Asymptotically Optimal Circuit Depth for Quantum State Preparation and General Unitary Synthesis

Xiaoming Sun*[1,2], Guojing Tian*[1,2], Shuai Yang*[1,2], Pei Yuan†[3] and Shengyu Zhang †[3]

[1]*State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*

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

[3]*Tencent Quantum Laboratory, Tencent, Shenzhen, Guangdong 518057, China*

Abstract

The Quantum State Preparation problem aims to prepare an n-qubit quantum state $|\psi_v\rangle = \sum_{k=0}^{2^n-1} v_k|k\rangle$ from the initial state $|0\rangle^{\otimes n}$, for a given unit vector $v = (v_0, v_1, v_2, \ldots, v_{2^n-1})^T \in \mathbb{C}^{2^n}$ with $\|v\|_2 = 1$. The problem is of fundamental importance in quantum algorithm design, Hamiltonian simulation and quantum machine learning, yet its circuit depth complexity remains open when ancillary qubits are available. In this paper, we study quantum circuits when there are m ancillary qubits available. We construct, for any m, circuits that can prepare $|\psi_v\rangle$ in depth $\tilde{O}\left(\frac{2^n}{m+n} + n\right)$ and size $O(2^n)$, achieving the optimal value for both measures simultaneously. These results also imply a depth complexity of $\Theta\left(\frac{4^n}{m+n}\right)$ for quantum circuits implementing a general n-qubit unitary for any $m \leq O(2^n/n)$ number of ancillary qubits. This resolves the depth complexity for circuits without ancillary qubits. And for circuits with exponentially many ancillary qubits, our result quadratically improves the currently best upper bound of $O(4^n)$ to $\tilde{\Theta}(2^n)$.

Our circuits are deterministic, prepare the state and carry out the unitary precisely, utilize the ancillary qubits tightly and the depths are optimal in a wide parameter regime. The results can be viewed as (optimal) time-space trade-off bounds, which are not only theoretically interesting, but also practically relevant in the current trend that the number of qubits starts to take off, by showing a way to use a large number of qubits to compensate the short qubit lifetime.

1 Introduction

Quantum computers provide a great potential of solving certain important information processing tasks that are believed to be intractable for classical computers. In recent years, quantum machine learning [1] and Hamiltonian simulation [2–5] have also been extensively investigated, including quantum principal component analysis (QPCA) [6], quantum recommendation systems [7], quantum singular value decomposition [8], quantum linear system algorithm [9, 10], quantum clustering [11, 12] and quantum support vector machine (QSVM) [13]. One of the challenges to fully exploit quantum algorithms for these tasks, however, is to efficiently prepare a starting state[1], which is usually the first step of those algorithms. This raises the fundamental question about the complexity of the quantum state preparation (QSP) problem.

The QSP problem can be formulated as follows. Suppose we have a vector $v = (v_0, v_1, v_2, \ldots, v_{2^n-1})^T \in \mathbb{C}^{2^n}$ with unit ℓ_2-norm, i.e. $\sqrt{\sum_{k=0}^{2^n-1} |v_k|^2} = 1$. The task is to generate a corresponding n-qubit quantum

*Email:{sunxiaoming, tianguojing, yangshuai21b}@ict.ac.cn

†Email: {peiyuan, shengyzhang}@tencent.com

[1] These starting states (for example, those in [9, 10]) are very generic. Indeed, the lower bound argument in our later Theorem 3 applies to the generation of these states as well.

state

$$|\psi_v\rangle = \sum_{k=0}^{2^n-1} v_k|k\rangle,$$

by a quantum circuit from the initial state $|0\rangle^{\otimes n}$, where $\{|k\rangle : k = 0, 1, \ldots, 2^n - 1\}$ is the computational basis of the quantum system.

Different cost measures can be studied for quantum circuits: Size, depth, and number of qubits are among the most prominent ones. For a quantum circuit, the depth corresponds to the time for executing the quantum circuit, and the number of qubits used to its space cost. Apart from minimizing each cost measure individually, it is of particular interest to study a time-space trade-off for quantum circuits. The reason is that in the past decade, we have witnessed a rapid development in qubit number and in qubit lifetime[2], but it seems hard to significantly improve *both* on the same chip. Looking into the near future, big players such as IBM and Google announced their roadmaps of designing and manufacturing quantum chips with about 1,000,000 superconducting qubits by 2026 and 2029, respectively, rocketing from 50-100 today [20, 21]. This raises a natural question for quantum algorithm design: How to utilize the fast-growing number of qubits to overcome the relatively limited decoherence time? This seems especially relevant in the near future when we have $10^4 - 10^5$ qubits, which are expected to run certain quantum simulation algorithms for chemistry problems, but are not sufficient for the full quantum error correction to fight the decoherence. Or put in a computational complexity language, how to efficiently trade space for time in a quantum circuit? In this paper, we will address this question in the fundamental tasks of quantum state preparation and general unitary circuit synthesis.

Let us first fix a proper circuit model. If we aim to generate the target state $|\psi_v\rangle$ or perform the target unitary precisely, then a finite universal gate set is not enough. A natural choice is the set of circuits that consist of arbitrary single-qubit gates and CNOT gates, which is expressive enough to generate arbitrary states $|\psi_v\rangle$ precisely with certainty. We will study the optimal depth for this class of circuits[3].

The study of QSP dates back to 2002, when Grover and Rudolph gave an algorithm for QSP for the special case of efficiently integrable probability density functions [22]. Their circuit has n stages, and each stage j has 2^{j-1} layers, with each layer being a rotation on last qubit conditioned on the first $j-1$ qubits being certain computational basis state. This type of multiple-controlled (2×2)-unitary can be implemented in depth $O(n)$ without ancillary qubit[4], yielding a depth upper bound of $O(n2^n)$ for the QSP problem. In [25], Bergholm *et al.* gave an upper bound of $2^{n+1} - 2n - 2$ for the *number* of CNOT gates, with depth also of order $O(2^n)$. The number of CNOT gates is improved to $\frac{23}{24}2^n - 2^{\frac{n}{2}+1} + \frac{5}{3}$ for even n, and $\frac{115}{96}2^n$ for odd n by Plesch and Brukner [26], based on a universal gate decomposition technique in [27]. The same paper [25] also gives a depth upper bound of $\frac{23}{48}2^n$ for even n and $\frac{115}{192}2^n$ for odd n. All these results are about the exact quantum state preparation without ancillary qubits.

With ancillary qubits, Zhang *et al.* [28] proposed circuits which involve measurements and can generate the target state in $O(n^2)$ depth but only with certain success probability, which is at least $\Omega(1/(\max_i |v_i|^2 2^n))$, but in the worst case can be an exponentially small order of $O(1/2^n)$. In addition, they need $O(4^n)$ ancillary qubits to achieve this depth. In a different paper [29], the authors showed that for $\epsilon \leq 2^{-\Omega(n)}$, an n-qubit quantum state $|\psi_v'\rangle$ can be implemented by an $O(n^3)$-depth quantum circuit with sufficiently many ancillary qubits[5], where $\||\psi_v'\rangle - |\psi_v\rangle\| \leq \epsilon$. Though QSP is only used as a tool for their main topic of parallel quantum walk, their concluding section did call for studies on the trade-off between the circuit depth and the number of ancillary qubits for better parallel quantum algorithms. Another related study is [30], which considers to prepare a state not in the binary encoding $\sum_{k=0}^{2^n-1} v_k|k\rangle$,

[2] Take superconducting qubits, for example, the qubit number jumped from 5 in 2014 to 127 in 2021 [14–19] .

[3] Since two-qubit gates are usually harder to implement, one may also like to consider CNOT depth, the number of layers with at least one CNOT gate. But note that between two CNOT layers, consecutive single-qubit gates on the same qubit can be compressed to one single-qubit gate, and single-qubit gates on different qubits can be paralleled to within one layer, we can always assume that the circuit has alternative single-qubit gate layers and CNOT gate layers. Therefore the circuit depth is at most twice of the CNOT depth, making the two measures the same up to a factor of 2.

[4] The standard method [23] gives a depth upper bound of $O(n^2)$ without ancillary qubit and $O(n)$ with sufficiently many ancillary qubits. The first bound can be improved to $O(n)$ by the method in [24].

[5] No explicit bound on the ancillary qubits is given.

but in the *unary* encoding $\sum_{k=0}^{2^n-1} v_k|e_k\rangle$, where $e_i \in \{0,1\}^{2^n}$ is the vector with the k-th bit being 1 and all other bits being 0. The paper shows that the unary encoding QSP can be carried out by a quantum circuit of depth $O(n)$ and size $O(2^n)$. Note that the unary encoding itself takes 2^n qubits, as opposed to n qubits in the binary encoding. The binary encoding is the most efficient one in terms of the number of qubits needed for the resulting state, and indeed in most quantum machine learning tasks the quantum speedup depends crucially on this encoding efficiency at the first place [7,9,31–34]. In [30] the authors also extended this by using a d-dimensional tensor $(k_1, k_2, \ldots, k_d)$ to encode k, which needs $d2^{n/d}$ qubits to encode and a circuit of depth $O(\frac{n}{d}2^{n-n/d})$ to prepare. When $d = n$ the encoding coincides with the binary encoding, but their depth bound is $O(2^n)$, which is not optimal.

In this paper, we tightly characterize the depth and size complexities of the quantum state preparation problem by constructing optimal quantum circuits. Our circuits generate the target state precisely, with certainty, and use an optimal number of ancillary qubits. We present our results on QSP first, where a general number m of ancillary qubits are available.

Theorem 1. *For any $m \geq 2n$, any n-qubit quantum state $|\psi_v\rangle$ can be generated by a circuit with m ancillary qubits, using single-qubit gates and CNOT gates, of size $O(2^n)$ and depth*

$$\begin{cases} O\Big(\frac{2^n}{m+n}\Big), & \text{if } m \in [2n, O(\frac{2^n}{n\log n})], \\ O\left(n\log n\right), & \text{if } m \in [\omega(\frac{2^n}{n\log n}), o(2^n)], \\ O\left(n\right), & \text{if } m = \Omega(2^n). \end{cases}$$

These depth bounds improve the depth of $O(2^n)$ in [25,26] by a factor of m for any $m \in [2n, O(\frac{2^n}{n\log n})]$, and the result shows that more ancillary qubits can indeed provide more help in shortening the depth for QSP. Compared with the result in [28] which needs $O(4^n)$ ancillary qubits to achieve depth $O(n^2)$, ours needs only $m = O(2^n/n^2)$ qubits to reach the same depth. In addition, our circuit is deterministic and generates the state with certainty, and the only two-qubit gates used are the CNOT gates.

