

Trading T gates for dirty qubits in state preparation and unitary synthesis

Guang Hao Low^{1,2}, Vadym Kliuchnikov^{1,2}, and Luke Schaeffer^{1,3,4}

¹Quantum Architectures and Computation, Microsoft Research, Washington, Redmond, USA

²Azure Quantum, Microsoft, Washington, Redmond, USA

³Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

⁴Joint Center for Quantum Information and Computer Science, University of Maryland, Maryland, College Park, USA
2024-06-11

Efficient synthesis of arbitrary quantum states and unitaries from a universal fault-tolerant gate-set e.g. Clifford+T is a key subroutine in quantum computation. As large quantum algorithms feature many qubits that encode coherent quantum information but remain idle for parts of the computation, these should be used if it minimizes overall gate counts, especially that of the expensive T-gates. We present a quantum algorithm for preparing any dimension- N pure quantum state specified by a list of N classical numbers, that realizes a trade-off between space and T-gates. Our scheme uses $\mathcal{O}(\log(N/\epsilon))$ clean qubits and a tunable number of $\sim (\lambda \log(\frac{\log N}{\epsilon}))$ dirty qubits, to reduce the T-gate cost to $\mathcal{O}(\frac{N}{\lambda} + \lambda \log \frac{N}{\epsilon} \log \frac{\log N}{\epsilon})$. This trade-off is optimal up to logarithmic factors, proven through an unconditional gate counting lower bound, and is, in the best case, a quadratic improvement in T-count over prior ancillary-free approaches. We prove similar statements for unitary synthesis by reduction to state preparation. Underlying our constructions is a T-efficient circuit implementation of a quantum oracle for arbitrary classical data.

Contents

1	Introduction	2
2	Data-lookup oracle by a SelectSwap network	3
3	Arbitrary quantum state preparation	5
4	Unitary synthesis by state preparation	6
5	Lower bound	6
6	Conclusion	7
7	Developments after 2018 release of preprint	8
A	Purified density matrix preparation	10
B	Data-lookup oracle implementation details	11
C	Indicator Function Construction	15
D	Pure-state preparation implementation details	16

Output	Qubits	T count	T Depth	Valid λ
$ \psi\rangle$	[12] $\log N$	$N \log \frac{N}{\epsilon}$	$N \log \frac{N}{\epsilon}$	–
	[15] $\log N + \lambda$	$N \log \frac{N}{\epsilon}$	$\frac{N}{\log N + \lambda} \log \frac{N}{\epsilon} + \log N \log \frac{N}{\epsilon}$	$\mathcal{O}\left(\frac{N}{\log N \log \log N}\right) \cup \Omega(N)$
	Here $\log N + \lambda$	$\lambda \log \frac{N}{\epsilon} + \frac{N}{\lambda} \log \frac{1}{\epsilon'}$	$\frac{N}{\lambda} \log \frac{1}{\epsilon'} + \log N \log \frac{\lambda}{\epsilon'}$	$[\log \frac{1}{\epsilon'}, N \log \frac{1}{\epsilon'}]$
U	[12] $\log N$	$N^2 \log \frac{N}{\epsilon}$	$N^2 \log \frac{N}{\epsilon}$	–
	[15] $\log N + \lambda$	$N^2 \log \frac{N}{\epsilon}$	$\frac{N^2}{\log N + \lambda} \log \frac{N}{\epsilon} + N \log N \log \frac{N}{\epsilon}$	$\mathcal{O}\left(\frac{N}{\log N \log \log N}\right) \cup \Omega(N)$
	Here $\log N + \lambda$	$\lambda K \log \frac{N}{\epsilon} + \frac{KN}{\lambda} \log \frac{K}{\epsilon'}$	$\frac{KN}{\lambda} \log \frac{K}{\epsilon'} + K \log N \log \frac{K\lambda}{\epsilon'}$	$[\log \frac{K}{\epsilon'}, N \log \frac{K}{\epsilon'}]$
$ \psi_{\text{gb}}\rangle$	[8] $\log N + \log \frac{1}{\epsilon}$	$\log \frac{N}{\epsilon} + N$	$N + \log \frac{1}{\epsilon}$	–
	Here $\log N + \lambda \log \frac{1}{\epsilon}$	$\lambda \log \frac{N}{\epsilon} + \frac{N}{\lambda}$	$\frac{N}{\lambda} + \log \frac{\lambda}{\epsilon}$	$[1, N]$

Table 1: Big- $\mathcal{O}(\cdot)$ cost of preparing to error ϵ a dimension N arbitrary quantum state $|\psi\rangle$ Eq. (2), quantum state with garbage $|\psi_{\text{gb}}\rangle$ Eq. (4), and an $N \times N$ unitary U Eq. (3) where $K \leq N$ columns are fully specified. Above, $\epsilon' = \frac{\epsilon}{\log N}$, and we have rescaled $\lambda b \rightarrow \lambda$ where appropriate to facilitate comparison.

1 Introduction

Many quantum algorithms require coherent access to classical data, that is, data that can be queried in superposition through a unitary quantum operation. This property is crucial in obtaining quantum speedups for applications such as machine learning [1], simulation of physical systems [2, 3] and solving systems of linear equations [4, 5].

The nature of quantum-encoded classical data is itself varied. For example, quantum data regression [6] queries a classical list of N data-points a_x through a unitary data-lookup oracle [7]

$$O|x\rangle|0\rangle = |x\rangle|a_x\rangle. \quad (1)$$

Other applications, particularly in quantum chemistry [8] instead access Hamiltonian coefficient data through a unitary $A|0\rangle = |\psi\rangle$ that prepares these numbers as amplitudes in a normalized quantum state, or as probabilities in a purified density matrix. Even more generally, the central challenge is synthesizing some arbitrary unitary $A \in \mathbb{C}^{N \times N}$ of which $k \leq N$ columns are either partially or completely specified by a list of complex coefficients that is, say, provided on paper.

Synthesis of these data-access unitaries is typically a dominant factor in the overall algorithm cost. In any scalable approach to quantum computation, all unitaries decompose into a universal fault-tolerant quantum gate set, such as Clifford gates $\{\text{H}, \text{S}, \text{CNOT}\}$ and T gates [9]. Solovay and Kitaev [9] were the first to recognize that any single-qubit unitary could be ϵ -approximated using $\mathcal{O}(\log^c(1/\epsilon))$ fault-tolerant gates for $c = 3.97$, which was later improved to $c = 1$ [10, 11]. By bootstrapping these results, it is well-known that a roughly equal number of $\mathcal{O}(kN \log(N/\epsilon))$ [12] Clifford and non-Clifford gates suffice for arbitrary dimensions. Notably, the total gate count scaling is optimal in all parameters, following gate-counting arguments [13].

The possibility that T gates could be substantially fewer in number than the Clifford gates, however, is not excluded by known lower bounds. It is believed that fault-tolerant Clifford gates $\{\text{H}, \text{S}, \text{CNOT}\}$ will be cheap in most practical implementations of fault-tolerant quantum computation. In contrast, the equivalent cost of each fault-tolerant non-Clifford T gates, implemented at machine precision, is placed at a space-time volume $\approx (225 \text{ logical qubits}) \times (10 \text{ Clifford depth})$ for realistic estimates [14] based on |T> magic-state distillation at a physical error rate of $\approx 10^{-3}$.

We present an approach to arbitrary quantum state preparation and unitary synthesis that focuses on minimizing the T count. Unique to our approach is the exploitation of a variable number $\mathcal{O}(\lambda \log(1/\epsilon))$ of ancillary qubits, in a manner not considered by prior gate-counting arguments or algorithms. We find a $\mathcal{O}(\lambda)$ improvement in the T count and circuit depth while keeping the Clifford count unchanged, excluding logarithmic factors. Most surprisingly, its benefit far exceeds the naïve approach of applying these ancillary qubits to producing |T> magic-states for any $\lambda = \mathcal{O}(\sqrt{N})$, as seen in Table 1. In the best-case, the T count of $\tilde{\mathcal{O}}(\sqrt{N})$ is a square-root factor smaller than prior art, such as for for preparing arbitrary pure states

$$|\psi\rangle = \sum_{x=0}^{N-1} \frac{a_x}{\|\vec{a}\|_2} |x\rangle, \quad \|\vec{a}\|_q = \left(\sum_{j=0}^{N-1} |a_x|^q \right)^{1/q}, \quad (2)$$

Operation	Qubits	T count $\leq \cdot + \mathcal{O}(\log \cdot)$	T Depth $\mathcal{O}(\cdot)$
SELECT	$b + 2\lceil \log_2 N \rceil$	$4N$	N
SWAP	$bN + \lceil \log_2 N \rceil$	$8bN$	$\log N$
SELSWAP	$b\lambda + 2\lceil \log_2 N \rceil$	$4\lceil \frac{N}{\lambda} \rceil + 8b\lambda$	$\frac{N}{\lambda} + \log \lambda$
Fig. 1d	$b(\lambda + 1) + 2\lceil \log_2 N \rceil$	$8\lceil \frac{N}{\lambda} \rceil + 32b\lambda$	$\frac{N}{\lambda} + \log \lambda$

Table 2: Upper bounds on cost of possible implementations of the data lookup oracle O of Eq. (5). See Appendices B.1, B.2 and C for other variations, such as a linear-depth phase-incorrect version of SWAP using $\leq 4bN$ T gates and no additional ancillary qubits. Our results allow for a space-depth trade-off determined by a choice of $\lambda \in [1, N]$, with a minimized T gate complexity of $\mathcal{O}(\sqrt{bN})$ by choosing $\lambda = \mathcal{O}(\sqrt{N/b})$. Note that $b\lambda$ qubits of the Fig. 1d implementation may be dirty.

Moreover, we prove this approach realizes an optimal ancillary qubit and T count trade-off up to log factors.

