_O & \Delta_E & \Delta_O & & \\ \Delta_O & \Delta_E & \Delta_O & \Delta_E & & \\ \Delta_E & \Delta_O & & & \Delta_E & \Delta_O \\ \Delta_O & \Delta_E & & & \Delta_O & \Delta_E \\ & & \Delta_E & \Delta_O & \Delta_O & \Delta_E \\ & & \Delta_O & \Delta_E & \Delta_E & \Delta_O \end{bmatrix} \text{ or } \begin{bmatrix} \Delta_E & \Delta_O & \Delta_E & \Delta_O & & \\ \Delta_O & \Delta_E & \Delta_O & \Delta_E & & \\ \Delta_E & \Delta_O & & & \Delta_E & \Delta_O \\ \Delta_O & \Delta_E & & & \Delta_O & \Delta_E \\ & & \Delta_E & \Delta_O & \Delta_E & \Delta_O \\ & & \Delta_O & \Delta_E & \Delta_O & \Delta_E \end{bmatrix}.$$

Furthermore, by Lemma 4(b), there must be an even number of δ's in each row (column). So V must be in one of the following patterns:

$$\begin{bmatrix} \Delta_E & \Delta_O & \Delta_E & \Delta_O & & \\ \Delta_O & \Delta_E & \Delta_O & \Delta_E & & \\ \Delta_E & \Delta_O & \kappa & \kappa & \Delta_E & \Delta_O \\ \Delta_O & \Delta_E & \kappa & \kappa & \Delta_O & \Delta_E \\ \kappa & \kappa & \Delta_E & \Delta_O & \Delta & \Delta \\ \kappa & \kappa & \Delta_O & \Delta_E & \Delta & \Delta \end{bmatrix}, \quad \begin{bmatrix} \Delta_E & \Delta_O & \Delta_E & \Delta_O & & \\ \Delta_O & \Delta_E & \Delta_O & \Delta_E & & \\ \Delta_E & \Delta_O & \delta & \delta & \Delta_E & \Delta_O \\ \Delta_O & \Delta_E & \delta & \delta & \Delta_O & \Delta_E \\ \kappa & \kappa & \Delta_E & \Delta_O & \Delta & \Delta \\ \kappa & \kappa & \Delta_O & \Delta_E & \Delta & \Delta \end{bmatrix},$$

$$\begin{bmatrix} \Delta_E & \Delta_O & \Delta_E & \Delta_O & & \\ \Delta_O & \Delta_E & \Delta_O & \Delta_E & & \\ \Delta_E & \Delta_O & \kappa & \kappa & \Delta_E & \Delta_O \\ \Delta_O & \Delta_E & \kappa & \kappa & \Delta_O & \Delta_E \\ \delta & \delta & \Delta_E & \Delta_O & \Delta & \Delta \\ \delta & \delta & \Delta_O & \Delta_E & \Delta & \Delta \end{bmatrix}, \text{ or } \begin{bmatrix} \Delta_E & \Delta_O & \Delta_E & \Delta_O & & \\ \Delta_O & \Delta_E & \Delta_O & \Delta_E & & \\ \Delta_E & \Delta_O & \delta & \delta & \Delta_E & \Delta_O \\ \Delta_O & \Delta_E & \delta & \delta & \Delta_O & \Delta_E \\ \delta & \delta & \Delta_E & \Delta_O & \Delta & \Delta \\ \delta & \delta & \Delta_O & \Delta_E & \Delta & \Delta \end{bmatrix}.$$