The above construction needs at least $2n$ ancillary qubits. Next we show an optimal depth construction of circuits without ancillary qubits.

Theorem 2. *Any n-qubit quantum state $|\psi_v\rangle$ can be generated by a quantum circuit, using single-qubit gates and CNOT gates, of depth $O(2^n/n)$ and size $O(2^n)$, without using ancillary qubits.*

These two theorems combined give asymptotically optimal bounds for depth and size complexity. Indeed, a lower bound of $\Omega(2^n)$ for size is known [26], and the same paper also presents a depth lower bound of $\Omega(2^n/n)$ for quantum circuits without ancillary qubits. This can be extended to a lower bound of $\Omega\Big(\frac{2^n}{n+m}\Big)$ for circuits with m ancillary qubits. This bound deteriorates to 0 as m grows to infinity. In [35], the authors gave a depth lower bound of $\Omega(\log n)$ for circuit with arbitrarily many ancillary qubits. We note that it can be improved to $\Omega(n)$ for any m, as stated in the next theorem as well as independently discovered in [28].

Theorem 3. *Given m ancillary qubits, there exist n-qubit quantum states which can only be prepared by quantum circuits of depth at least $\Omega\Big(\max\Big\{n, \frac{2^n}{m+n}\Big\}\Big)$, for circuits using arbitrary single-qubit and 2-qubit gates.*

The proof of Theorem 3 is shown in Appendix A.

Putting the above results together, we can tightly characterize the size and depth complexity of QSP, except for a logarithmic factor gap over a small parameter regime for m. It is interesting to note that our circuits achieve the optimal depth and size simultaneously. Our results are summarized in the next Corollary 4 and illustrated in Figure 1.

Corollary 4. *For a circuit preparing an n-qubit quantum state with m ancillary qubits, the minimum size is $\Theta(2^n)$, and the minimum depth $D_{\mathrm{QSP}}(n,m)$ for different ranges of m are characterized as follows.*

$$\begin{cases} \Theta\Big(\frac{2^n}{m+n}\Big), & \text{if } m = O\Big(\frac{2^n}{n\log n}\Big), \\ [\Omega(n), O(n\log n)]\,, & \text{if } m \in [\omega\Big(\frac{2^n}{n\log n}\Big), o\left(2^n\right)], \\ \Theta(n), & \text{if } m = \Omega\left(2^n\right). \end{cases}$$

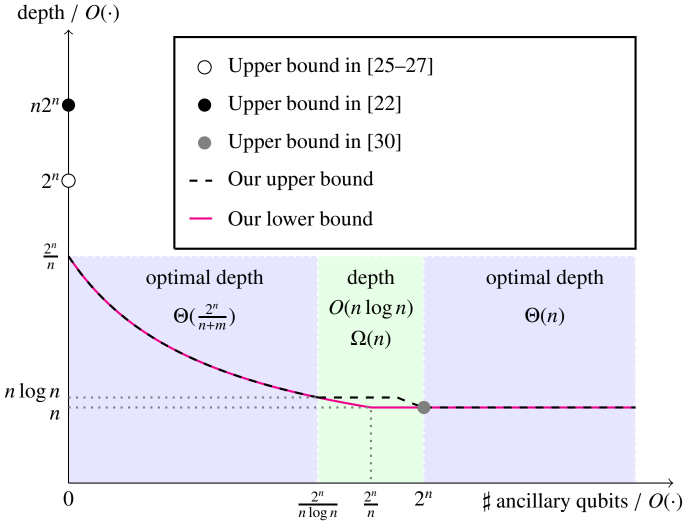


Figure 1: Circuit depth upper and lower bound for n-qubit quantum state preparation. m denote the number of ancillary qubits. If $m = O(\frac{2^n}{n\log n})$ and $\Omega(2^n)$, our circuit depths are $\Theta\left(\frac{2^n}{n+m}\right)$ and $\Theta(n)$, which are asymptotically optimal. When $m \in [\omega(\frac{2^n}{n\log n}), o(2^n)]$, the gap between our depth upper and lower bound is at most logarithmic.

Now we give two applications of the result, the first of which is general unitary synthesis. Given a unitary matrix, a fundamental question is to find a circuit implementing it in optimal depth or size. Previous studies on this problem focus on circuits without ancillary qubits. Barenco *et al.* [36] gave an upper bound $O(n^3 4^n)$ for the number of CNOT gates for arbitrary n-qubit unitary matrix. Knill [37] improved the upper bound to $O(n4^n)$. Vartiainen *et al.* [38] constructed a quantum circuit for an n-qubit unitary matrix with $O(4^n)$ CNOT gates. Mottonen and Vartiainen [27] designed a quantum circuit of depth $O(4^n)$ using $\frac{23}{48}4^n$ CNOT gates. The best known lower bound for *number* of CNOT gates is $\left\lceil \frac{1}{4}(4^n - 3n - 1)\right\rceil$ [39], which also implies a depth lower bound of $\Omega(4^n/n)$. In a nutshell, the previous work put the optimal depth to within the range of $[\Omega(4^n/n), O(4^n)]$ for general n-qubit circuit compression without ancillary qubits.

Our results on QSP can be applied to close this gap, by showing a circuit of depth $O(4^n/n)$. And this is actually a special case of the next theorem which handles a general number m of ancillary qubits.

Theorem 5. *Any unitary matrix $U \in \mathbb{C}^{2^n\times 2^n}$ can be implemented by a quantum circuit of size $O(4^n)$ and depth $O\left(n2^n + \frac{4^n}{m+n}\right)$ with $m \leq 2^n$ ancillary qubits.*

The second application of our QSP result is approximate QSP, for which one can obtain the following bound for circuit with a finite set of gates such as $\{CNOT, H, S, T\}$ using a variant of the Solovay–Kitaev theorem.

Corollary 6. *For any n-qubit target state $|\psi_v\rangle$, one can prepare a state $|\psi'_v\rangle$ which is ϵ-close to $|\psi_v\rangle$ in ℓ_2-distance, by a circuit consisting of $\{CNOT, H, S, T\}$ gates of depth*

$$\begin{cases} O\left(\frac{2^n\log(2^n/\epsilon)}{m+n}\right), & \text{if } m = O\left(\frac{2^n}{n\log n}\right),\\ O(n\log n\log(2^n/\epsilon)), & \text{if } m \in \left[\omega\left(\frac{2^n}{n\log n}\right), o(2^n)\right],\\ O(n\log(2^n/\epsilon)), & \text{if } m = \Omega(2^n),\end{cases}$$

using m ancillary qubits.

Proof techniques We give a brief account of the proof techniques used in our circuit constructions. We first reduce the problem to implementing diagonal unitary matrices. Making a phase shift for each computational basis state costs at least $\Omega(n2^n)$-size, which is unnecessarily high. We make the shift in

Fourier basis, and carefully use ancillary qubits to parallelize the process. With ancillary qubits, we can first make some copies of the computational basis variables x_i, then partition $\{0,1\}^n$ into some parts of equal size, and use the ancillary qubits to handle different parts in parallel. We define the partition via a Gray code to minimize the update cost. Gray codes were also used in [40] to minimize the circuit size. They only need to minimize the difference between adjacent two words, so the defining property of Gray Code is enough. In our construction, however, we also need to make sure that the changed bits in different parts of the Gray code are evenly distributed.

When no ancillary qubits are available, designing efficient circuit needs more ideas. Since there is no ancillary qubit available, all phase shifts need be made inside the input register. We divide the input register into two parts, control register and target register, and make phase shifts in the latter. As we only have a small space, we cannot use it to enumerate all $2^{r_t}-1$ suffixes as in the previous case, where $r_t \approx n/2$ is the length of suffixes. But we can enumerate them in many stages, by which we pay the price of time to compensate for the shortage of space. We need to make a transition between two consecutive stages. It turns out that the transition can be realized by a low-depth circuit if the suffixes enumerated in each stage are linearly independent as vectors over $\mathbb{F}_2$. Thus we need carefully divide the set of suffixes into sets of linearly independent vectors to facilitate the efficient update. Some other parts need special treatment as well. One is that we need to reset the suffix to the original input variables after going along a Gray code path. Another one is that the all-zero suffix cannot be handled in the same way for some singularity reason, for which we will use a recursion to solve the issue. It turns out that the overall depth and size obtained this way are asymptotically optimal.

The above constructions work well when m is relatively small, but do not give a tight bound when $m = \Omega(2^n/n^2)$, for which we use another method. As we mentioned earlier, [30] shows that unary-encoded QSP can be made in $O(n)$ depth and $O(2^n)$ size. Though the resulting state uses an exponentially long unary encoding, we can transform it to a binary encoding. A direct parallelization for this transform takes $O(n2^n)$ ancillary qubits, which can be improved to the $O(2^n)$ by first transforming it to a $2^{n/2+1}$-long matrix encoding $|e_i\rangle \to |e_s\rangle|e_t\rangle$, and then to the binary encoding. This gives the optimal depth and size for the regime $m \geq 2^n$. For $m \in [\omega(2^n/n^2), o(2^n)]$, the ancillary qubits only suffice for conducting the above for the first $\log_2 m$ qubits of the target state. For the rest $\leq 2\log_2 n$ qubits, we invoke our first construction to complete the generation. This gives the optimal depth if $m \in [\omega(2^n/n^2), O(2^n/(n\log n))]$, the overall depth is asymptotically optimal, leaving a gap $[\Omega(n), O(n\log n)]$ only when m is in a small range $[\omega(2^n/(n\log n)), o(2^n)]$.

Other related work Besides the standard QSP, researchers have also studied some relaxed versions. Araujo *et al.* [41] have given a depth upper bound of $O(n^2)$ to prepare a state $\sum_{k=0}^{2^n-1} v_k|k\rangle|\text{garbage}_k\rangle$, where $|\text{garbage}_k\rangle$ is $O(2^n)$-qubit state entangled with the target state register. Note that there is no generic way to remove the entangled garbage, this cannot be directly used to solve the standard QSP problem.

One may also consider to approximately prepare quantum states by quantum circuits made of $\{H, S, T, CNOT\}$ gates to generate $|\psi_v'\rangle$ satisfying $\left\||\psi_v'\rangle - |\psi_v\rangle\right\| \leq \epsilon$ for various distance measures $\|\cdot\|$. Previous attention was paid to minimizing the number and depth of T gates [42, 43], which is non-Clifford and usually thought to be hard to realize experimentally [42]. They have applied ancillary qubits to implement a circuit such that the number of T gates can be optimized to $\frac{2^n}{\lambda} + \lambda\log^2\frac{2^n\lambda}{\epsilon}$ [42], where $\lambda \in [1, O(\sqrt{2^n})]$. We shall show that our construction can be adapted to this gate set and the circuit depth increases only by $O(n + \log(1/\epsilon))$.