In particular, our approach is always advantageous as all but a logarithmic number $\mathcal{O}(\log(N/\epsilon))$ of qubits, independent of λ , may be dirty, meaning that they start in, and are returned to the same undetermined initial state. At first glance, the full quadratic speedup is not always desirable as any clean ancillary qubit, initialized in the $|0\rangle$ state, is a resource that may be better allocated to magic-state distillation. However, dirty qubits may not be used for magic-state distillation, and are a resource typically abundant in many algorithms, such as quantum simulation by a linear combination of unitaries [2]. Even in the most pessimistic scenario where no dirty qubits are available, a reduction in the overall execution time of the algorithm, including the effective cost of magic-state distillation, is possible.

We also consider applications of our approach. For instance, a similar speedup to unitary synthesis

$$U = \left(\sum_{k=0}^K |u_k\rangle\langle k| \right) + \dots, \quad (3)$$

where $K \leq N$ columns are specified, follows by a well-known reduction based on Householder reflections. Improvements to state preparation with garbage

$$|\psi_{\text{gb}}\rangle = \sum_{x=0}^{N-1} \sqrt{\frac{a_x}{\|\vec{a}\|_1}} |x\rangle |\text{garbage}_x\rangle. \quad (4)$$

relevant to the most advanced quantum simulation techniques [16, 17, 8] are also presented in Appendix A.

Underlying our T gate scaling results is an improved implementation of a data-lookup oracle

$$O|x\rangle|0\rangle|0\rangle = |x\rangle|a_x\rangle|\text{garbage}_x\rangle. \quad (5)$$

Note the attached garbage state may always be uncomputed by applying O in reverse. We begin by describing our implementation of Eq. (5), which we call a ‘SELECTSWAP’ network, with costs outlined in Table 2. Subsequently, we apply SELTSWAP to the state preparation problem using the fact that there exists classical data such that preparing any $|\psi\rangle$ requires only $\mathcal{O}(\text{polylog}(N))$ queries and additional primitive quantum gates [18]. The reduction of unitary synthesis to state preparation is then described. Finally, we prove optimality of our approach through matching lower bounds, and discuss the results.

2 Data-lookup oracle by a SELTSWAP network

The unitary data-lookup oracle of Eq. (5) accepts an input number state $|x\rangle \in \mathbb{C}^N$ where $x \in \{0, 1, \dots, N-1\} \equiv [N]$, and returns an arbitrary b -bit number $a_x \in \{0, 1\}^b$. Our approach combines a multiplexer implementation of O [20], called SELECT and a unitary swap network SWAP, with costs summarized in Table 2.

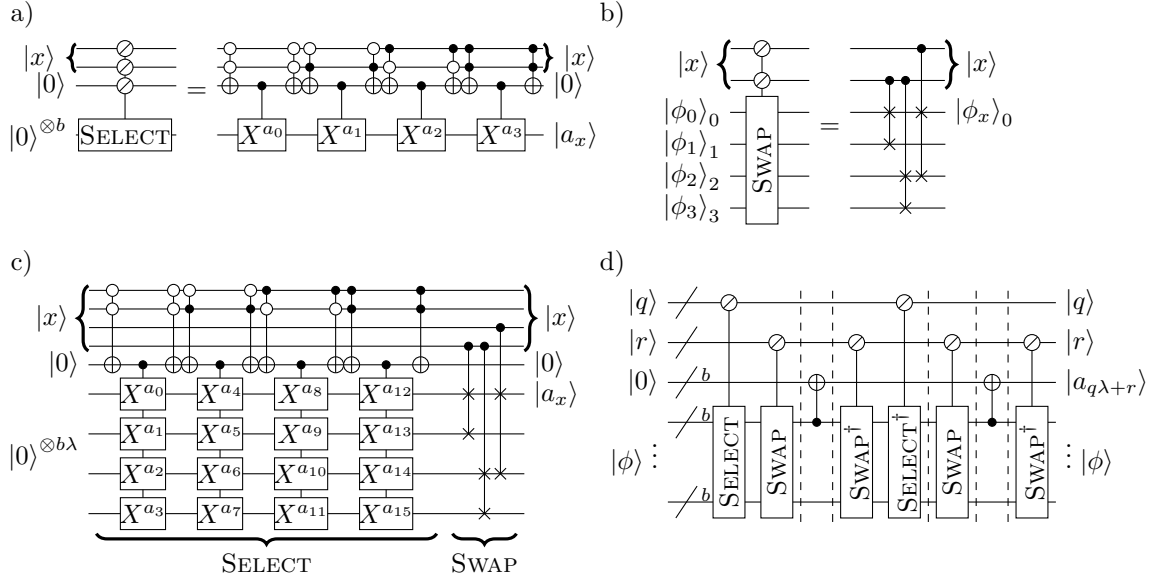


Figure 1: (a) Example SELECT operator $\sum_{x=0}^{N-1} |x\rangle\langle x| \otimes X^{a_x}$ with $N = 4$. The symbol \odot indicates control by a number state. A naive decomposition of all multiply-controlled-NOTs requires $\mathcal{O}(N \log N)$ Clifford+T gates and only one dirty qubit [19]. Cancellation of adjacent gates can reduce this to only $\mathcal{O}(N)$ [20, 8], but by using additional $\lceil \log_2 N \rceil$ clean qubits. (b) Example SWAP network with $N = 4$ using $\mathcal{O}(bN)$ Clifford+T. Any arbitrary state $|\phi_x\rangle_x$ in register index x is swapped to the $x = 0$ position. (c) The SELECTSWAP network with $N = 16, \lambda = 4$ that combines the above two approaches. (d) Modification of SELECTSWAP network that uses $2\lceil \log_2 N \rceil + b$ clean qubits and $b\lambda$ dirty qubits to implement the data-lookup oracle of Eq. (5) without garbage. We omit the SELECT ancillary qubits for clarity.

The SELECT operator applies some arbitrary unitary U_x controlled by the index state $|x\rangle$, that is

$$\text{SELECT} = \sum_{x=0}^{N-1} |x\rangle\langle x| \otimes U_x. \quad (6)$$

Thus O is realized by choosing $U_x = X^{a_x} \equiv \bigotimes_{j=0}^{b-1} X^{a_{x,j}}$ [8] to either be identity or the Pauli- X gate depending on the bit string a_x . As described in Fig. 1a, the costs, excluding $\{U_x\}$, is $\mathcal{O}(N)$ Clifford+T gates. As controlled- X is Clifford too, an additional $\mathcal{O}(bN)$ Clifford gates are applied. These CNOTs may be applied in logarithmic depth using an ancillary qubit free quantum fanout discussed in Appendix B.1.

The unitary SWAP network moves a b -qubit quantum register indexed by x to the $x = 0$ register, controlled by the state $|x\rangle$. For any quantum states $\bigotimes_{x=0}^{N-1} |\phi_x\rangle_x$,

$$\text{SWAP} \left[|x\rangle \bigotimes_{x=0}^{N-1} |\phi_x\rangle_x \right] = |x\rangle |\phi_x\rangle_0 \otimes \dots, \quad (7)$$

where the remaining quantum states (\dots) in registers $x > 0$ are unimportant.

As illustrated in Fig. 1b, this decomposes into a network of controlled-swap operators. As each controlled-swap operator decomposes into two CNOTs and one TOFFOLI, this network uses $\mathcal{O}(bN)$ Clifford+T gates. An ancillary qubit free logarithmic-depth version of SWAP is discussed in Appendix B.2

Our SELECTSWAP network illustrated in Fig. 1c is a simple hybrid of the above two schemes. Similar to the SWAP approach, we duplicate the b -bit register λ times, where $\lambda \in \{1, \dots, N\}$ is an integer. For λ that is not a power of 2, we compute $|x\rangle \rightarrow |q\rangle|r\rangle$, which is the quotient $q = x/\lambda$ and remainder $r = x \bmod \lambda$. This contributes an additive cost of $\mathcal{O}(\log N \log \lambda)$ gates. SELECT is controlled by $|q\rangle$ to write multiple values of a_x simultaneously into these duplicated registers by choosing $U_x = \bigotimes_{j=0}^{\lambda-1} X^{a_{xN/\lambda+j}}$, where $x \in [N/\lambda]$. SWAP is then controlled $|r\rangle$ to move the desired

data entry $|a_x\rangle$ to the output register. As the T gate complexity of $\mathcal{O}(\lambda b + \frac{N}{\lambda})$ is determined only by the dimension of the SELECT and SWAP control registers, this is minimized with value $\mathcal{O}(\sqrt{Nb})$ at $\lambda = \mathcal{O}(\sqrt{N/b})$.

Importantly, all but $b + \lceil \log_2 N \rceil$ of the qubits may be made dirty using a simple modification shown in Fig. 1d. Then for any computational basis state $|\phi\rangle = \otimes_{r=0}^{\lambda-1} |\phi_r\rangle_r$, and any input state $|x\rangle = |q\rangle|r\rangle$, let us evaluate $|0\rangle|\phi\rangle$ at each dotted line:

$$\begin{aligned} |0\rangle|\phi\rangle &\rightarrow |0\rangle|\phi_r \oplus a_x\rangle_0 \cdots \rightarrow |\phi_r \oplus a_x\rangle|\phi_r \oplus a_x\rangle_0 \cdots \\ &\rightarrow |\phi_r \oplus a_x\rangle|\phi\rangle \rightarrow |\phi_r \oplus a_x\rangle|\phi_r\rangle_0 \cdots \\ &\rightarrow |a_x\rangle|\phi_r\rangle_0 \cdots \rightarrow |a_x\rangle|\phi\rangle \end{aligned} \quad (8)$$

By linearity, this is true for all quantum states $|\phi\rangle$.

As the T gate complexity begins to increase with sufficiently large λ , one may simply elect to not use excess available dirty qubits. However, continued reduction of the T depth down to $\mathcal{O}(\log N)$ might be a useful property. In Appendix C we discuss an alternate construction that achieves logarithmic T depth and preserves the quadratic T count improvement for larger $\lambda = \Omega(\sqrt{N})$.

3 Arbitrary quantum state preparation

Preparation of an arbitrary dimension $N = 2^n$ quantum state $|\psi\rangle = \frac{1}{\|\vec{a}\|_2} \sum_{x \in \{0,1\}^n} a_x |x\rangle$ using the data-lookup oracle O of Eq. (5) is well-known in prior art. The basic idea was introduced by [21], and an ancillary-free implementation was presented in [12]. We outline the inductive argument of [18], and evaluate its cost using our SELECTSWAP implementation of O .