After right multiplying $M_{[1,3]}M_{[2,4]}$, by Lemma 12, these patterns become

$$\begin{bmatrix} \Delta'_E & \Delta'_O & \Delta'_E & \Delta'_O & \Delta_E & \Delta_O \\ \Delta'_O & \Delta'_E & \Delta'_O & \Delta'_E & \Delta_O & \Delta_E \\ \Delta'_E & \Delta'_O & \Delta'_E & \Delta'_O & \Delta & \Delta \\ \Delta'_O & \Delta'_E & \Delta'_O & \Delta'_E & \Delta & \Delta \end{bmatrix}, \quad \begin{bmatrix} \Delta'_O & \Delta'_E & \Delta'_O & \Delta'_E & \Delta_E & \Delta_O \\ \Delta'_E & \Delta'_O & \Delta'_E & \Delta'_O & \Delta_O & \Delta_E \\ \Delta'_E & \Delta'_O & \Delta'_E & \Delta'_O & \Delta & \Delta \\ \Delta'_O & \Delta'_E & \Delta'_O & \Delta'_E & \Delta & \Delta \end{bmatrix},$$

$$\begin{bmatrix} \Delta'_E & \Delta'_O & \Delta'_E & \Delta'_O & \Delta_E & \Delta_O \\ \Delta'_O & \Delta'_E & \Delta'_O & \Delta'_E & \Delta_O & \Delta_E \\ \Delta'_O & \Delta'_E & \Delta'_O & \Delta'_E & \Delta & \Delta \\ \Delta'_E & \Delta'_O & \Delta'_E & \Delta'_O & \Delta & \Delta \end{bmatrix}, \text{ or } \begin{bmatrix} \Delta'_O & \Delta'_E & \Delta'_O & \Delta'_E & \Delta_E & \Delta_O \\ \Delta'_E & \Delta'_O & \Delta'_E & \Delta'_O & \Delta_O & \Delta_E \\ \Delta'_O & \Delta'_E & \Delta'_O & \Delta'_E & \Delta & \Delta \\ \Delta'_E & \Delta'_O & \Delta'_E & \Delta'_O & \Delta & \Delta \end{bmatrix}, \tag{B3}$$

where Δ' denotes an entry with the LDE $\tau(V)+1$. Then by two left multiplications on the last four rows, any case of pattern (B3) can be reduced to

$$\begin{bmatrix} Q & Q & Q & Q & Q & Q \\ Q & Q & Q & Q & Q & Q \\ P & P & P & P & P & P \\ P & P & P & P & P & P \\ P & P & P & P & P & P \\ P & P & P & P & P & P \end{bmatrix}, \tag{B4}$$

where $\tau(P) \leqslant \tau(V)$ and $\tau(Q) < \tau(V)$. Since the LDE of the first two rows is less than $\tau(V)$, pattern (B4) cannot be case 7 or 8. In summary, after four multiplications, we turn case 8(b) to some case among 1–6. Since any case among 1–6 needs at most six multiplications to decrease $\tau(V)$, case 8(b) needs at most ten multiplications.

APPENDIX C: OMITTED PROOFS IN SEC. IV

1. Proof of Lemma 8

Proof of Lemma 8. It is a basic fact in linear algebra that any $\widehat{U} \in \mathrm{SO}(6)$ can be decomposed in constant time into a multiplication of at most 15 Givens rotations $\{M(\theta_i)_{[p_i,q_i]}\}$ along with a diagonal matrix D. Here $M(\theta)_{[p,q]}$ is a 6×6 matrix defined similarly with $M_{[p,q]}$ in Sec. III except that $M = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix} 1 & -1 \\ 1 & 1 \end{smallmatrix}\right]$ is replaced with $M(\theta) = \left[\begin{smallmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{smallmatrix}\right]$.

It can be verified that the preimage of $M(\theta_i)_{[p_i,q_i]}$ can be written as $\exp(-\theta'_i P_i/2)$ for some $\theta'_i \in \{\pm\theta_i\}$ and $P_i \in \mathscr{P}_2^*$. In addition, $\exp(-\theta'_i P_i/2) = C_i G_{Z\otimes\mathbb{I}}(\theta'_i) C_i^\dagger$ for some $C_i \in \mathscr{C}_2$. In addition, the preimage of the orthogonal diagonal matrix D is some Clifford operator $C_0 \in \mathscr{C}_2$. Thus $U = [\prod_{i=1}^m C_i G_{Z\otimes\mathbb{I}}(\theta'_i) C_i^\dagger] C_0$. ∎