Subsequent work After this work appeared on arXiv [44], Rosenthal [45] constructed a QSP circuit of depth $O(n)$, using $O(n2^n)$ ancillary qubits, as opposed to ours that only uses $O(2^n)$ ancillary qubits. Rosenthal also presented a circuit for general unitary synthesis of depth $\tilde{O}(2^{n/2})$ using $\tilde{O}(4^n)$ ancillary qubits. This year, Zhang *et al.* [46] presented yet another QSP circuit of depth $O(n)$ using $\Theta(2^n)$ ancillary qubits, which is a special cases of our results.

Organization The rest of this paper is organized as follows. In Section 2, we will review notations and a framework of quantum state preparation. Then we will present how to decompose the uniformly controlled gate to diagonal unitary matrices and show the depth of quantum state preparation when the number of ancillary qubits $m = O(2^n/n^2)$ in Section 3. Next we will show two quantum circuit for diagonal unitary matrices used in previous section, with and without ancillary qubits in Section 4 and Section 5, respectively. Furthermore, we present a new circuit framework for quantum state preparation when $m = \Omega\left(2^n/n^2\right)$ in Section 6. In Section 7, we will show some extensions and implications of the above bounds. Finally we conclude in Section 8.

2 Preliminaries

In this section, we will introduce some basic concepts and notation.

Notation Let $[n]$ denote the set $\{1, 2, \cdots, n\}$. All logarithms $\log(\cdot)$ are base 2 in this paper. Let $\mathbb{I}_n \in \mathbb{R}^{2^n \times 2^n}$ be the n-qubit identity operator. Denote by $\mathbb{F}_2$ the field with 2 elements, with multiplication $\cdot$ and addition $\oplus$, which can be overloaded to vectors: $x \oplus y = (x_1 \oplus y_1, x_2 \oplus y_2 \cdots, x_n \oplus y_n)^T$ for any $x, y \in \mathbb{F}_2^n$. The inner product of two vectors $s, x \in \mathbb{F}_2^n$ is $\langle s, x\rangle := \oplus_{i=1}^n s_i \cdot x_i$ in which the addition and multiplication are over $\mathbb{F}_2$. We use 0^n and 1^n for the all-zero and all-one vectors of length n, respectively. Vector e_i is the vector where the i-th element is 1 and all other elements are 0. The multiplication $\cdot$ is sometimes dropped if no confusion is caused. For $t, k \geq 1$ and $U_1, \ldots, U_k \in \mathbb{C}^{t \times t}$, $diag(U_1, U_2, \ldots, U_k)$ is defined as

$$diag(U_1, \ldots, U_k) \overset{\text{def}}{=} \begin{bmatrix} U_1 & & \\ & \ddots & \\ & & U_k \end{bmatrix} \in \mathbb{C}^{kt \times kt}.$$

Elementary gates We will use the following $R_y(\theta)$, $R_z(\theta)$ and $R(\theta)$ to denote 1-qubit rotation (about Y-axis, Z-axis) gates and phase-shift gate, i.e.,

$$R_y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \quad R_z(\theta) = \begin{bmatrix} e^{-i(\theta/2)} & \\ & e^{i(\theta/2)} \end{bmatrix}, \quad R(\theta) = \begin{bmatrix} 1 & \\ & e^{i\theta} \end{bmatrix},$$

where $\theta \in \mathbb{R}$ is a parameter. All blank elements denote zero throughout this paper. Three important and special cases are the $\pi/8$ gate T, the phase gate S and the Hadamard gate H,

$$T = \begin{bmatrix} 1 & \\ & e^{i\pi/4} \end{bmatrix}, \; S = \begin{bmatrix} 1 & \\ & i \end{bmatrix}, \; H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

The 2-qubit controlled-NOT gate is

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

The gate flips the *target qubit* conditioned that the *control qubit* is $|1\rangle$.

Single-qubit gate decomposition Any single-qubit operator $U \in \mathbb{C}^{2\times 2}$ can be decomposed as

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta)$$

for some $\alpha, \beta, \gamma, \delta \in \mathbb{R}$ [23]. It is not hard to verify that the Y-axis rotation $R_y(\gamma) \in \mathbb{R}^{2\times 2}$ can be decomposed as $R_y(\gamma) = SHR_z(\gamma)HS^\dagger$, for any $\gamma \in \mathbb{R}$. Putting these two facts together, we know that for a single-qubit operation U, there exist $\alpha, \beta, \gamma, \delta \in \mathbb{R}$ such that

$$U = e^{i\alpha} R_z(\beta) SHR_z(\gamma)HS^\dagger R_z(\delta). \tag{1}$$

Gray code A Gray code path is an ordering of all n-bit strings $\{0, 1\}^n$ in which any two adjacent strings differ by exactly one bit [47–49], and the first and the last string differ by one bit. That is, a Gray code path/cycle is a Hamiltonian path/cycle on the Boolean hypercube graph. Gray code paths/cycles are not unique, and a common one, called reflected binary code (RBC) or Lucal code, is as follows. Denote the ordering of n-bit strings by $x^1, x^2, \dots, x^{2^n}$ and we will construct them one by one. Take $x^1 = 0^n$. For each $i = 1, 2, \dots, 2^n - 1$, the next string x^{i+1} is obtained from x^i by flipping the $\zeta(i)$-th bit, where the Ruler function $\zeta(i)$ is defined as $\zeta(i) = \max\{k : 2^{k-1}|i\}$. In other words, $\zeta(i)$ is 1 plus the exponent of 2 in the prime factorization of i. The following fact is easily verified.

Lemma 7. *The reflected binary code defined above is a Gray code cycle.*

Note that in the above construction, if we list all the bits changed between circularly adjacent strings, we will get a list of length 2^n. For instance, when $n = 4$, the list is: 1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4. In general, bit 1 appears 2^{n-1} times, bit 2 appears 2^{n-2} times, ..., bit $n - 1$ appears twice, and bit n appears twice as well. If we regard the code as a path, i.e. ignore the change of bit from the last string to the first string, then bit n appears once.

By circularly shifting the bits, we can also construct Gray code cycle such that bit 2 appears 2^{n-1} times, ..., bit n and bit 1 appear twice. In general, for any $k \in [n]$, we can make each bit $k, k + 1, \dots, n, 1, 2, \dots, k - 1$ to appear 2^n, $2^{n-1}, 2^{n-2}, \dots, 2^2, 2, 2$ times, respectively. Let us call this construction (k, n)-*Gray code path/cycle*, or simply the k-Gray code path/cycle if n is clear from context.

3 Quantum state preparation with $O(2^n/n^2)$ ancillary qubits

In this section, we will review a natural framework of algorithm for quantum state preparation, first appeared in [22]. Our results presented in Section 4 and 5, which achieve the optimal circuit depth, also fall into this framework. The framework to prepare an n-qubit quantum state is depicted in Figure 2(a), where each qubit j is handled by the circuit V_j. The task for V_j is to apply a single-qubit unitary on the last qubit conditioned on the basis state of the first $j - 1$ qubits. In a matrix form, V_j is a block-diagonal operator

$$V_j = diag(U_1, U_2, \dots, U_{2^{j-1}}) \in \mathbb{C}^{2^j \times 2^j}, \tag{2}$$

where each U_i is a 2×2 unitary matrix. There are different ways to implement V_j, and the most natural one, which is also the one suggested in [22], is in Figure 2(b): it includes 2^{j-1} layers, and each layer is a controlled gate, which conditions on every possible computational basis state of the previous $j - 1$ qubits and operates on the current qubit j. This is why sometimes V_j is called *uniformly controlled gate* (UCG). We give a specific example for illustration in Appendix B.

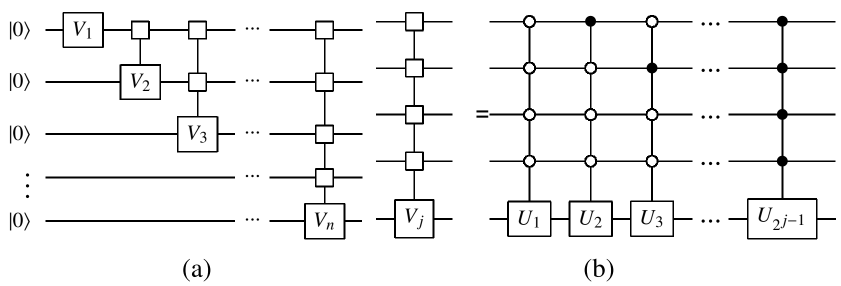


Figure 2: (a) A quantum circuit to prepare an n-qubit quantum state. Every V_j ($j \in [n]$) is a j-qubit uniformly controlled gate, where the first $j - 1$ qubits are controlled qubits and the last one qubit is the target qubit. (b) A j-qubit uniformly controlled gate.

Thus the depth of the circuit for quantum state preparation in the above framework crucially depends on the circuit depth of the implementation of V_j's.

Lemma 8. *If each V_j can be implemented by a quantum circuit of depth d_j, then the quantum state can be prepared by a circuit of depth $\sum_{j=1}^{n} d_j$.*

As mentioned in Section 1, if we implement each V_j directly as in [22], then the whole QSP circuit has a depth of $\Theta(n2^n)$, which is sub-optimal compared to our bound of $\Theta(2^n/n)$ in Theorem 2. More importantly, the method in [22] cannot well utilize ancillary qubits to reduce the circuit depth. In this section, we will give a framework of efficient implementation of UCGs with the help of $m = O(2^n/n^2)$ ancillary qubits. The case when more ancillary qubits are available, i.e. $m = \Omega(2^n/n^2)$, is handled by a different framework in Section 6.