For any bit-string $y \in \{0,1\}^w$ of length $w \leq n$, let the probability that the first w qubits of $|\psi\rangle$ are in state $|y\rangle$ be $p_y = \frac{1}{\|\vec{a}\|_2^2} \sum_{\text{prefix}_w(x)=y} |a_x|^2$. Thus a single-qubit rotation $e^{-iY\theta}|0\rangle$ by angle $\theta = \cos^{-1} \sqrt{p_0}$ prepares the state $|\psi_1\rangle = \sqrt{p_0}|0\rangle + \sqrt{p_1}|1\rangle$, where p_0 is the probability that the first qubit of $|\psi\rangle$ is in state $|0\rangle$. We then recursively apply single-qubit rotations on the $(w+1)^{\text{th}}$ qubit conditioned on the first w qubits being in state $|y\rangle$. The rotation angles $\theta_y = \cos^{-1} \sqrt{p_{y0}/p_y}$ are chosen so that the state produced $|\psi_{w+1}\rangle$ reproduces the correct probabilities on the first $w+1$ qubits.

These conditional rotations are implemented using a sequence of data-lookup oracles O_1, \dots, O_{n-1} , where O_w stores a b -bit approximation of all θ_y where $y \in \{0,1\}^w$. At the w^{th} iteration,

$$\begin{aligned} |\psi_w\rangle &= \sum_{y \in \{0,1\}^w} \sqrt{p_y} |y\rangle \xrightarrow{O_w} \sum_{y \in \{0,1\}^w} \sqrt{p_y} |y\rangle |\theta_y\rangle \\ &\xrightarrow{} \sum_{y \in \{0,1\}^w} \sqrt{p_y} |y\rangle \left(\sqrt{\frac{p_{y0}}{p_y}} |0\rangle + \sqrt{\frac{p_{y1}}{p_y}} |1\rangle \right) |\theta_y\rangle \\ &\xrightarrow{O_w^\dagger} \sum_{y \in \{0,1\}^{w+1}} \sqrt{p_y} |y\rangle = |\psi_{w+1}\rangle. \end{aligned} \quad (9)$$

Note that we omit any garbage registers as they are always uncomputed. Also, the second line is implemented using b single-qubit rotations each controlled by a bit of θ_y . The complex phases of the target state $|\psi\rangle$ are applied to $|\psi_n\rangle$ by a final step with a data-lookup oracle storing $\phi_x = \arg[a_x/\sqrt{p_x}]$. Thus $\mathcal{O}(b \log N)$ single-qubit rotations are applied in total.

We implement these oracles with the SELECTSWAP network of Fig. 1, using a fixed value of λ for all O_k . A straightforward sum over the T count of Fig. 1 is $\mathcal{O}(b\lambda \log(N) + \frac{N}{\lambda})$, which is then added to the total T count of $\mathcal{O}(b \log(\frac{N}{\delta}))$ for synthesizing all single-qubit rotations each to error δ using the phase gradient technique [22], outlined in Appendix D. The error of the resulting state $|\psi'\rangle$ produced is determined by the number of bits b used to represent the rotation angles, in addition to rotation synthesis errors δ . Adding these errors leads to

$$\|\psi'\rangle - |\psi\rangle\| = \underbrace{\mathcal{O}(\delta)}_{\text{rotation synthesis}} + \underbrace{\mathcal{O}(2^{-b} \log N)}_{\text{bits of precision}} \leq \epsilon, \quad (10)$$

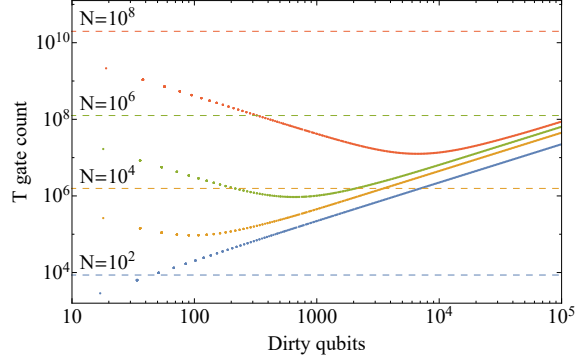


Figure 2: T gate count dependence on number of dirty qubits exploited for approximating an arbitrary quantum state of dimension $N = 10^{\{2,4,6,8\}}$ to error 10^{-3} using our algorithm (dots) in comparison with the standard ancillary-free approach (dashed) [12]. Note that at this error, $b \geq 17$ qubits are required to represent each coefficient in binary, but this may be halved by randomization techniques [23]. Moreover, one may always use fewer than the maximum number of available dirty qubits.

which is bounded by ϵ with the choice $b = \Theta(\log(\frac{\log N}{\epsilon}))$ and $\delta = \Theta(\epsilon)$. As a function of ϵ , the total T gate complexity is then $\mathcal{O}(\frac{N}{\lambda} + b(\lambda n + b))$, where

$$b(\lambda n + b) = \mathcal{O}\left(\lambda \log\left(\frac{N}{\epsilon}\right) \log\left(\frac{\log N}{\epsilon}\right)\right) \quad (11)$$

$$= \mathcal{O}\left(\lambda \log^2\left(\frac{N}{\epsilon}\right)\right), \quad (12)$$

and is plotted in Fig. 2.

4 Unitary synthesis by state preparation

The ability to prepare arbitrary quantum states enables synthesis of arbitrary unitaries $U \in \mathbb{C}^{N \times N}$. Given the matrix elements $\{|u_k\rangle \mid k \in [K]\}$ for the first K columns of U , the isometry synthesis problem is to find a quantum circuit that implements a unitary V that approximates U in the first K columns to error $\|U - V\|_{2,K} = \max_{x < K} \|(U - V)|x\rangle\| \leq \epsilon$.

We use the Householder reflections decomposition [24] to find a V that is a product of K reflections $\mathbb{I} - 2|v_k\rangle\langle v_k|$ and a diagonal gate $\text{diag}(e^{i\phi_1}, \dots, e^{i\phi_K}, 1, \dots, 1)$, for some set of quantum states $|v_k\rangle$. Note that this representation is not unique. The diagonal gate can be eliminated by using one ancillary qubit as discussed in [25]. There, it suffices to implement the unitary $W = |0\rangle\langle 1| \otimes U + |1\rangle\langle 0| \otimes U^\dagger$, which is equal to the product of reflections $\mathbb{I} - 2|w_k\rangle\langle w_k|$ where $|w_k\rangle = (|1\rangle \otimes |k\rangle - |0\rangle \otimes |u_k\rangle)/\sqrt{2}$ for $k \in [K]$.

Given a state preparation unitary $A_k|0\rangle = |u_k\rangle$, one can prepare state $|w_k\rangle$ starting from $|0\rangle \otimes |e_1\rangle$. Apply Hadamard to the first qubit, then a sequence of CNOT gates to prepare $|1\rangle \otimes |k\rangle + |0\rangle \otimes |0\rangle$, and finally apply controlled- A_k negatively controlled on the first qubit. Note that when this method is applied to the synthesis of sparse isometries, the states being synthesized are again sparse. Moreover, the cost of converting a state into a reflection doubles the number of non-Clifford gates. Thus the number of T gates used to synthesize an isometry is twice that for all the Controlled- A_k operations, and scales like $\mathcal{O}\left(K\left(\frac{N}{\lambda} + \lambda \log\left(\frac{N}{\epsilon}\right) \log\left(\frac{K \log N}{\epsilon}\right)\right)\right)$.

5 Lower bound

We prove the optimality of our construction through a circuit counting argument. The most general circuit on q qubits that uses Γ T-gates has the canonical form [26] $C \cdot \prod_{j=1}^{\Gamma} e^{-i\pi P_j/8}$, where each P_j is one of 4^q possible Pauli operators, and C is one of $2^{\mathcal{O}(q^2)}$ possible Clifford operators. Thus

the number of unique quantum circuits is at most

$$\text{Unique quantum circuits} = \mathcal{O}(4^{q\Gamma + \mathcal{O}(q^2)}). \quad (13)$$

A lower bound on the qubit and T-gate complexity of the data-lookup oracle of Eq. (5) is obtained by counting the number of unique Boolean functions $f : [N] \rightarrow \{0, 1\}^b$. As there are 2^{bN} such functions, we compare with Eq. (13). This leads to a lower bound on the space-T gate product

$$q\Gamma = \Omega(bN - q^2). \quad (14)$$

As the SELECTSWAP complexity in Fig. 1 is $q\Gamma = \mathcal{O}(\lambda^2 b^2 + bN + \log(N)(1/\lambda + \lambda b))$, this is optimal up to logarithmic factors so long as the number of T-gates dominates the qubit count like $\lambda = o(\sqrt{N/b})$, which is the case in most quantum circuits of interest.

A similar lower bound on state preparation is obtained by counting the number of dimension- N quantum states that are distinguishable with error ϵ . Without loss of generality, we only count quantum states $|\psi\rangle \in \mathbb{R}^N$ with real coefficients. These states live on the surface of a unit-ball \mathbb{B}_N of dimension N , with area $\text{Area}[\mathbb{B}_N] = \frac{2\pi^{N/2}}{(N/2-1)!}$. Let us now fix a state $|\psi\rangle$. Then the states $|\chi\rangle$ that satisfy $\| |\psi\rangle - |\chi\rangle \| \leq \epsilon$ live inside a ϵ -ball \mathbb{B}_{N-1} with volume $\mathcal{O}(\text{Vol}[\mathbb{B}_{N-1}]\epsilon^{N-1}) = \mathcal{O}(\frac{\pi^{N/2}}{(N/2)!}\epsilon^{N-1})$ [27]. Thus there are at least $\Omega(\frac{\text{Area}[\mathbb{B}_N]}{\text{Vol}[\mathbb{B}_{N-1}]\epsilon^{N-1}}) = \Omega(\sqrt{N}\epsilon^{-N+1})$ quantum states. Once again by comparing with Eq. (13), we obtain a T-gate lower bound of

$$q\Gamma = \Omega(N \log(1/\epsilon) - q^2). \quad (15)$$

This also matches the cost of our approach in Eq. (13) up to logarithmic factors, so long as $\lambda = o(\sqrt{N/\log(1/\epsilon)})$. The total number of isometries within at least distance ϵ from each other can also be estimated using Lemma 4.3 on Page 14 in [28], and is roughly $\Omega((1/\epsilon)^{KN})$. An analogous argument can be made for state preparation with garbage by considering the unit simplex instead of the unit ball.