2. Proof of Lemma 9

Proof of Lemma 9. Let $z = e^{-i\theta/2}$. Then $\|T(\alpha,\beta) - G_{Z\otimes\mathbb{I}}(\theta)\|_F \leqslant \epsilon$ if and only if

$$\sqrt{4(\alpha\alpha^\dagger + \beta\beta^\dagger + zz^\dagger) - 4(\alpha z^\dagger + \alpha^\dagger z)} \leqslant \epsilon,$$

which is further equivalent to $\sqrt{8[1-\mathrm{Re}(\alpha z^\dagger)]} \leqslant \epsilon$, i.e., $1 - \epsilon^2/8 \leqslant \mathrm{Re}(\alpha z^\dagger) \leqslant 1$.

In addition, for $x \in \mathbb{Z}[\frac{1}{\sqrt{2}}, i]$, define the LDE of x as

$$\tau'(x) := \min\{k \in \mathbb{N} \mid \sqrt{2}^k x \in \mathbb{Z}[\sqrt{2}, i]\}.$$

For a matrix $T \in \mathbb{Z}[\frac{1}{\sqrt{2}}, i]^{4\times 4}$, define $\tau'(T) = \max_{i,j\in[4]} T_{ij}$. Following Definition 2.9 of Ref. [19], we have $\tau(\widehat{T}) \leqslant 2\tau'(T)+1$ for any $T \in \mathscr{J}_2$.

So finding a desired $T(\alpha,\beta)$ requires the following two steps: (i) Find a candidate $\alpha \in \mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ such that $1 - \frac{\epsilon^2}{8} \leqslant \mathrm{Re}(\alpha z^\dagger) \leqslant 1$ and (ii) then find a candidate $\beta \in \mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ such that $\beta\beta^\dagger = 1 - \alpha\alpha^\dagger$ and $\tau'[T(\alpha,\beta)] = O(\log 1/\epsilon)$. These two steps can be computed in the expected time complexity $\mathrm{polylog}(1/\epsilon)$. The algorithm can be found in Ref. [9]. ∎

[1] P. W. Shor, *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE, Piscataway, 1994), pp. 124–134
[2] L. K. Grover, *Proceedings of the 28th Annual ACM Symposium on Theory of Computing* (ACM, New York, 1996), pp. 212–219.
[3] J. Preskill, Quantum **2**, 79 (2018).
[4] E. T. Campbell, H. Anwar, and D. E. Browne, Phys. Rev. X **2**, 041021 (2012).
[5] S. Bravyi and A. Kitaev, Phys. Rev. A **71**, 022316 (2005).
[6] D. Litinski, Quantum **3**, 128 (2019).
[7] D. Gosset, V. Kliuchnikov, M. Mosca, and V. Russo, Quantum Inf. Comput. **14**, 1261 (2014).
[8] V. Kliuchnikov, D. Maslov, and M. Mosca, Phys. Rev. Lett. **110**, 190502 (2013).
[9] P. Selinger, Quantum Inf. Comput. **15**, 159 (2015).
[10] N. J. Ross and P. Selinger, Quantum Inf. Comput. **16**, 901 (2016).
[11] V. Kliuchnikov, arXiv:1306.3200.
[12] V. Gheorghiu, M. Mosca, and P. Mukhopadhyay, npj Quantum Inf. **8**, 141 (2022).
[13] M. Möttönen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, Phys. Rev. Lett. **93**, 130502 (2004).
[14] V. Kliuchnikov, D. Maslov, and M. Mosca, Quantum Inf. Comput. **13**, 607 (2013).
[15] B. Giles and P. Selinger, Phys. Rev. A **87**, 032332 (2013).
[16] M. Mosca and P. Mukhopadhyay, Quantum Sci. Technol. **7**, 015003 (2022).
[17] P. Niemann, R. Wille, and R. Drechsler, Quantum Inf. Process. **19**, 317 (2020).
[18] A. Baker, *Matrix Groups: An Introduction to Lie Group Theory* (Springer Science + Business Media, New York, 2012).
[19] A. N. Glaudell, N. J. Ross, and J. M. Taylor, npj Quantum Inf. **7**, 103 (2021).