To overcome these drawbacks, we will first reduce the implementation of UCG to that of diagonal operators of the following form:

$$\Lambda_n = diag(1, e^{i\theta_1}, e^{i\theta_2}, \dots, e^{i\theta_{2^n-1}}) \in \mathbb{C}^{2^n \times 2^n}. \tag{3}$$

Lemma 9. *If one can implement Λ_n in Eq.* (3) *by a circuit of depth $D(n)$ and size $S(n)$ using $m \geq 0$ ancillary qubits, then any n-qubit quantum state can be prepared by a circuit of depth $3\sum_{k=1}^{n} D(k) + 2n + 1$ and size $3\sum_{k=1}^{n} S(k) + 2n + 1$.*

Proof. According to Eq. (1), each unitary matrix $U_k \in \mathbb{C}^{2\times 2}$ can be decomposed as

$$U_k = e^{i\alpha_k} R_z(\beta_k) SHR_z(\gamma_k) HS^{\dagger} R_z(\delta_k).$$

Then the UCG V_n can thus be decomposed to

$$\begin{aligned} V_n = &\underbrace{diag(e^{i\alpha_1}, \cdots, e^{i\alpha_{2^{n-1}}}) \otimes \mathbb{I}_1}_{A_1} \cdot \underbrace{diag(R_z(\beta_1), \cdots, R_z(\beta_{2^{n-1}}))}_{A_2} \\ &\cdot \underbrace{\mathbb{I}_{n-1} \otimes (SH)}_{A_3} \cdot \underbrace{diag(R_z(\gamma_1), \cdots, R_z(\gamma_{2^{n-1}}))}_{A_4} \cdot \underbrace{\mathbb{I}_{n-1} \otimes (HS^{\dagger})}_{A_5} \cdot \underbrace{diag(R_z(\delta_1), \cdots, R_z(\delta_{2^{n-1}}))}_{A_6}. \end{aligned} \tag{4}$$

Note that the unitary matrix A_3 can be implemented by a Hadamard gate H and a phase gate S operating on the last qubit, and similarly for A_5. The rest matrices, A_1, A_2, A_4, and A_6 are all n-qubit diagonal unitary matrices. Since a global phase can be easily implemented by a rotation on any one qubit, we can focus on implementing diagonal matrices of the form as in Eq. (3). If Λ_n can be implemented by a circuit of depth $D(n)$ and size $S(n)$, so will be QSP by a circuit of depth and size $\sum_{k=1}^{n}(3D(k) + 2) + 1 = 3\sum_{k=1}^{n} D(k) + 2n + 1$ and $\sum_{k=1}^{n}(3S(k) + 2) + 1 = 3\sum_{k=1}^{n} S(k) + 2n + 1$, where the terms "$3D(k)$" and "$3S(k)$" are for diagonal matrices A_1, A_2, A_4 and A_6, the term "2" is for A_3 and A_5, and the term "1" is for the global phase. □

Thus we only need to consider how to implement diagonal operators as in Eq. (3). We will prove the following lemmas in Section 4 and Section 5.

Lemma 10. *For any $m \in [2n, 2^n/n]$, any diagonal unitary matrix $\Lambda_n \in \mathbb{C}^{2^n \times 2^n}$ as in Eq.* (3) *can be implemented by a quantum circuit of depth $O\left(\log m + \frac{2^n}{m}\right)$ and size $O(2^n)$, with m ancillary qubits.*

Lemma 11. *Any diagonal unitary matrix $\Lambda_n \in \mathbb{C}^{2^n \times 2^n}$ as in Eq.* (3) *can be implemented by a quantum circuit of depth $O\left(\frac{2^n}{n}\right)$ and size $O\left(2^n\right)$ without ancillary qubits.*

Lemmas 10 and 11 imply Lemma 12.

Lemma 12. *For $m \geq 0$, any uniformly controlled gate $V_n \in \mathbb{C}^{2^n \times 2^n}$ as in Eq.* (2) *can be implemented by a quantum circuit of depth $O\left(n + \frac{2^n}{n+m}\right)$ and size $O(2^n)$ with m ancillary qubits.*

Proof. According to Eq. (4), every V_n can be decomposed into 3 n-qubit diagonal unitary matrices and 4 single-qubit gates. Combining with Lemma 10 and 11, V_n can be realized by a quantum circuit of depth $O\left(n + \frac{2^n}{n+m}\right)$ and size $O(2^n)$ with m ancillary qubits. □

Once we prove these two lemmas, we will be able to prove Theorems 1 and 2. Indeed, we can apply the next Lemma 13 to prove Theorem 2 ($m = 0$) and the $m = O(2^n/n^2)$ part of Theorem 1. The other part $m = \Omega(2^n/n^2)$ of Theorem 1 is the same as Corollary 30 and will be treated in Section 6.

Lemma 13. *For any $m \geq 0$, any n-qubit quantum state $|\psi_v\rangle$ can be generated by a quantum circuit with m ancillary qubits, using single-qubit gates and CNOT gates, of size $O(2^n)$ and depth $O\Big(n^2 + \frac{2^n}{m+n}\Big)$.*

Proof. We prove the case $m = 0$ first. Plugging Lemma 11 into Lemma 9, we get a circuit solving QSP in size $\sum_{j=1}^{n} O(2^j) + 2n + 1 = O(2^n)$ and depth $O\Big(\sum_{j=1}^{n} \frac{2^j}{j} + n\Big) = O\Big(\sum_{j=1}^{n-\lceil \log n\rceil} \frac{2^j}{j} + \sum_{j=n-\lceil \log n\rceil+1}^{n} \frac{2^j}{j}\Big) = O\Big(\sum_{j=1}^{n-\lceil \log n\rceil} 2^j + \sum_{j=n-\lceil \log n\rceil+1}^{n} \frac{2^j}{n-\lceil \log n\rceil+1}\Big) = O\Big(\frac{2^n}{n}\Big)$, as desired.

Now we prove the case $m > 0$. If $1 \leq m < 2n$, we will not use the ancillary qubits—we just invoke Theorem 2 to obtain a circuit of depth $O(2^n/n)$. If $2n \leq m \leq 2^n/n^2 (\leq 2^n/n)$, we can combine Lemma 9 and Lemma 10 to give a circuit of size $3\sum_{j=1}^{n} O(2^j)+2n+1 = O(2^n)$ and depth $O\Big(\sum_{j=1}^{n}\Big(\log m+\frac{2^j}{m}\Big)+n\Big) = O\Big(n^2 + \frac{2^n}{m}\Big)$. If $m > 2^n/n^2$, we only use the first $2^n/n^2$ ancillary qubits, then the above equality gives a circuit of depth $O(n^2)$. Putting these three cases together, we obtain the claimed size upper bound of $O(2^n)$ and depth upper bound of $O\Big(n^2 + \frac{2^n}{m+n}\Big)$. □

Next let us consider how to efficiently implement Λ_n, which essentially makes a phase shift on each computational basis state. Again, if we do this on each basis state, it takes at least $\Omega(2^n)$ rounds, with each round implementing an n-qubit controlled phase shift. One way of avoiding sequential applications of $(n-1)$-qubit controlled unitaries is to make rotations on its Fourier basis. Indeed, there are several pieces of work to synthesis a diagonal unitary matrix, and a common approach is generating all the linear functions of variables and adding corresponding rotation $R(\theta)$ gate when a new combination generated [40, 50, 51]. In [40] the authors use Gray code to adjust the order of combinations so the size and depth of the circuit are $O(2^n)$. With ancillary qubits, we can actually achieve this with much smaller depth by carefully parallelizing the operations (Section 4). Interestingly, this approach turns out to inspire our construction for circuits *without* ancillary qubits (Section 5), to achieve the optimal depth complexity as in Theorem 2.

We now give more details. Suppose we can accomplish the following two tasks:

1. For every $s \in \{0,1\}^n - \{0^n\}$, make a phase shift of α_s on each basis $|x\rangle$ when $\langle s, x\rangle = 1$ (recall that $\langle\cdot,\cdot\rangle$ is over $\mathbb{F}_2$), i.e.
$$|x\rangle \rightarrow e^{i\alpha_s\langle s,x\rangle}|x\rangle. \tag{5}$$

2. Find $\{\alpha_s : s \in \{0,1\}^n - \{0^n\}\}$ s.t.
$$\sum_{s\in\{0,1\}^n-\{0^n\}} \alpha_s\langle x, s\rangle = \theta(x), \quad \forall x \in \{0,1\}^n - \{0^n\}. \tag{6}$$

Then we get
$$|x\rangle \rightarrow \prod_{s\in\{0,1\}^n-\{0^n\}} e^{i\alpha_s\langle s,x\rangle}|x\rangle = e^{i\Sigma_s\alpha_s\langle s,x\rangle}|x\rangle = e^{i\theta(x)}|x\rangle,$$

as required in Λ_n. For notational convenience, we define $\alpha_{0^n} = 0$.

The implementations of above two tasks in Eq. (5) and Eq. (6) are accomplished in Appendix C.

4 Diagonal unitary implementation with ancillary qubits

In this section, we prove Lemma 10. That is, for any $m \in [2n, 2^n/n]$, any diagonal unitary matrix $\Lambda_n \in \mathbb{C}^{2^n\times 2^n}$ as in Eq. (3) can be implemented by a quantum circuit of depth $O\Big(\log m + \frac{2^n}{m}\Big)$ and size $O(2^n)$ with m ancillary qubits. Let us first give a high-level explanation of the circuit. We divide the ancillary qubits into two registers: One is used to make multiple copies of basis input bits to help on parallelization, and the other is used to generate all n-bit strings and apply the rotation gates. State

$|\langle s, x\rangle\rangle$ will be generated for all $s \in \{0,1\}^n - \{0^n\}$. To reduce the depth of the circuit, these strings are split as equally as possible, and we use Gray Code to minimize the cost of generating a new n-bit string from an old one. A quantum circuit to implement Λ_4 by using 8 ancillary qubits is shown in Appendix D.

We will show how to implement Λ_n with m ancillary qubits. Let us assume m to be an even number to save some floor or ceiling notation without affecting the bound. The framework is shown in Figure 3. Our framework consists of three registers and five stages. The first n qubits labeled as $x_1, x_2, \cdots, x_n$ form the *input register*, the next $\frac{m}{2}$ qubits are the *copy register*, and the last $\frac{m}{2}$ qubits are the *phase register*. The linear functions $\langle s, x\rangle$ of the input variables $x = x_1 \ldots x_n$ are generated in the phase register. We use the copy register to make copies of x for parallelizing the circuit later. Partition s into a prefix s_1 and a suffix s_2. We then generate a specific function $\langle s_1 0 \cdots 0, x\rangle$ on each qubit in the phase register, and iterate other non-zero suffixes s_2 in the order of a Gray code and generate $\langle s_1 s_2, x\rangle$. All qubits in the copy and phase registers are initialized to $|0\rangle$.