Let us now establish a lower bound on state preparation that holds when measurements and arbitrary number of ancillae are used. For the purpose of the lower bound we also allow the use of post-selected measurements of multiple-qubit Pauli observables. Every preparation of an n -qubit state by Clifford+T circuit with ancillae and post-selected Pauli measurements can be rewritten [29] as the following sequence of operations: 1) initialization of Γ qubits into T state $|0\rangle + e^{i\pi/4}|1\rangle$; 2) post-selected measurement of $\Gamma - n$ commuting Pauli observables; 3) application of a Clifford unitary on Γ qubits. After the three steps first $\Gamma - n$ qubits are in a zero state and last n qubits are in the state being prepared. Let us count the number of distinct states that can be prepared by the steps described above. For the step two, there are at most $2 \cdot 4^\Gamma$ ways of choosing first Pauli observable and at most 4^Γ ways of choosing each of the remaining $\Gamma - n - 1$ observables because each of them needs to commute with the first observable. Therefore, on step two we have at most $2 \cdot 4^{\Gamma(\Gamma-n)}$ choices of Pauli observables. For the step three, two distinct Clifford unitaries can lead to preparing the same state and counting total number of Clifford unitaries on Γ qubits leads to an overestimate. The prepared $|\psi\rangle$ state is completely described by 4^n numbers $\alpha_P = \text{Tr}(P|\psi\rangle\langle\psi|)$ where P goes over all n -qubit Pauli matrices $\{I, X, Y, Z\}^{\otimes n}$. Let us count how many distinct 4^n dimensional vectors of α_P we can get on the step three by applying different Clifford unitaries C . Let ρ' be a density matrix describing a state of all qubits after step two, then $\alpha_P = \text{Tr}(C\rho'C^\dagger(I_{\Gamma-n} \otimes P))$ which is equal to $\text{Tr}(\rho'C^\dagger(I_{\Gamma-n} \otimes P)C)$. We see that the vector of α_P is uniquely defined by action of C on $2n$ Pauli operators $Z_{\Gamma-n+k}, X_{\Gamma-n+k}$ for k from 1 to n . There are at most $2 \cdot 4^{2\Gamma n}$ ways of choosing the action of C on the listed Pauli operators. Therefore we can prepare at most $4 \cdot 4^{\Gamma(\Gamma+n)}$ distinct states. This leads to the lower bound on the number of required T gates $\Omega(\sqrt{N \log N \log(1/\epsilon)})$.

6 Conclusion

We have shown that arbitrary quantum states with N coefficients, or unitaries with KN values specified by classical data may be synthesized with a T gate complexity that is an optimal $\tilde{\mathcal{O}}(\sqrt{N})$

reduction over prior art. As these subroutines are ubiquitous in many quantum algorithms we expect this result to be widely applicable.

We also expect our approach to be practical due to its almost exclusive usage of dirty qubits, which are typically abundant in larger quantum algorithms. Though our results are asymptotically optimal, constant factor and logarithmic improvements in costs could still be possible through careful optimization [30, 31]. For instance, our approach can be modified to use only $\mathcal{O}(1)$ additional clean qubits, but this increases the T count by a logarithmic factor. As more limited trade-offs between T gates and ancillary qubits are observed in other quantum circuits, such as for addition [22] or AND [19], a major open question highly relevant to implementation in nearer-term quantum computers, is whether such a property could be generic for many other quantum circuits and algorithms.

6.1 Acknowledgements

We thank Nicolas Delfosse, Jeongwan Haah, Robin Kothari, Jessica Lemieux, Martin Roetteler, and Matthias Troyer for insightful discussions.

7 Developments after 2018 release of preprint

There have been numerous improvements in trade-offs [15] between qubits, depth, and overall spacetime volume [32] of quantum circuits for state preparation and unitary synthesis [15], especially in connection to the block-encoding framework [16, 33] for matrices represented by classical data [34]. A notable new direction is error-resilient table-lookup [35], which achieves logarithmic error scaling by using a large number of $\mathcal{O}(N)$ qubits. However, minimizing use of expensive non-Clifford gates is rarely a priority in these results, and almost all exhibit the same scaling for number of Clifford and non-Clifford gates. We have updated Table 1 with a comparison to recent work [15] that claims an optimal qubit–depth trade-off. To facilitate comparison, we rescale λ and we use the tighter bound Eq. (11) on the T gate cost that was originally Eq. (12).

Our SELECTSWAP architecture of table lookup remains state-of-art. To date, all methods with an asymptotic advantage in non-Clifford gate cost for state preparation or unitary synthesis [34] do so by reduction to SELECTSWAP. The circuit implementation of SELECTSWAP has also seen some optimizations. By allowing intermediate circuit measurements, the multi-target CNOT_n gates in Appendix B.1 can be implemented in constant, rather than logarithmic depth, though at the cost of using an additional $\mathcal{O}(n)$ clean qubits. Intermediate measurements also enable uncomputing the garbage register Eq. (5) of SELECTSWAP using $4\lceil \frac{N}{\lambda} \rceil + 4\lambda$ T gates and $\lambda + \lceil \log(N/\lambda) \rceil$ clean qubits, which is independent of b [36].

The T count reductions enabled by SELECTSWAP have been key to numerous state-of-art resource estimates for quantum computing applications, such as in chemistry [36, 37, 38, 39], and other classical-data-intense routines [40, 34].

References

- [1] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. “Quantum principal component analysis”. *Nature Physics* **10**, 631–633 (2014).
- [2] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. “Simulating Hamiltonian dynamics with a truncated Taylor series”. *Physical Review Letters* **114**, 090502 (2015).
- [3] Guang Hao Low and Isaac L. Chuang. “Optimal Hamiltonian simulation by quantum signal processing”. *Physical Review Letters* **118**, 010501 (2017).
- [4] Aram W. Harrow, Avinandan Hassidim, and Seth Lloyd. “Quantum algorithm for linear systems of equations”. *Physical Review Letters* **103**, 150502 (2009).
- [5] Chunhao Wang and Leonard Wossnig. “A quantum algorithm for simulating non-sparse Hamiltonians”. *Quantum Information & Computation* **20**, 597–615 (2020).
- [6] Nathan Wiebe, Daniel Braun, and Seth Lloyd. “Quantum algorithm for data fitting”. *Physical Review Letters* **109**, 050505 (2012).

- [7] Dorit Aharonov and Amnon Ta-Shma. “Adiabatic quantum state generation and statistical zero knowledge”. [Proceedings of the thirty-fifth ACM symposium on Theory of computing-Page 20](#) (2003).
- [8] Ryan Babbush, Craig Gidney, Dominic W. Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. “Encoding electronic spectra in quantum circuits with linear T complexity”. [Physical Review X](#) **8**, 041015 (2018).
- [9] Michael A. Nielsen and Isaac L. Chuang. “Quantum computation and quantum information”. [Cambridge University Press](#). (2010). 10th anniversary edition.
- [10] Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. “Fast and efficient exact synthesis of single-qubit unitaries generated by Clifford and T gates”. [Quantum Information & Computation](#) **13**, 607–630 (2013).
- [11] Neil J Ross and Peter Selinger. “Optimal ancilla-free Clifford+T approximation of Z-rotations”. [Quantum Information & Computation](#) **16**, 901–953 (2016).
- [12] V. V. Shende, S. S. Bullock, and I. L. Markov. “Synthesis of quantum-logic circuits”. [IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems](#) **25**, 1000–1010 (2006).
- [13] Aram W. Harrow, Benjamin Recht, and Isaac L. Chuang. “Efficient discrete approximations of quantum gates”. [Journal of Mathematical Physics](#) **43**, 4445–4451 (2002).
- [14] Daniel Litinski. “A game of surface codes: large-scale quantum computing with lattice surgery”. [Quantum](#) **3**, 128 (2019).
- [15] Xiaoming Sun, Guojing Tian, Shuai Yang, Pei Yuan, and Shengyu Zhang. “Asymptotically optimal circuit depth for quantum state preparation and general unitary synthesis”. [IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems](#) **42**, 3301–3314 (2023).
- [16] Guang Hao Low and Isaac L. Chuang. “Hamiltonian simulation by qubitization”. [Quantum](#) **3**, 163 (2019).
- [17] Guang Hao Low and Isaac L. Chuang. “Hamiltonian simulation by uniform spectral amplification” (2017). [arXiv:1707.05391](#).
- [18] Scott Aaronson. “The complexity of quantum states and transformations: From quantum money to black holes” (2016). [arXiv:1607.05256](#).
- [19] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. “Elementary gates for quantum computation”. [Physical Review A](#) **52**, 3457–3467 (1995).
- [20] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. “Toward the first quantum simulation with quantum speedup”. [Proceedings of the National Academy of Sciences](#) **115**, 9456–9461 (2018).
- [21] Lov Grover and Terry Rudolph. “Creating superpositions that correspond to efficiently integrable probability distributions” (2002). [arXiv:quant-ph/0208112](#).
- [22] Craig Gidney. “Halving the cost of quantum addition”. [Quantum](#) **2**, 74 (2018).
- [23] Guang Hao Low. “Halving the cost of quantum multiplexed rotations” (2021). [arXiv:2110.13439](#).
- [24] Alston S. Householder. “Unitary triangularization of a nonsymmetric matrix”. [Journal of the ACM](#) **5**, 339–342 (1958).
- [25] Vadym Kliuchnikov. “Synthesis of unitaries with Clifford+T circuits” (2013). [arXiv:1306.3200](#).
- [26] David Gosset, Vadym Kliuchnikov, Michele Mosca, and Vincent Russo. “An algorithm for the T-count”. [Quantum Information & Computation](#) **14**, 1261–1276 (2014).
- [27] Caterina E. Mora and Hans J. Briegel. “Algorithmic complexity and entanglement of quantum states”. [Physical Review Letters](#) **95**, 200503 (2005).
- [28] E. Knill. “Approximation by quantum circuits” (1995). [arXiv:quant-ph/9508006](#).
- [29] Michael Beverland, Earl Campbell, Mark Howard, and Vadym Kliuchnikov. “Lower bounds on the non-clifford resources for quantum computations”. [Quantum Science and Technology](#) **5**, 035009 (2020).
- [30] Olivia Di Matteo, Vlad Gheorghiu, and Michele Mosca. “Fault-tolerant resource estimation of quantum random-access memories”. [IEEE Transactions on Quantum Engineering](#) **1**, 1–13 (2020).