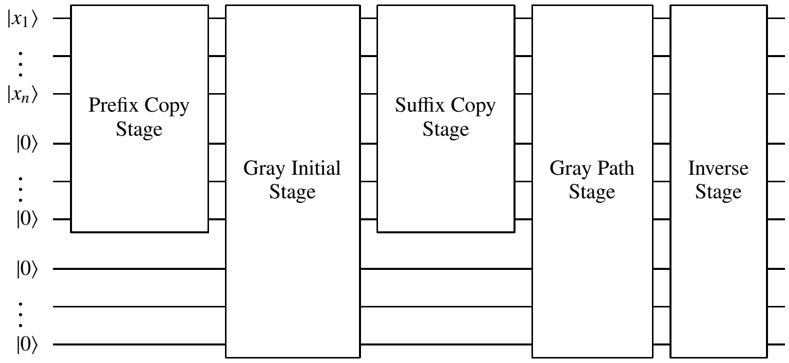


Figure 3: Framework for the circuit of Λ_n with m ancillary qubits. The first n qubits $|x_1 \cdots x_n\rangle$ form the input register, the next $\frac{m}{2}$ qubits the copy register and the last $\frac{m}{2}$ qubits the phase register. The framework consists of five stages: Prefix Copy, Gray Initial, Suffix Copy, Gray Path and Inverse. The depth of the five stages are $O(\log m)$, $O(\log m)$, $O(\log m)$, $O\left(\frac{2^n}{m}\right)$ and $O\left(\log m + \frac{2^n}{m}\right)$, respectively.

Stage 1: Prefix Copy In this stage, we make $\left\lfloor \frac{m}{2t} \right\rfloor$ copies of each qubit $x_1, x_2, \cdots, x_t$ in the input register, where $t = \left\lfloor \log \frac{m}{2} \right\rfloor < n$. More formally, the circuit implements the unitary $U_{copy,1}$ which operates on the input and copy registers only. Its effect is

$$|x\rangle\, |0^{m/2}\rangle \xrightarrow{U_{copy,1}} |x\rangle\, |x_{pre}\rangle \tag{7}$$

where the two parts in the ket notation are for the input and copy register, respectively, and

$$|x\rangle = |x_1 x_2 \cdots x_n\rangle\,,$$

$$|x_{pre}\rangle = |\overbrace{\underbrace{x_1 \cdots x_1}_{\left\lfloor \frac{m}{2t} \right\rfloor \text{ qubits}} \underbrace{x_2 \cdots x_2}_{\left\lfloor \frac{m}{2t} \right\rfloor \text{ qubits}} \cdots \underbrace{x_t \cdots x_t}_{\left\lfloor \frac{m}{2t} \right\rfloor \text{ qubits}} 0 \cdots 0}^{m/2 \text{ qubits}}\rangle.$$

The next lemma says that this operation can be carried out by a circuit of small depth.

Lemma 14. *We can make $\left\lfloor \frac{m}{2t} \right\rfloor$ copies of each qubit $x_1, x_2, \cdots, x_t$ in the input register and the copy register, by an $(m/2)$-size circuit $U_{copy,1}$ of CNOT gates only, in depth at most $\log m$.*

Proof. First, we make 1 copy of each x_i in the input register to a qubit in the copy register by applying a CNOT gate. Note that the CNOT gates for different x_i's are applied on different pairs of qubits, thus they can be implemented in parallel in depth 1. Next, we utilize the x_i in the input register and the x_i in the copy register (that we just obtained) to make two more copies of x_i in the copy register, and again all these $2t$ CNOT gates can be implemented in depth 1. We continue this until we get $\lfloor m/(2t) \rfloor$ copies of each qubit $x_1, x_2, \cdots, x_t$ in the copy register. The depth of this Copy stage is $\left\lceil \log \left\lfloor m/2t \right\rfloor \right\rceil \leq \log m$. And the size of this stage is $m/2$, since each qubit in copy register is used as the target qubit of CNOT gate only once. □

Stage 2: Gray Initial In this stage, the circuit includes two steps. The first step U_1 implements $m/2$ linear functions $f_{j1}(x) = \langle s(j,1), x \rangle$ for some n-bit strings $s(j,1)$, one for each qubit j in the phase register. The second step implements some rotations in the phase registers. To elaborate on which strings are implemented in the first step, we need the following lemma and notation. Recall that we are in the parameter regime $m \in [2n, 2^n/n]$.

Lemma 15. *Let $t = \lfloor \log \frac{m}{2} \rfloor$ and $\ell = 2^t$. The set $\{0,1\}^n$ can be partitioned into a 2-dimensional array $\{s(j,k) : j \in [\ell], k \in [2^n/\ell]\}$ of n-bit strings, satisfying that*

1. *Strings in the first column $\{s(j,1) : j \in [\ell]\}$ have the last $(n-t)$ bits being all 0, and strings in each row $\{s(j,k) : k \in [2^n/\ell]\}$ share the same first t bits.*
2. *$\forall j \in [\ell], \forall k \in [2^n/\ell - 1]$, $s(j,k)$ and $s(j,k+1)$ differ by 1 bit.*
3. *For any fixed $k \in [2^n/\ell - 1]$, and any $t' \in \{t+1, ..., n\}$, there are at most $\left(\frac{m}{2(n-t)} + 1\right)$ many $j \in [\ell]$ s.t. $s(j,k)$ and $s(j,k+1)$ differ by the t'-th bit.*

The proof of Lemma 15 is shown in Appendix E.

Let us denote by t_{jk} the index of the bit that $s(j,k)$ and $s(j,k+1)$ differ by. We can now describe this stage in more details.

1. The first step U_1 aims to let each qubit j in the phase register have the state $|f_{j1}(x)\rangle$ at the end of this step, where $f_{j1}(x) = \langle s(j,1), x \rangle$.
2. The second step applies the rotation $R_{j,1} \overset{\text{def}}{=} R(\alpha_{s(j,1)})$ on each qubit j in the phase register. That is, the state is rotated by an phase angle of $\alpha_{s(j,1)}$ if $\langle x, s(j,1) \rangle = 1$, and left untouched otherwise. Put $R_1 = \otimes_{j \in [\ell]} R_{j,1}$.

The next lemma gives the cost and effect of this stage.

Lemma 16. *The Gray Initial Stage can be implemented in depth at most $2\log m$ and in size at most $\frac{(n+1)m}{2}$ such that its unitary $U_{GrayInit}$ satisfies*

$$|x\rangle |x_{pre}\rangle |0^{m/2}\rangle \xrightarrow{U_{GrayInit}} e^{i \sum_{j \in [\ell]} f_{j,1}(x) \alpha_{s(j,1)}} |x\rangle |x_{pre}\rangle |f_{[\ell],1}\rangle, \tag{8}$$

where $|f_{[\ell],1}\rangle = \otimes_{j \in [\ell]} |f_{j,1}(x)\rangle$.

Proof. We will show how to implement the first step U_1 such that all $\ell = 2^t = 2^{\lfloor \log \frac{m}{2} \rfloor}$ linear functions of the prefix variables $x_1, \ldots, x_t$ are implemented, namely after U_1, the states of the 2^t qubits in the phase register are exactly $\{a_1 x_1 \oplus \cdots \oplus a_t x_t : a_1, \ldots, a_t \in \{0,1\}\}$. The implementation makes each qubit j in the phase register have state $|f_{j,1}(x)\rangle$. Then in the second step, each qubit j adds a phase of $f_{j,1}(x) \cdot \alpha_{s(j,1)}$ to $|x\rangle |x_{pre}\rangle |0^{m/2}\rangle$. We thus have

$$|x\rangle |x_{pre}\rangle |0^{m/2}\rangle \xrightarrow{U_1} |x\rangle |x_{pre}\rangle |f_{[\ell],1}\rangle, \tag{9}$$

$$\xrightarrow{R_1} e^{i \sum_{j \in [\ell]} f_{j,1}(x) \alpha_{s(j,1)}} |x\rangle |x_{pre}\rangle |f_{[\ell],1}\rangle. \tag{10}$$

Now let us construct a shallow circuit for the first step U_1. Recall that we have $\ell = 2^t$ qubits j each with a corresponding linear function in variables $x_1, \ldots, x_t$. Since $\ell \leq m/2$, the phase register has enough qubits to hold these linear functions. For a qubit j in the phase register with corresponding linear function $x_{i_1} \oplus \cdots \oplus x_{i_{t'}}$ ($t' \leq t$), we will use CNOT gates to copy the qubits $x_{i_1}, ..., x_{i_{t'}}$ from the work and the copy registers to qubit j. We just need to allocate these CNOT gates evenly to make the overall depth small. This step can be divided into $\left\lceil \frac{2^t}{t\lfloor m/(2t)\rfloor} \right\rceil$ mini-steps, each mini-step handling $t\left\lfloor \frac{m}{2t} \right\rfloor$ qubits j by assigning the state $|\langle s(j,1), x\rangle\,\rangle$ to it. Since we have $\ell = 2^t$ qubits to handle, it needs $\left\lceil \frac{2^t}{t\lfloor m/(2t)\rfloor} \right\rceil$ mini-steps.

For all positions $i \in [t]$ with $s(j,1)_i = 1$, we use CNOT to copy x_i to qubit j. We have t variables $x_1, \ldots, x_t$, each with $\lfloor m/(2t)\rfloor$ copies. To utilize these copies for parallelization, we break the $t\lfloor m/(2t)\rfloor$ target qubits into t blocks of size $\lfloor m/(2t)\rfloor$ each. Each mini-step gives all needed variables for $t(\lfloor \frac{m}{2t} \rfloor + 1)$ qubits j, in depth t. In the first layer, we use the $\lfloor \frac{m}{2t} \rfloor$ copies of x_1 as control qubits in CNOT to copy x_1 to the first block of target qubits j, use the $\lfloor \frac{m}{2t} \rfloor$ copies of x_2 for the second block of target qubits, and so on, to x_t for the t-th block. Then in the second layer, we repeat the above process with a circular shift: Copy x_1 to block 2, x_2 to block 3, ..., x_{t-1} to block t, and x_t to block 1. Repeat this and we can complete this mini-step in depth t, such that $t\lfloor \frac{m}{2t} \rfloor$ many qubits j get their needed variables.

Since there are $\left\lceil \frac{2^t}{t\lfloor m/(2t)\rfloor} \right\rceil$ mini-steps, each of depth t, the total depth for U_1 is $\left\lceil \frac{2^t}{t\lfloor m/(2t)\rfloor} \right\rceil \cdot t \leq \frac{m/2}{m/(2t)} + t = 2t = 2\lfloor \log(m/2) \rfloor \leq 2\log m - 2$.

The rotations in the second step are on different qubits and thus can be put into one layer, thus the overall depth for Gray Initial Stage is at most $2\log m$.

The size of this stage is at most $(n+1)m/2$, because each qubit in the phase register has at most n CNOT gates and one R_z gate on it. □

Stage 3: Suffix Copy In this stage, we first undo $U_{copy,1}$, and then make $\left\lfloor \frac{m}{2(n-t)} \right\rfloor$ copies for each of the suffix variables, namely $x_{t+1}, ..., x_n$. The next lemma is similar to Lemma 14 and we omit the proof.

Lemma 17. *We can make* $\left\lfloor \frac{m}{2(n-t)} \right\rfloor$ *copies of each qubit* $x_{t+1}, x_{t+2}, \cdots, x_n$ *in the input register and the copy register, by applying on* $|x\rangle\,|0^{m/2}\rangle$ *an* m*-size circuit* $U_{copy,2}$ *of CNOT gates only, in depth at most* $\log m$*.*

Define

$$|x_{suf}\rangle \overset{\text{def}}{=} |\overbrace{\underbrace{x_{t+1}\cdots x_{t+1}}_{\left\lfloor \frac{m}{2(n-t)} \right\rfloor \text{ qubits}} \cdots \underbrace{x_n \cdots x_n}_{\left\lfloor \frac{m}{2(n-t)} \right\rfloor \text{ qubits}} \;\; 0\cdots 0}^{m/2 \text{ qubits}}\rangle,$$

then the effect of $U_{copy,2}$ is

$$|x\rangle\,|0^{m/2}\rangle \xrightarrow{U_{copy,2}} |x\rangle\,|x_{suf}\rangle\,.$$

The operator of this stage is $U_{copy,2}U_{copy,1}^{\dagger}$, and the depth is at most $2\log m$ and the size is at most m. The effect of this stage $U_{copy,2}U_{copy,1}^{\dagger}$ is

$$|x\rangle\,|x_{pre}\rangle \xrightarrow{U_{copy,1}^{\dagger}} |x\rangle\,|0^{m/2}\rangle \xrightarrow{U_{copy,2}} |x\rangle\,|x_{suf}\rangle\,. \tag{11}$$

Stage 4: Gray Path This stage contains $2^n/\ell - 1$ phases, indexed by $k = 2, 3, \ldots, 2^n/\ell$. The previous Gray Initial Stage can be also viewed as the phase $k = 1$. We single it out as a stage because it implements linear functions from scratch, while each phase in the Gray Path Stage implements linear functions only by a small update from the previous phase.

In each phase k in this stage, the circuit has two steps:

1. Step $k.1$ is a unitary circuit U_k that applies a CNOT gate on each qubit $j \in [\ell]$ in the phase register, controlled by $x_{t_{j,(k-1)}}$, the bit where $s(j, k-1)$ and $s(j,k)$ differ.

2. Step $k.2$ applies the rotation gate $R(\alpha_{s(j,k)})$ on qubit j. Put $R_k = \otimes_{j\in[\ell]} R(\alpha_{s(j,k)})$.

Lemma 18. *The phase k of the Gray Path Stage implements*

$$|x\rangle\,|x_{suf}\rangle\,|f_{[\ell],k-1}\rangle \xrightarrow{U_k} |x\rangle\,|x_{suf}\rangle\,|f_{[\ell],k}\rangle \xrightarrow{R_k} e^{i\sum_{j\in[\ell]} f_{j,k}(x)\alpha_{s(j,k)}}\,|x\rangle\,|x_{suf}\rangle\,|f_{[\ell],k}\rangle\,, \tag{12}$$

where $f_{j,k}(x) = \langle s(j,k), x\rangle$ and $|f_{[\ell],k}\rangle = \otimes_{j\in[\ell]}\,|f_{j,k}(x)\rangle$. The depth and size of the whole Gray Path Stage are at most $2\cdot 2^n/\ell$ and 2^{n+1}.

Proof. The operation can be easily seen in a similar way as that for Lemma 16. Next we show the depth bound. The Gray Path stage repeats step $k.1$-$k.2$ for $2^n/\ell - 1$ times. Since $s(j,k-1)$ and $s(j,k)$ differ by only 1 bit by Lemma 15, one CNOT gate suffices to implement the function $\langle x, s(j,k)\rangle$ from $\langle x, s(j,k-1)\rangle$ in the previous phase: The control qubit is $x_{t_{j,(k-1)}}$ and the target qubit is j. Moreover, the third property in Lemma 15 shows that each variables x_i is used as a control qubit for at most $\left(\left\lfloor \frac{m}{2(n-t)}\right\rfloor + 1\right)$ different $j\in[\ell]$. Since we have $\left(\left\lfloor \frac{m}{2(n-t)}\right\rfloor + 1\right)$ copies in the input register and the copy register, these CNOT gates in step $k.1$ can be implemented in depth 1.

The step $k.2$ consists of only single qubit gates, which can be all paralleled in depth 1. Thus the total depth of Gray Path stage is at most $2^n/\ell\cdot(1+1)\le 2\cdot 2^n/\ell$.

The size of this stage is 2^{n+1} since each linear combination of input variables is generated once and applied single-qubit phase-shift gates R_k. The number of linear combinations of input variables is 2^n , so the size is 2^{n+1}. □

Stage 5: Inverse In this stage, the circuit applies $U^\dagger_{copy,1}U_1^\dagger U_{copy,1}U^\dagger_{copy,2}U_2^\dagger\cdots U^\dagger_{2^n/\ell}$.

Lemma 19. *The depth and size of the Inverse Stage are at most $O(\log m + 2^n/m)$ and $\frac{m}{2}+\frac{nm}{2}+m+2^n = 2^n+\frac{3m+nm}{2}$. The effect of this stage is*

$$|x\rangle\,|x_{suf}\rangle\,|f_{[\ell],2^n/\ell}\rangle \xrightarrow{U_{Inverse}} |x\rangle\,|0^{m/2}\rangle\,|0^{m/2}\rangle\,. \tag{13}$$

The proof of Lemma 19 is shown in Appendix F.

Putting things together After explaining all the five stages, we are ready to put them together to see the overall depth and operation of the circuit.

Lemma 20. *The circuit implements the operation in Eq.* (3) *in depth $O(\log m + 2^n/m)$ and in size $3\cdot 2^n + nm + \frac{7}{2}m$.*

The proof of Lemma 20 is shown in Appendix G. In summary, Λ_n can be implemented in $O\Big(\log m + \frac{2^n}{m}\Big)$ depth and size $3\cdot 2^n + nm + \frac{7}{2}m$ with $m\in[2n, 2^n/n]$ ancillary qubits, proving Lemma 10.

5 Diagonal unitary implementation without ancillary qubits

In this section, we prove Lemma 11. That is, any diagonal unitary $\Lambda_n\in\mathbb{C}^{2^n\times 2^n}$ as in Eq. (3) can be implemented by a quantum circuit of depth $O(2^n/n)$ and size $O(2^n)$ without ancillary qubits. In Section 5.1, we present the framework of our circuit and the functionalities of the operators inside. We then prove the correctness and analyze the depth of our circuit in Section 5.2. Finally, we give the detailed construction of some operators in Section 5.3.

5.1 Framework and functionalities

The framework of our circuit implementing Λ_n is a recursive procedure shown in Figure 4.

The n-qubit work register is divided into two registers: A *control register* consisting of the first r_c qubits, and a *target register* consisting of the last r_t qubits. The circuit has the following components.

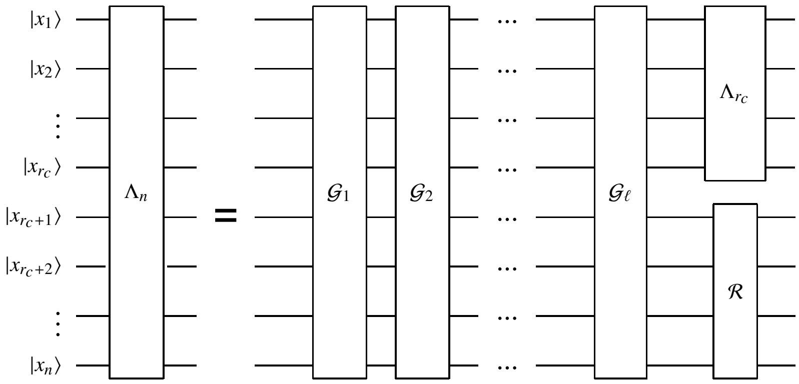


Figure 4: A circuit framework to implement an n-qubit unitary diagonal matrix Λ_n, where $r_t = \lfloor n/2 \rfloor$, $r_c = n - r_t = \lceil n/2 \rceil$ and $\ell \leq \frac{2^{r_t+2}}{r_t+1} - 1$. The first r_c qubits are control register and the last r_t qubits are target register. The depth of the operator $\mathcal{G}_k$ is $O(2^{r_c})$ for each $k \in [\ell]$ and the depth of the operator $\mathcal{R}$ is $O(r_t/\log r_t)$

. The r_c-qubit diagonal unitary matrix Λ_{r_c} is implemented recursively.

1. A sequence of n-qubit unitary operators $\mathcal{G}_1, \ldots, \mathcal{G}_\ell$, the detailed construction of which will be given in Section 5.3.

2. An r_t-qubit unitary operator $\mathcal{R}$, which resets the state in the target register to the input value $|x_{r_c+1}, \ldots, x_n\rangle$.

3. An r_c-qubit diagonal unitary operator Λ_{r_c}, which is implemented recursively.

The parameters are set as follows: $r_t = \lfloor n/2 \rfloor \approx n/2, \quad r_c = n - r_t \approx n/2, \quad$ and $\quad \ell \leq \frac{2^{r_t+2}}{r_t+1} - 1 \approx \frac{2^{n/2+3}}{n}$.

Next we describe the function of each operator in Figure 4, for which it suffices to specify their effects on an arbitrary computational basis state

$$|x\rangle = |x_1 x_2 \cdots x_{r_c} x_{r_c+1} \cdots x_n\rangle = \underbrace{|x_{control}\rangle}_{r_c \text{ qubits}} \underbrace{|x_{target}\rangle}_{r_t \text{ qubits}},$$

where $x \in \{0,1\}^n$. Let us first highlight some key similarities and differences between this circuit and the one presented in the previous section. Recall that in Section 4, an n-bit string $s \in \{0,1\}^n - \{0^n\}$ is broken into two parts, a $\left\lfloor \log(\frac{m}{2}) \right\rfloor$-bit prefix and an $\left(n - \lfloor \log(\frac{m}{2}) \rfloor\right)$-bit suffix. In the Gray Initial Stage there, we use $2^{\lfloor \log(m/2) \rfloor}$ qubits in the phase register to enumerate all possible $\lfloor \log(m/2) \rfloor$-bit prefixes, one prefix on each phase qubit j. Then on each such qubit j we enumerate all $(n - \lfloor \log(m/2) \rfloor)$-bit suffixes in the Gray Path Stage. In this section, we again break s into a prefix and a suffix, and enumerate all prefixes and all suffixes to run over all n-bit strings. However, due to the lack of the ancillary qubits, the circuit here differs from the last one in the following two aspects.