- [31] Thomas Häner, Vadym Kliuchnikov, Martin Roetteler, and Mathias Soeken. “Space-time optimized table lookup” (2022). [arXiv:2211.01133](#).
- [32] Kaiwen Gui, Alexander M. Dalzell, Alessandro Achille, Martin Suchara, and Frederic T. Chong. “Spacetime-efficient low-depth quantum state preparation with applications”. *Quantum* **8**, 1257 (2024).
- [33] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. “The power of block-encoded matrix powers: Improved regression techniques via faster Hamiltonian simulation”. *46th International Colloquium on Automata, Languages, and Programming* **132**, 33:1–33:14 (2019).
- [34] B. David Clader, Alexander M. Dalzell, Nikitas Stamatopoulos, Grant Salton, Mario Berta, and William J. Zeng. “Quantum resources required to block-encode a matrix of classical data”. *IEEE Transactions on Quantum Engineering* **3**, 1–23 (2022).
- [35] Connor T. Hann, Gideon Lee, S.M. Girvin, and Liang Jiang. “Resilience of quantum random access memory to generic noise”. *PRX Quantum* **2**, 020311 (2021).
- [36] Dominic W. Berry, Craig Gidney, Mario Motta, Jarrod R. McClean, and Ryan Babbush. “Qubitization of arbitrary basis quantum chemistry leveraging sparsity and low rank factorization”. *Quantum* **3**, 208 (2019).
- [37] Vera von Burg, Guang Hao Low, Thomas Häner, Damian S. Steiger, Markus Reiher, Martin Roetteler, and Matthias Troyer. “Quantum computing enhanced computational catalysis”. *Physical Review Research* **3**, 033055 (2021).
- [38] Joonho Lee, Dominic W. Berry, Craig Gidney, William J. Huggins, Jarrod R. McClean, Nathan Wiebe, and Ryan Babbush. “Even more efficient quantum computations of chemistry through tensor hypercontraction”. *PRX Quantum* **2**, 030305 (2021).
- [39] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan. “Quantum algorithms for quantum chemistry and quantum materials science”. *Chemical Reviews* **120**, 12685–12717 (2020).
- [40] Yuval R. Sanders, Dominic W. Berry, Pedro C.S. Costa, Louis W. Tessler, Nathan Wiebe, Craig Gidney, Hartmut Neven, and Ryan Babbush. “Compilation of fault-tolerant quantum heuristics for combinatorial optimization”. *PRX Quantum* **1**, 020312 (2020).
- [41] Steven A Cuccaro, Thomas G Draper, Samuel A Kutin, and David Petrie Moulton. “A new quantum ripple-carry addition circuit” (2004). [arXiv:quant-ph/0410184](#).
- [42] Thomas Häner, Martin Roetteler, and Krysta M Svore. “Factoring using $2n + 2$ qubits with Toffoli based modular multiplication”. *Quantum Information & Computation* **17**, 673–684 (2017).
- [43] Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. “Asymptotically optimal approximation of single qubit unitaries by Clifford and T circuits using a constant number of ancillary qubits”. *Physical Review Letters* **110**, 190502 (2013).

A Purified density matrix preparation

In some applications, particular in quantum simulation based on a linear combination of unitaries or qubitization [2, 16], it suffices to prepare the density matrix $\rho = \sum_{x=0}^{N-1} \frac{|a_x|}{\|\vec{a}\|_1} |x\rangle\langle x|$ through a quantum state $|\psi\rangle = \sum_{x=0}^{N-1} \sqrt{\frac{a_x}{\|\vec{a}\|_1}} |x\rangle|\text{garbage}_x\rangle$ of Eq. (4) where the number state $|x\rangle$ is entangled with some garbage that depends only on x . By allowing garbage, it was shown by [8] that strictly linear T gate complexity in N is achievable, using a SELECT data-lookup oracle corresponding to the $\lambda = 1$ case of Table 2. We outline the original idea, then generalize the procedure using the SELECTSWAP network, which enables sublinear T gate complexity and better error scaling than the garbage-free approach. As density matrices have positive diagonals, we only consider the case of positive $a_x \geq 0$.

The original approach is based on a simple observation. By comparing a b -bit number state $|a\rangle$ together with a uniform superposition state $|u_{2^b}\rangle = \frac{1}{\sqrt{2^b}} \sum_{j=0}^{2^b-1} |j\rangle$ over 2^b elements, $|a\rangle$ may be mapped to

$$|a\rangle \mapsto |a\rangle \left(\sqrt{\frac{a}{2^b}} |0\rangle |u_a\rangle + \sqrt{\frac{2^b - a}{2^b}} |1\rangle |u_{\geq a}\rangle \right), \quad (16)$$

where we denote a uniform superposition after the first a elements by $|u_{\geq a}\rangle = \sum_{j=a}^{2^b-1} \frac{|j\rangle}{\sqrt{2^b-a}}$. This may be implemented using quantum addition [41], which costs $\mathcal{O}(b)$ Clifford+T gates with depth $\mathcal{O}(b)$.

This observation is converted to state-preparation in four steps. First, the normalized coefficients $a_x \frac{N2^b}{\|\vec{a}\|_1} \approx a'_x$ are rounded to nearest integer values such that $\|\vec{a}'\|_1 = N2^b$. Second, the data-lookup oracle that writes two numbers $a''_x \in [2^b]$ and $f(x) \in [N]$ such that $a'_x = a''_x + \sum_{y \in \{f^{-1}(x)\}} (2^b - a''_y)$. Thus

$$O|x\rangle|0\rangle|0\rangle = |x\rangle|a''_x\rangle|f(x)\rangle, \quad (17)$$

where we have omitted the irrelevant garbage state. Third, the oracle O is applied to a uniform superposition over $|x\rangle$, and the comparator trick of Eq. (16) is applied. This produces the state

$$\sum_{x=0}^{N-1} \frac{|x\rangle}{\sqrt{N}} |a''_x\rangle|f(x)\rangle \left(\sqrt{\frac{a''_x}{2^b}} |0\rangle|u_{a''_x}\rangle + \cdots |1\rangle|u_{\geq a''_x}\rangle \right). \quad (18)$$

Finally, $|f(x)\rangle$ is swapped with $|x\rangle$, controlled on the $|1\rangle$ state. This leads to a state $|\psi\rangle = \sum_{x=0}^{N-1} \sqrt{\frac{a'_x}{\|\vec{a}'\|_1}} |x\rangle|\text{garbage}_x\rangle$. After tracing out the garbage register, the resulting density matrix ρ' approximates the desired state ρ with trace distance

$$\|\rho' - \rho\|_1 = \mathcal{O}(2^{-b}) \leq \epsilon. \quad (19)$$

The T gate complexity is then the cost of the data-lookup oracle of Eq. (17) plus $\mathcal{O}(b)$ for the comparator of Eq. (16), plus $\mathcal{O}(\log N)$ for the controlled swap with $|f(x)\rangle$. By implementing this data-lookup oracle with the SELECTSWAP network, one immediately obtains the stated T gate complexity of $\mathcal{O}(\lambda(b + \log N) + \frac{N}{\lambda}) = \mathcal{O}(\lambda \log(N/\epsilon) + \frac{N}{\lambda})$, where we choose $b = \mathcal{O}(\log(1/\epsilon))$.

B Data-lookup oracle implementation details

In this section, we present additional details on the implementation of the data-lookup oracle. In particular, we discuss a multi-target CNOT implementation in logarithmic depth without ancillary qubits in Appendix B.1, and a swap network SWAP with similar properties in Appendix B.2. Also evaluated is the T count and Clifford depth of these implementations up to constant factors. We define the Clifford depth, to be the number of layers of two-qubit Clifford gates that cannot be executed in parallel, assuming all-to-all qubit connectivity. We also assume that each T magic-state injection circuit has a Clifford depth of 1.

B.1 Quantum fanout in logarithmic depth without ancillary qubits

In this section, we construct a controlled-NOT gate that targets n qubits, that is,

$$\text{CNOT}_n = |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes X^{\otimes n}. \quad (20)$$

The most straightforward approach applies n NOT gates in sequence, each controlled by the same qubit. A slight modification results in logarithmic depth as shown in Table 3.

Given any number of n qubits in states $|x_j\rangle$ for $j = 0, 1, \dots, n-1$, one may use a ladder of $n-1$ controlled-NOT gates to realize the transformation

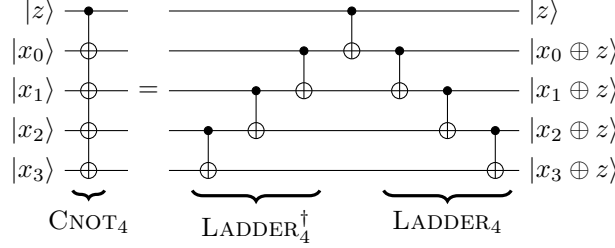
$$\begin{aligned} & |x_0\rangle|x_1\rangle|x_2\rangle \cdots |x_{n-1}\rangle \\ \rightarrow & |x_0\rangle|x_0 \oplus x_1\rangle|x_2\rangle \cdots |x_{n-1}\rangle \\ \rightarrow & |x_0\rangle|x_0 \oplus x_1\rangle|x_0 \oplus x_1 \oplus x_2\rangle \cdots |x_{n-1}\rangle \\ \rightarrow & |x_0\rangle|x_0 \oplus x_1\rangle|x_0 \oplus x_1 \oplus x_2\rangle \cdots |\oplus_{j=0}^{n-1} x_j\rangle. \end{aligned} \quad (21)$$

Let us call this unitary operation LADDER_n .