1. In Section 4, $s \in \{0,1\}^n - \{0^n\}$ is generated in the phase register, which is initialized to $|0\rangle$. In this section, $s = ct$, in which c is the r_c-bit prefix and t is the r_t-bit suffix. The state $|\langle s, x\rangle\rangle$ is generated in target register, whose initial state is $|x_j\rangle$ for some $j \in \{r_c+1, r_c+2, \ldots, r_n\}$. Hence, we enumerate s recursively in this section. That is, we first generate $s = ct$ for $t \neq 0^{r_t}$ and then generate $c0^{r_t}$ recursively.

2. In Section 4, there are $2^{\lfloor \log(m/2) \rfloor}$ $(\leq \frac{m}{2})$ prefixes which can be enumerated in $\frac{m}{2}$ qubits in phase register exactly. In this section, $2^{r_t} - 1$ $(\approx 2^{n/2})$ suffixes should be generated in r_t qubits in target register. As we only have r_t qubits, the small space is insufficient to enumerate all $2^{r_t} - 1$ suffixes. Thus we need to enumerate them in many stages, and r_t suffixes in each stage; in other words, we pay the price of time to compensate the shortage of space. It turns out that the transition from one stage to another can be made in a low depth if the suffixes enumerated in each stage

are linearly independent as vectors in $\{0,1\}^{r_t}$. Thus we need carefully divide $2^{r_t}-1$ suffixes into ℓ sets $T^{(1)},\dots,T^{(\ell)}$ with $T^{(k)}=\{t_1^{(k)},t_2^{(k)},\dots,t_{r_t}^{(k)}\}$ each $t_a^{(k)}\neq 0^{r_t}$ for $a\in[r_t]$ and $k\in[\ell]$, and the strings in each $T^{(k)}$ linearly independent. We allow overlap between these sets, but maintain the total number ℓ of sets only a constant times of $(2^{r_t}-1)/r_t$, so that the overall depth is still under the control. As the sets have overlaps, a suffix may appear more than once, so we need to note this and avoid repeatedly applying rotation when the suffix appears multiple times.

We now show how to implement the above high-level ideas. We will need to find sets $T^{(1)},T^{(2)},\dots,T^{(\ell)}$ satisfying the following two key properties.

1. For each $k\in[\ell]$, the set $T^{(k)}=\left\{t_1^{(k)},t_2^{(k)},\dots,t_{r_t}^{(k)}\right\}$ contains r_t vectors from $\{0,1\}^{r_t}$ that are linearly independent over the field $\mathbb{F}_2$.

2. The collection of these sets covers all the r_t-bit strings except for 0^{r_t}, i.e. $\bigcup_{k\in[\ell]}T^{(k)}=\{0,1\}^{r_t}-\{0^{r_t}\}$.

The constructions of sets $T^{(1)},\dots,T^{(\ell)}$ are shown in Appendix H. For each $k\in[\ell]\cup\{0\}$, define an r_t-qubit state

$$|y^{(k)}\rangle=|y_1^{(k)}y_2^{(k)}\cdots y_{r_t}^{(k)}\rangle,\ \text{where}\quad y_j^{(k)}=\begin{cases}x_{r_c+j} & \text{if } k=0,\\ \langle 0^{r_c}t_j^{(k)},x\rangle & \text{if } k\in[\ell].\end{cases}\tag{14}$$

Namely, $y^{(0)}$ is the same as x_{target} (the suffix of x), and other $y_j^{(k)}$ are linear functions of variables in x_{target} with coefficients given by $t_j^{(k)}$. Next, let us define disjoint families $F_1,\dots,F_\ell$ which apply the rotation when a suffix appears for the first time.

$$\begin{aligned}F_1&=\left\{ct:\ t\in T^{(1)},c\in\{0,1\}^{r_c}\right\},\\ F_k&=\left\{ct:\ t\in T^{(k)},c\in\{0,1\}^{r_c}\right\}-\bigcup_{d\in[k-1]}F_d,\ 2\le k\le\ell.\end{aligned}\tag{15}$$

These families of sets $F_1,F_2,\cdots,F_\ell$ satisfy $F_i\cap F_j=\emptyset$ for all $i\neq j\in[\ell]$ and

$$\bigcup_{k\in[\ell]}F_k=\{0,1\}^{r_c}\times\bigcup_{k\in[\ell]}T^{(k)}=\{0,1\}^{r_c}\times(\{0,1\}^{r_t}-\{0^{r_t}\})=\{0,1\}^n-\{c0^{r_t}:\ c\in\{0,1\}^{r_c}\}.\tag{16}$$

With the above concepts, we can now show the desired effect of the operators $\mathcal{G}_k$, $\mathcal{R}$ and Λ_{r_c}.

1. For $k\in[\ell]$,

$$\mathcal{G}_k\,|x_{control}\rangle\,|y^{(k-1)}\rangle=e^{i\sum_{s\in F_k}\langle s,x\rangle\alpha_s}\,|x_{control}\rangle\,|y^{(k)}\rangle\,,\tag{17}$$

where α_s is determined by Eq. (6). In words, $\mathcal{G}_k$ has two effects: (1) It puts a phase and (2) it transits from the stage $k-1$ to the stage k.

2. The transformation $\mathcal{R}$ acts on the target register and resets the suffix state as follows

$$\mathcal{R}\,|y^{(\ell)}\rangle=|y^{(0)}\rangle\,.\tag{18}$$

As a map on $\{0,1\}^{r_t}$ (instead of $\{|x\rangle:x\in\{0,1\}^{r_t}\}$), $\mathcal{R}$ is an invertible linear transformation over $\mathbb{F}_2$.

3. The operator Λ_{r_c} is an r_c-qubit diagonal matrix satisfying that

$$\Lambda_{r_c}\,|x_{control}\rangle=e^{i\sum_{c\in\{0,1\}^{r_c}-\{0^{r_c}\}}\langle c0^{r_t},x\rangle\alpha_{c0^{r_t}}}\,|x_{control}\rangle\,,\tag{19}$$

and will be implemented recursively.

We will define these operators and show these properties in Section 5.3.

5.2 Correctness and depth

In this section, we will prove the correctness and analyze the depth of the circuit. We will need a fact about the depth of invertible linear transformation from [52] (Theorem 1). The original version says that any CNOT circuit, a circuit consisting of only CNOT gates, on n qubits can be compressed into $O(n/\log n)$ depth. But note that any n-dimensional invertible linear transformation over $\mathbb{F}_2$ can be implemented by a CNOT circuit [53]. We thus have the following result.

Lemma 21. *Suppose that $U \in \mathbb{F}_2^{n\times n}$ is an invertible linear transformation over $\mathbb{F}_2$. Then as a $2^n \times 2^n$ unitary matrix which permutes computational basis $\{|x\rangle : x \in \{0,1\}^n\}$, the map U can be realized by a CNOT circuit of depth at most $O(\frac{n}{\log n})$ and size at most $O(\frac{n^2}{\log n})$ without ancillary qubits.*

As mentioned in Section 5.1, $\mathcal{R}$ is an invertible linear transformation on the computational basis variables, thus the above lemma immediately implies the following depth upper bounds for $\mathcal{R}$.

Lemma 22. *The operator $\mathcal{R}$ can be realized by an $O(\frac{r_t}{\log r_t})$-depth and $O(\frac{r_t^2}{\log r_t})$-size CNOT circuit without ancillary qubits.*

The depth of $\mathcal{G}_k$ will be easily seen from its construction in Section 5.3.

Lemma 23. *The operator $\mathcal{G}_k$ can be realized by an $O(2^{r_c})$-depth and $O(r_c 2^{r_c+1})$-size quantum circuit using single-qubit and CNOT gates without ancillary qubits.*

Now we are ready to prove the correctness and depth of the whole circuit. The correctness of the circuit framework in Figure 4 is shown in Appendix I.

Lemma 24. *Any diagonal unitary matrix Λ_n can be realized by the quantum circuit $(\Lambda_{r_c}\otimes\mathcal{R})\mathcal{G}_\ell\mathcal{G}_{\ell-1}\cdots\mathcal{G}_1$ as in Figure 4, which has depth $O(2^n/n)$ and size $2^{n+3} + O\left(\frac{n^2}{\log n}\right)$ and uses no ancillary qubits.*

Proof. We prove that the circuit has depth $D(n) = O(2^n/n)$. Lemma 23 shows $\mathcal{G}_k$ can be realized in depth at most $\lambda_1 \cdot 2^{r_c}$ for a constant $\lambda_1 > 0$ and Lemma 22 shows $\mathcal{R}$ can be implemented in depth at most $\lambda_2 \cdot \frac{r_t}{\log r_t}$ without ancillary qubits for a constant $\lambda_2 > 0$. Therefore, $D(n)$ satisfies the following recurrence

$$\begin{aligned} D(n) &\le \max\left\{D(r_c),\ \lambda_2\cdot\tfrac{r_t}{\log r_t}\right\} + \lambda_1\cdot 2^{r_c}\cdot\ell \\ &\le D(\lceil n/2\rceil) + \tfrac{\lambda_2\lceil n/2\rceil}{\log\lceil n/2\rceil} + \lambda_1 2^{\lceil n/2\rceil}\left(\tfrac{2^{\lfloor n/2\rfloor+2}}{\lfloor n/2\rfloor+1} - 1\right) \\ &= D(\lceil n/2\rceil) + O(2^n/n). \end{aligned}$$

Solving the above recursive relation, we obtain the bound $D(n) = O(2^n/n)$ as desired. The size of this circuit $S(n)$ satisfies $S(n) \le S(n/2) + (2^{n+3} - 2^{n/2+3}) + O\left(\frac{n^2}{\log n}\right) \le 2^{n+3} + O\left(\frac{n^2}{\log n}\right)$. □

5.3 Construction of $\mathcal{G}_k$ and $\mathcal{R}$

In this section, we will show how to construct operator $\mathcal{G}_k$, which consists of two stages: Generate Stage and Gray Path Stage, see Figure 5. Along the way, we will also show the construction of $\mathcal{R}$.