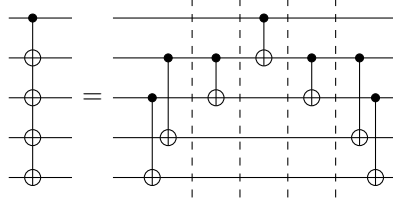
We now introduce a control qubit $|z\rangle$. One implementation of CNOT_n is then obtained by applying LADDER_n^\dagger , followed by a NOT on $|x_0\rangle$ controlled by $|z\rangle$, and finally followed by LADDER_n . This has a Clifford depth of $2n-1$ as depicted below for the example $n=4$.

Approach	Clifford Depth (d)	Clifford Count	Volume $= (n+1)d$
Linear	n	n	$(n+1)d$
Logarithmic Eq. (22)	$2\lceil \log_2(n) \rceil + 1$ $\leq 2\lceil \log_2(n/2 + 1) \rceil$	$2n - 1$ $\leq 2n - 1$	$(n+1)d$ $(n+1)d$

Table 3: Different implementations of a controlled-NOT gate Eq. (20) that targets n qubits.



By distributing the controls and targets above in a tree-structure as depicted below for the example $n = 4$, the Clifford depth of CNOT_n may be reduced to $2\lceil \log_2(n) \rceil + 1$.



As the control qubit $|z\rangle$ is only used once in the above circuit, a further reduction in depth is possible by repeatedly using it to apply additional multi-target Cnot gates in each time-slice. Let us denote by $g(d) = 2^{(d-1)/2}$ the maximum number of qubits targeted by the above circuit in a depth of d . Then the total number of qubits $n(d)$ targeted with this additional reduction satisfies the recurrence

$$n(1) = g(1), \quad (22)$$

$$n(2) = n(1) + g(1), \quad (23)$$

$$\vdots \quad (24)$$

$$n(d) = n(d-1) + g(2\lceil d/2 \rceil - 1) = \begin{cases} (3 \cdot 2^{(d-1)/2} - 2), & d \text{ odd,} \\ 2(2^{d/2} - 1), & d \text{ even.} \end{cases} \geq 2(2^{d/2} - 1)$$

Let us denote by $D[\text{CNOT}_n]$ the depth of this implementation of CNOT_n , which satisfies

$$D(\text{CNOT}_n) \leq \left\lceil 2 \log_2 \left(\frac{n+2}{2} \right) \right\rceil. \quad (25)$$

B.2 Implementations of a SWAP network

In this section, we detail various implementations of the unitary swap network SWAP that moves an b -qubit quantum register indexed by $x \in \{0, \dots, N-1\}$ to the position of the $x = 0$ register, controlled by an index state $|x\rangle$. More precisely, for any set of quantum states $\bigotimes_{x=0}^{N-1} |\phi_x\rangle_x$ in the b -qubit register indexed by x ,

$$\text{SWAP} \left[|x\rangle \bigotimes_{x=0}^{N-1} |\phi_x\rangle_x \right] = |x\rangle |\phi_x\rangle_0 \cdots, \quad (26)$$

Approach	Clifford Depth (d_c)	T count	T depth (d_t)	Volume = $(2n + 1)(d_c + d_t)$
Linear	$4n + 4$	$7n$	4	$8n^2 + \mathcal{O}(n)$
Logarithmic	$4D[\text{CNOT}_{\lceil n/2 \rceil}] + 2$	$14n$	8	$\leq 8n(\lceil 2 \log_2(\frac{n+2}{2}) \rceil + 2.5) + \mathcal{O}(\log n)$
Phase incorrect	$D[\text{CNOT}_n] + 4$	$4n$	4	$\leq 2n(\lceil 2 \log_2(\frac{n+2}{2}) \rceil + 8) + \mathcal{O}(\log n)$

Table 4: Different implementations of a controlled-swap between two n -qubit registers. The depth $D[\text{CNOT}_n] \leq \lceil 2 \log_2(\frac{n+2}{2}) \rceil$ is from Eq. (25).

where the final quantum states in registers indexed by $x > 0$ are unimportant. Let us express the index $x \equiv x_0 x_1 \dots$ in binary, where x_0 is the smallest bit. Then it suffices to perform swaps between all pairs of registers indexed by $\{(i, i + 2^j) \mid i \in 2^{j+1}\mathbb{N}_0\}$, controlled by the j^{th} qubit $|x_j\rangle$ of the index state $|x\rangle$, in the order of $j = 0, 1, \dots, \lceil \log_2 N \rceil - 1$.

Each controlled pair-wise swap may be understood as a circuit CSWAP_n that swaps two n -qubit quantum registers in any state $|\psi\rangle, |\phi\rangle \in \mathbb{C}^{2^n}$, controlled by a single qubit $|z\rangle \in \mathbb{C}^2$. That is,

$$\text{CSWAP}_n |z\rangle |\psi\rangle |\phi\rangle = \begin{cases} |z\rangle |\psi\rangle |\phi\rangle, & z = 0, \\ |z\rangle |\phi\rangle |\psi\rangle, & z = 1. \end{cases} \quad (27)$$

The overall cost of SWAP is then the sum of costs of CSWAP_n , over all j , where $n = \lfloor \frac{N-1}{2^{j+1}} + \frac{1}{2} \rfloor$.

We now consider different implementations considered realize trade-offs between Toffoli-gate count, circuit depth, and ancillary qubit usage, as summarized in Table 4.

B.2.1 CSWAP_n in linear depth without ancillary qubits

It is simple to construct CSWAP_n with depth $\mathcal{O}(n)$ without any ancillary qubits. As the circuit that swaps two qubits is constructed from three CNOT gates as follows,

$$\begin{array}{c} |\psi\rangle \\ |\phi\rangle \end{array} \begin{array}{c} \times \\ \times \end{array} = \begin{array}{c} \oplus \\ \bullet \\ \oplus \end{array} \begin{array}{c} \bullet \\ \oplus \\ \bullet \end{array} \begin{array}{c} \oplus \\ \bullet \\ \oplus \end{array} \begin{array}{c} |\phi\rangle \\ |\psi\rangle \end{array}$$

a controlled-swap below is obtained by replacing the middle CNOT with a Toffoli gate.

$$\begin{array}{c} |z\rangle \\ \times \\ \times \end{array} = \begin{array}{c} \bullet \\ \oplus \\ \bullet \\ \oplus \end{array} \begin{array}{c} \bullet \\ \oplus \\ \bullet \end{array} \begin{array}{c} |z\rangle \\ \times \\ \times \end{array}$$

A circuit that implements Eq. (27), a swap between n pairs of qubits, is then the above repeated n times in sequence, each controlled by the same qubit as follows.

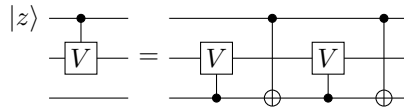
$$\begin{array}{c} |z\rangle \\ |\psi\rangle_0 \\ |\phi\rangle_0 \\ \vdots \\ |\psi\rangle_{n-1} \\ |\phi\rangle_{n-1} \end{array} \begin{array}{c} \bullet \\ \times \\ \times \\ \vdots \\ \times \\ \times \end{array} \begin{array}{c} \bullet \\ \oplus \\ \bullet \\ \oplus \\ \bullet \\ \oplus \end{array} \begin{array}{c} |z\rangle \\ \times \\ \times \\ \vdots \\ \times \\ \times \end{array}$$

B.2.2 CSWAP_n in logarithmic depth without ancillary qubits

Constructing CSWAP_n with depth $\mathcal{O}(\log n)$ without any ancillary qubits requires a little more thought. Let us consider a more general problem. Suppose we have an arbitrary unitary operator V that is self-inverse, meaning $V^2 = \mathbb{I}$ – one may verify that the two-qubit swap satisfies this property. Our goal is to implement a multi-target controlled- V gate on n registers

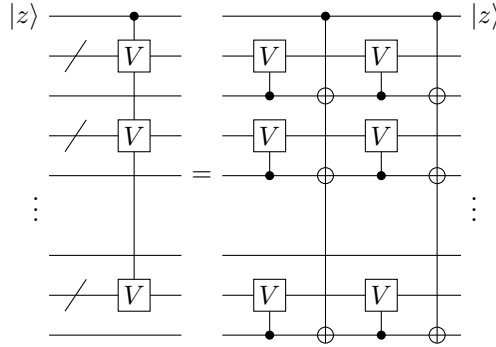
$$|0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes V^{\otimes n}. \quad (28)$$

To begin, consider the following circuit identity, which is motivated by the ‘toggling’ trick in [42].



Observe the the bottom qubit may be dirty – its state does not affect the computation, and remains unchanged at the end of it.

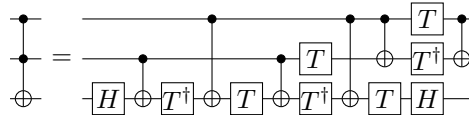
Thus a multi-target controlled- V on n registers may be constructed by applying n singly-controlled V gates in parallel before and after a single multiply-controlled NOT gate, using a total of n extra dirty qubits as follows.



As the additional qubits may be dirty, this is easily modified to use no ancillary qubits at all. Let us apply the multi-target V on $\lceil n/2 \rceil$ registers, using any $\lfloor n/2 \rfloor$ qubits from the other registers as dirty qubits. When n is odd, the topmost V may be controlled directly by the $|z\rangle$ qubit. We then apply the same circuit on the remaining registers by using qubits in the initial targets as control qubits. In total, this uses at most $2n$ controlled- V gates and two multiply-controlled NOT gates on $\lceil n/2 \rceil$ qubits, each with cost given by [Table 3](#).

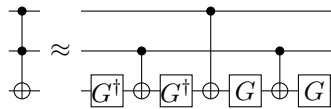
B.2.3 T-gate decomposition

Each controlled-swap may be decomposed into Clifford+T gates using standard techniques. For instance, the standard synthesis of each Toffoli uses 7 T -gates [\[12\]](#), as seen below.

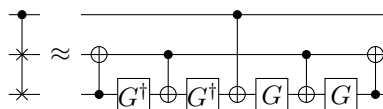


Thus one might expect that CSWAP_n in logarithmic depth requires $14N$ T -gates. However, simple cancellations using the above decomposition reduces this to $10N$ T -gates.

A further reduction to just $4N$ T -gates is possible if we allow the output state to be correct up to a phase factor. The decomposition by [\[19\]](#) below, using the gate $G = S^\dagger \cdot H \cdot T \cdot H \cdot S$, approximates the Toffoli gate up to a minus sign on one of the matrix elements.



Thus a controlled-swap is obtained by a simple modification as follows.



Using this approximate controlled-swap, a version of CSWAP_n that is correct up to a ± 1 phase may be obtained by replacing the middle CNOT with CNOT_n , which may be implemented with logarithmic depth. As this incorrect phase may be absorbed into the garbage state of the data-lookup oracle, we may apply this phase-incorrect swap operation without loss of generality.

C Indicator Function Construction

There is an alternative construction based on implementing an *indicator function*. That is, the function

$$e^{(n)}: \{0, 1\}^n \rightarrow \{0, 1\}^N, \quad (29)$$

where $N = 2^n$, which maps a string $x \in \{0, 1\}^n$ to an exponentially long encoding which is 1 at position x and zeros everywhere else.

We observe that $e^{(n)}$ can be implemented with the following parameters.

Theorem 1. *For $n \geq 1$, let U_n be the unitary for the n th indicator function, $e^{(n)}$. That is, the circuit mapping*

$$|x\rangle|y\rangle \mapsto |x\rangle|e^{(n)}(x) \oplus y\rangle \quad (30)$$

for all $x \in \{0, 1\}^n$ and $y \in \{0, 1\}^N$. There is a circuit computing U_n in depth $O(\log^2 N)$ with $O(N)$ T gates without any ancillary qubits.

Proof. When $n = 1, 2$ there are trivial circuits without ancillary qubits. For $n > 2$, let us suppose we have arbitrarily many clean ancillary qubits. We divide the input bits x into two halves, x_{hi} and x_{lo} , then recursively compute $e^{(n/2)}(x_{hi})$ and $e^{(n/2)}(x_{lo})$ in $O(2^{n/2}) = O(\sqrt{N})$ clean ancillary qubits each.

Each output of $e^{(n)}(x)$ is the AND of a bit in $e^{(n/2)}(x_{lo})$ and a bit in $e^{(n/2)}(x_{hi})$. Each bit is used $O(\sqrt{N})$ times, so we can finish XORing $e^{(n)}(x)$ into y with an array of Toffoli gates of $O(\sqrt{N})$. Alternatively, we can make $O(\sqrt{N})$ copies of each vector and do all the Toffoli gates simultaneously.

Now suppose the ancillary qubits are dirty. Dirty qubits cannot store a qubit in the same way as clean qubits; since the initial state of the qubit is arbitrary, the information is encoded in the *change* in state rather than the actual state. To implement a CNOT controlled on a dirty qubit, for instance, we apply the CNOT first, flip or not flip the qubit, then apply another CNOT. If there was no change, the two gates cancel, otherwise exactly one fires.

Recall that the Toffoli gate has an implementation up to a faulty sign (which we can tolerate) as three CNOT interleaved with G or G^\dagger . As described above, we can implement each CNOT with a pair of CNOTs and, more importantly, a recursive call to the subroutine which populates that qubit, i.e., an indicator function. Actually, we use the previously described circuit for $\text{CNOT}_{O(\sqrt{N})}$ instead of many individual CNOT gates for each Toffoli involving that bit.

The depth cost for $\text{CNOT}_{O(\sqrt{N})}$ is $O(\log(\sqrt{N})) = O(n)$. Additionally, to compute U_n we need 4 recursive calls to $U_{n/2}$. The depth satisfies the recurrence $D(n) = 4D(n/2) + O(n)$, so $D(n) = O(n^2) = O(\log^2 N)$.

Finally, suppose we apply the array of Toffoli gates in *two* layers instead of just one simultaneous layer. Since we are only applying Toffolis to half of the $|y\rangle$ register at any time, the other half can be used as dirty ancillary qubits. Asymptotically, this is much more than the $O(\sqrt{N})$ dirty qubits we need to compute $U_{n/2}$. Hence, we can implement U_n without any additional qubits. \square

It is wasteful to compute all of $e^{(n)}(x)$ in ancillary qubits, but it can be used to compute an arbitrary function. Better is to divide the input into two pieces, similar to how we compute the indicator function.

Theorem 2. *Suppose $f: \{0, 1\}^n \rightarrow \{0, 1\}^b$ is an arbitrary function, and let U be the unitary mapping*

$$|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle, \quad (31)$$

for all $x \in \{0, 1\}^n$ and $y \in \{0, 1\}^b$.

There is a circuit for U of depth $O(\log^2 N + \log b)$, using $O(bN)$ dirty ancillary qubits and $O(\sqrt{bN})$ T gates. Alternatively, there is a circuit for U using $O(\sqrt{bN})$ T gates and only $O(\lambda\sqrt{bN})$ dirty ancillary qubits, but with depth $O(\sqrt{bN}/\lambda + \log^2 N)$, for any $1 \leq \lambda \leq \sqrt{bN}$.

Proof. Divide the input into $n - k$ and k bit pieces, where k is to be determined later. Let $\hat{f}: \{0, 1\}^{n-k} \rightarrow \{0, 1\}^{b \times 2^k}$ be

$$\hat{f}(x_{hi}) = (f(x_{hi}x_{lo}))_{x_{lo} \in \{0, 1\}^k}, \quad (32)$$

the function which outputs the function for all possible values of the low order bits, given the high order bits.

Suppose, for the moment, that we have as many clean ancillary qubits as we want. We can naively compute $\hat{f}(x_{hi})$ by constructing $e^{(n-k)}(x_{hi})$, making $b \cdot 2^k$ copies, and computing the parity of some subset of bits of $e^{(n-k)}(x_{hi})$ for each output bit of $\hat{f}(x_{hi})$. Constructing $e^{(n-k)}(x_{hi})$ requires $O(2^{n-k})$ ancillary qubits, $O(2^{n-k})$ T gates and $O((n-k)^2)$ depth. The rest is done with $\text{CNOT}_{b \cdot 2^k}$ gates to make copies and $\text{CNOT}_{O(2^{n-k})}$ gates conjugated by Hadamards to compute parities. This uses many ancillary qubits— $O(2^{n-k})$ for the original vector times $O(b2^k)$ copies is $O(bN)$ qubits.

Since the layers of CNOT gates compute (in the output bits) a linear function of the inputs, it is not difficult to adapt for dirty ancillary qubits. Just apply the linear function, flip the dirty ancillary qubits that are set to 1, then apply the linear function again, appealing to the identity $T(x \oplus y) \oplus T(y) = T(x)$ for a linear function. Thus, whatever state $|y\rangle$ was in the dirty qubits, we still manage to compute $T(x)$.

We have shown how to compute $\hat{f}(x_{hi})$ and XOR it into dirty ancillary qubits. We also know how to compute $e^k(x_{lo})$ and XOR b copies of it into dirty ancillary qubits. Think of $\hat{f}(x_{hi})$ as a $b \times 2^k$ matrix, then all that remains is to return the correct column of $\hat{f}(x_{hi})$. If we also think of $e^k(x_{lo})$ as a length 2^k column vector, then we are computing a matrix/vector product. The simplest way to do this is to make b copies of $e^k(x_{lo})$ and execute b vector/vector products in parallel, at a cost of $O(2^k)$ Toffoli gates for each one.

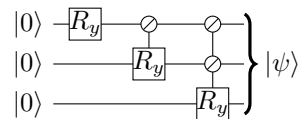
To compute the Toffoli gates on dirty ancillary qubits, we decompose them into CNOT gates and single qubit gates. The layers of CNOT gates are linear, so it is possible to compute each such layer with dirty ancillary qubits.

Computing $\hat{f}(x_{hi})$ uses $O(bN)$ ancillary qubits, $O(2^{n-k})$ T gates, and has depth $O((n-k)^2 + k + \log b)$. The rest of the circuit has $O(b2^k)$ ancillas to store $\hat{f}(x_{hi})$ and/or copies of $e^k(x_{lo})$, and uses $O(b2^k)$ T gates in depth $O(k^2 + \log b)$. Since the T gate count is $O(2^{n-k} + b2^k)$, we set k such that $2^k \approx \sqrt{N/b}$, and it becomes $O(\sqrt{bN})$.

It is possible to trade off depth and number of ancillary qubits. We only need $O(2^{n-k})$ qubits to store $e^{(n-k)}(x_{hi})$ in the computation of $\hat{f}(x_{hi})$, if we are willing to compute parities for each of the $b2^k$ output bits of $\hat{f}(x_{hi})$ one at a time, in depth $O(b2^k)$. More generally, we can use $O(\lambda 2^{n-k})$ ancillary qubits for any integer $\lambda \in [1, b2^k]$ and use depth $O(b2^k/\lambda)$. For the optimal T count we use the same setting of k , giving $O(\sqrt{bN}/\lambda + \log^2 N)$ depth with $O(\lambda\sqrt{bN})$ ancillary qubits and $O(\sqrt{bN})$ T gates. □

D Pure-state preparation implementation details

The approach by Shende, Bullock, and Markov [12] synthesizes a unitary A that prepares a pure state $A|0\rangle = \sum_{x=0}^N \frac{a_x}{\|a\|_2} |x\rangle = |\psi\rangle$ with arbitrary coefficients in $N = 2^n$ dimensions. The underlying circuit, illustrated below for the example of $N = 8$ for positive coefficients,



is built from $j \in [n]$ multiply-controlled arbitrary single qubit rotations, U_j where

$$U_j = \sum_{x=0}^{2^j-1} |x\rangle\langle x| \otimes e^{i2\pi\theta_{j,x}Z}, \quad (33)$$

for some set of rotation angles $\theta_{j,x}$. Note that it suffices to consider Z -phase rotations as rotations about the X, Y Pauli operators are equivalent up to a single-qubit Clifford similarity transformation. Each multiplexor is applied twice – once to create a pure state with the right probabilities $|a_x|^2$, and once to apply the correct phase $e^{i\arg[a_x]}$. Below we describe how U_j may be implemented using the data-lookup oracle of Eq. (5)

$$O|x\rangle|0\rangle|0\rangle = |x\rangle|a_x\rangle|\text{garbage}_x\rangle, \quad (34)$$

and evaluate the overall error and cost of state preparation.

D.1 Multiply-controlled phase gate from data lookup oracles

Consider a multiply- n -controlled arbitrary single qubit rotation

$$U = \sum_{x=0}^{2^n-1} |x\rangle\langle x| \otimes e^{i2\pi\theta_x Z}, \quad (35)$$

$$\begin{array}{c} |x\rangle \text{---} \text{---} \text{---} |x\rangle \\ |z\rangle \text{---} \boxed{R_z} \text{---} e^{i2\pi\theta_x Z} |z\rangle \end{array}$$

where each rotation angle $\theta_x \in [0, 1)$. Given a number state $|x\rangle$ and an arbitrary single-qubit state $|z\rangle$, this unitary performs a controlled-rotation

$$U|x\rangle|z\rangle = |x\rangle e^{i2\pi\theta_x Z}|z\rangle. \quad (36)$$

Each rotation angle has a binary expansion

$$\theta_x = \sum_{k=0}^{\infty} \frac{a_{x,i}}{2^{k+1}}. \quad (37)$$

By truncating to the above to b -bits of precision, we obtain an integer approximation a_x of θ_x where

$$a_x = 2^b \sum_{k=0}^{b-1} \frac{a_{x,k}}{2^{k+1}}, \quad \left| \frac{a_x}{2^b} - \theta_x \right| < \frac{1}{2^b}. \quad (38)$$

Let us encode these values of a_x into the data-lookup oracle, and express its T cost as the function $f(n, b)$. Its output is then

$$O|x\rangle|0\rangle^{\otimes b}|0\rangle = |x\rangle \left[\bigotimes_{k=0}^{b-1} |a_{x,k}\rangle \right] |\text{garbage}_x\rangle, \quad (39)$$

where we explicitly represent the number state $|a_x\rangle$ in terms of its component qubits.

D.1.1 Approach using arbitrary single-qubit synthesis

One possible approximation, call it U' , of U in Eq. (35) applies the single-qubit rotation $e^{i\pi Z/2^k}$ to the target state $|z\rangle$, controlled by the state $|a_{x,k}\rangle$. The garbage register is then uncomputed by

running O in reverse. Explicitly, this circuit realizes the transformation

$$\begin{aligned}
& |x\rangle|0\rangle^{\otimes b}|0\rangle|z\rangle \\
& \rightarrow |x\rangle \left[\bigotimes_{k=0}^{b-1} |a_{x,k}\rangle \right] |\text{garbage}_x\rangle|z\rangle \\
& \rightarrow |x\rangle \left[\bigotimes_{k=0}^{b-1} |a_{x,k}\rangle \right] |\text{garbage}_x\rangle e^{i\pi \sum_{k=0}^{b-1} a_{x,k} Z/2^k} |z\rangle \\
& = |x\rangle|a_x\rangle|\text{garbage}_x\rangle e^{i2\pi a_x/2^b Z} |z\rangle \\
& \rightarrow |x\rangle|0\rangle^{\otimes b}|0\rangle e^{i2\pi a_x/2^b Z} |z\rangle.
\end{aligned} \tag{40}$$

Now, each controlled-arbitrary phase rotation decomposes into 2 arbitrary single-qubit rotations, and CNOT gates – Note that a decomposition into 1 arbitrary single-qubit rotation is possible if we modify the above for the range $\theta_x \in (-1/2, 1/2)$, but the explanation is slightly more complicated. As any arbitrary single-qubit rotation is approximated to error ϵ using $\mathcal{O}(\log(1/\epsilon))$ T gates [43], we may use a triangle inequality to bound the error to

$$\begin{aligned}
\|U - U'\| & \leq \left\| \sum_{x=0}^{2^n-1} |x\rangle\langle x| \otimes (e^{i2\pi\theta_x Z} - e^{i2\pi a_x/2^b Z}) \right\| + 2b\epsilon \\
& \leq \max_{|y| < 1/2^b} |e^{i2\pi y} - 1| + 2b\epsilon \\
& < \frac{2\pi}{2^b} + 2b\epsilon.
\end{aligned} \tag{41}$$

Thus for any target error $\|U - U'\| = \delta$, we may solve for the b, ϵ parameters and bound the number of T gates required to

$$f(n, b) + \mathcal{O}(\log^2(1/\delta)), \quad b = \mathcal{O}(\log(1/\delta)). \tag{42}$$

D.1.2 Approach using phase gradients

A more efficient approach [22] uses a Fourier state as a resource

$$\mathcal{F} = \frac{1}{\sqrt{2^b}} \sum_{k=0}^{2^b-1} e^{-2\pi i k/2^b} |k\rangle. \tag{43}$$

combined with a reversible adder

$$\text{ADD}|x\rangle|y\rangle = |x\rangle|y+x \pmod{2^b}\rangle \tag{44}$$

Observe that

$$\text{ADD}|x\rangle|\mathcal{F}\rangle = e^{2\pi i x/2^b} |x\rangle|\mathcal{F}\rangle. \tag{45}$$

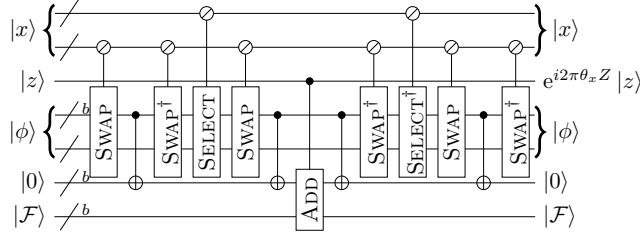
Thus the controlled adder, known to cost $\mathcal{O}(b)$ T gates,

$$\text{CADD} = \text{ADD} \otimes |0\rangle\langle 0| + \text{ADD}^\dagger \otimes |1\rangle\langle 1| \tag{46}$$

realizes the controlled-phase rotation

$$\text{CADD}|x\rangle|\mathcal{F}\rangle|z\rangle = |x\rangle|\mathcal{F}\rangle e^{2\pi i x/2^b Z} |z\rangle. \tag{47}$$

Thus another possible approximation, call it U'' , uses this adder, controlled by the target state $|z\rangle$, to the registers containing the desired phase rotation $|a_x\rangle$, and the Fourier state. This realizes the same transformation in Eq. (40), using the circuit depicted below.



Assuming that the Fourier state is prepared perfectly, this approximates U with error

$$\|U - U''\| \leq \left\| \sum_{x=0}^{2^n-1} |x\rangle\langle x| \otimes (e^{i2\pi\theta_x Z} - e^{i2\pi a_x/2^b Z}) \right\| \leq \max_{|y| < 1/2^b} |(e^{i2\pi y} - 1)| < \frac{2\pi}{2^b}. \quad (48)$$

The state preparation unitary A'' applies n such approximations U_j'' to the multiply-controlled rotations U_j , leading to an overall error bounded by

$$\left\| \prod_{j=0}^{n-1} U_j - \prod_{j=0}^{n-1} U_j'' \right\| \leq \sum_{j=0}^{n-1} \|U_j - U_j''\| < \frac{2n\pi}{2^b}. \quad (49)$$

Note that the cost of approximating the Fourier state to error ϵ has a T cost of $\mathcal{O}(b \log(1/\epsilon))$. This imperfect Fourier state $|\mathcal{F}'\rangle$ contributes a one-time error of ϵ , following the inequality

$$\|W(|\mathcal{F}'\rangle \otimes \mathbb{I}) - (W|\mathcal{F}\rangle \otimes \mathbb{I})\| \leq \| |\mathcal{F}'\rangle - |\mathcal{F}\rangle \| \leq \epsilon, \quad (50)$$

for any arbitrary unitary operator W . Thus A'' prepares the state $|\psi''\rangle = A''|0\rangle|\mathcal{F}'\rangle$, which approximates $|\psi\rangle$ with error

$$\delta = \| |\psi''\rangle - |\psi\rangle \| \leq \frac{2\pi n}{2^b} + \epsilon. \quad (51)$$

The total error δ may be controlled by choosing $b = \Theta(\log(\frac{n}{\delta}))$ and $\epsilon = \Theta(\delta)$.

The total T cost of arbitrary state preparation is then the sum of costs of the data-lookup oracles $f(j, b) = \mathcal{O}(\frac{2^j}{\lambda} + b\lambda)$, adders, and the Fourier state:

$$\begin{aligned} \text{T cost} &= \left(\sum_{j=0}^{n-1} f(j, b) + \mathcal{O}(b) \right) + \mathcal{O}(b \log(1/\epsilon)) \\ &= \mathcal{O}(2^n/\lambda + nb\lambda + nb + b \log(1/\epsilon)) \\ &= \mathcal{O}\left(\frac{N}{\lambda} + b(\lambda n + \log(1/\delta))\right) \\ &= \mathcal{O}\left(\frac{N}{\lambda} + \log\left(\frac{\log N}{\delta}\right) (\lambda \log N + \log(1/\delta))\right) \\ &= \mathcal{O}\left(\frac{N}{\lambda} + \lambda \log\left(\frac{\log N}{\delta}\right) \log\left(\frac{N}{\delta}\right)\right) \\ &= \mathcal{O}\left(\frac{N}{\lambda} + \lambda \log^2\left(\frac{N}{\delta}\right)\right). \end{aligned} \quad (52)$$