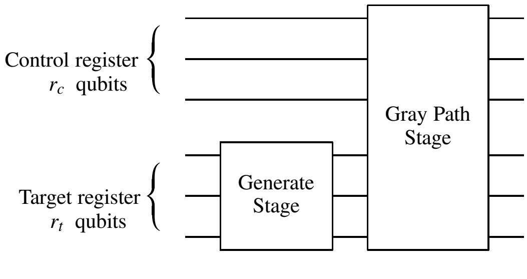


Figure 5: Implementation of operator $\mathcal{G}_k$, which consists of Generate Stage and Gray Path Stage. The depth of Generate Stage is $O\left(\frac{r_t}{\log r_t}\right)$ and the depth of Gray Path Stage is $O(2^{r_c})$.

Generate Stage In this stage, we implement operator $U_{Gen}^{(k)}$, such that

$$|y^{(k-1)}\rangle \xrightarrow{U_{Gen}^{(k)}} |y^{(k)}\rangle, \ k \in [\ell], \tag{20}$$

where $y^{(k-1)}$ and $y^{(k)}$ are defined in Eq. (14) and determined by $T^{(k-1)}$ and $T^{(k)}$, respectively. For $k \in [\ell]$, recall that $T^{(k)} = \{t_1^{(k)}, \cdots, t_{r_t}^{(k)}\}$. Fix this ordering, view each $t_i^{(k)}$ as a column vector, and define a matrix $\hat{T}^{(k)} = [t_1^{(k)}, \cdots, t_{r_t}^{(k)}]^T \in \{0,1\}^{r_t \times r_t}$ for $k \in [\ell]$, with special case $\hat{T}^{(0)} \overset{\text{def}}{=} I_{r_t}$. Then the vectors $y^{(k)}$ can be rewritten as

$$y^{(k)} = \hat{T}^{(k)} x_{target}, \quad \forall k \in [r_t] \cup \{0\}. \tag{21}$$

Since $t_1^{(k)}, t_2^{(k)}, \cdots, t_{r_t}^{(k)}$ are linearly independent over $\mathbb{F}_2$, $\hat{T}^{(k)}$ is an invertible linear transformation over $\mathbb{F}_2$. Now define a unitary $U_{Gen}^{(k)}$ by $U_{Gen}^{(k)} |y\rangle = |\hat{T}^{(k)}(\hat{T}^{(k-1)})^{-1} y\rangle$, where the matrix-vector multiplication at the right hand side is over $\mathbb{F}_2$. From Eq. (21), we see that

$$U_{Gen}^{(k)} |y^{(k-1)}\rangle = |\hat{T}^{(k)}(\hat{T}^{(k-1)})^{-1} y^{(k-1)}\rangle = |\hat{T}^{(k)} x_{target}\rangle = |y^{(k)}\rangle$$

satisfying Eq. (20). Also note that when viewed as a linear transformation over $\mathbb{F}_2$, $U_{Gen}^{(k)}$ is invertible. Thus according to Lemma 21, the following depth upper bound applies.

Lemma 25. *The Generate Stage unitary $U_{Gen}^{(k)}$ can be realized by an $O\left(\frac{r_t}{\log r_t}\right)$-depth and $O\left(\frac{r_t^2}{\log r_t}\right)$-size CNOT circuit without ancillary qubits.*

Similar to the discussion of $U_{Gen}^{(k)}$, operator $\mathcal{R}$ can be defined by $\mathcal{R}|y\rangle = |(\hat{T}^{(\ell)})^{-1} y\rangle$, then $\mathcal{R}|y^{(\ell)}\rangle = |(\hat{T}^{(\ell)})^{-1} y^{(\ell)}\rangle = |x_{target}\rangle = |y^{(0)}\rangle$. Thus $\mathcal{R}$ can be also viewed an invertible linear transformation over $\mathbb{F}_2$. Applying Lemma 21 gives the bound in Lemma 22.

Gray Path Stage This stage implements the following operator

$$|x_{control}\rangle |y^{(k)}\rangle \xrightarrow{U_{GrayPath}} e^{i \sum_{s \in F_k} \langle s,x \rangle \alpha_s} |x_{control}\rangle |y^{(k)}\rangle, \tag{22}$$

where $k \in [\ell]$ and F_k is defined in Eq. (15). The Gray Path Stage in this section is similar to the Gray Path Stage in Section 4, though we need to use a Gray code cycle here instead of a Gray code path. For every $i \in [r_t]$, let $c_1^i, c_2^i, \cdots, c_{2^{r_c}-1}^i, c_{2^{r_c}}^i$ denote the i-Gray code of r_c bits starting at $c_1^i = 0^{r_c}$ for $i \in [r_t]$. Let h_{ij} denote the index of the bit that c_{j-1}^i and c_j^i differ for each $j \in \{2, 3, \ldots, 2^{r_c}\}$ and h_{i1} the index of the bit that c_1^i and $c_{2^{r_c}}^i$ differ. For the i-Gray code cycle of r_c bits,

$$h_{ij} = \begin{cases} (r_c + i - 2 \mod r_c) + 1, & \text{if } j = 1 \\ (\zeta(j-1) + i - 2 \mod r_c) + 1, & \text{if } j \neq 1 \end{cases} \tag{23}$$

The exact form of h_{ij} is not crucial; the important fact to be used later is that the indices $h_{1p}, h_{2p}, \ldots, h_{r_t p}$ are all different.

This stage consists of $2^{r_c} + 1$ phases.

1. In phase 1, circuit C_1 applies a rotation $R(\alpha_{0^{r_c} t_i^{(k)}})$ on the i-th qubit in the target register for all $i \in [r_t]$ if the string $0^{r_c} t_i^{(k)} \in F_k$, where $\alpha_{0^{r_c} t_i^{(k)}}$ is defined in Eq. (6).
2. In phase $p \in \{2, \ldots, 2^{r_c}\}$, circuit C_p consists of 2 steps:
   (a) Step $p.1$ is a unitary that, for all $i \in [r_t]$, applies a CNOT gate on the i-th qubit in target register, controlled by the h_{ip}-th qubit in control register.
   (b) Step $p.2$ is a unitary that, for all $i \in [r_t]$, applies a rotation $R(\alpha_{c_p^i t_i^{(k)}})$ on the i-th qubit in target register if $c_p^i t_i^{(k)} \in F_k$, where $\alpha_{c_p^i t_i^{(k)}}$ is defined in Eq. (6).

3. In phase $2^{r_c}+1$, circuit $C_{2^{r_c}+1}$ implements a unitary that, for all $i \in [r_t]$, applies a CNOT gate on the i-th qubit in target register, controlled by the h_{i1}-th qubit in control register .

The next lemma gives the correctness and depth of this constructed circuit. The proof of Lemma 26 is shown in Appendix J.

Lemma 26. *The quantum circuit defined above is of depth $O(2^{r_c})$ and size $O(r_c 2^{r_c+1})$, and implements Gray Path Stage $U_{GrayPath}$ in Eq.* (22).

According to Lemma 25 and Lemma 26, operator $\mathcal{G}_k$ can be implemented in depth $O(2^{r_c})+O(\frac{r_t}{\log r_t}) = O(2^{r_c})$. And the size of the circuit is at most $O(\frac{n^2}{\log n}) + r_c 2^{r_c+1} = O(r_c 2^{r_c+1})$. This completes the proof of Lemma 23.

6 Quantum state preparation with $\Omega(2^n/n^2)$ ancillary qubits

In this section, we will introduce a different framework that can improve the upper bound in Section 4 when the number of ancillary qubits $m = \Omega(2^n/n^2)$. In Section 6.1, we will present the framework, and in Section 6.2, we will give implementation details with the depth and correctness analyzed.

In the following, we will use $e_i \in \{0,1\}^{2^n}$ to denote the vector where the i-th bit is 1 and all other bits are 0. It is a unary encoding of $i \in \{0,1,\dots,2^n-1\}$, and $|e_i\rangle$ is the corresponding 2^n-qubit state. We use n-qubit state $|i\rangle = |i_0 i_1 \cdots i_{n-1}\rangle \in (\{|0\rangle, |1\rangle\})^{\otimes n}$ to denote the binary encoding of i, where $i_0, \cdots, i_{n-1} \in \{0,1\}$ and $i = \sum_{j=0}^{n-1} i_j \cdot 2^j$.

6.1 New framework for quantum state preparation

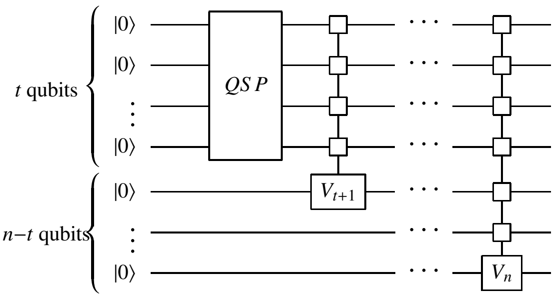


Figure 6: A new circuit framework to prepare an n-qubit quantum state $|\psi_v\rangle$ with $m \in [\Omega(2^n/n^2), 3\cdot 2^n]$. Let $t = \lfloor \log(m/3) \rfloor$. The first t-qubit unitary QSP implement the same transformation as the first t UCGs in Figure 2(a). The last $n-t$ UCGs in are the same as the last $n-t$ UCGs in Figure 2(a).

The quantum circuit in Section 2 for quantum state preparation consists of n UCGs $V_1, V_2, \dots, V_n$ (Figure 2(a)). In Section 4, we showed that any j-qubit UCG V_j can be implemented by a quantum circuit of depth $O\left(j + \frac{2^j}{m+j}\right)$ with m ancillary qubits. Summing this up over $j \in [n]$ gives the $O(n^2 + 2^n/m)$ upper bound for QSP, and this quadratic term seems hard to be improved within the framework of [22]. In the new framework, we first generate the quantum state in the unary encoding $\sum_i v_i |e_i\rangle$ using the result in [30], and then make an encoding transform $|e_i\rangle \to |i\rangle$, from the unary encoding to the binary encoding.

Two issues need to be handled here. The first one is the need to design an encoding transform circuit that has small depth and size, using ancillary qubits efficiently. We will give an optimal construction in Section 6.2. The second issue is that the unary encoding itself needs 2^n qubits, and the encoding transform also needs $O(2^n)$ qubits, which may be beyond m, the number of ancillary qubits that are available in the first place. To handle this, we will use a hybrid method. We break the generation into a prefix part and a suffix part, where the length of the prefix is whatever m can support. We prepare the prefix part by unary QSP construction in [30] and our encoding transformation, and then employ the methods in Section 4 for the suffix part.

Our new circuit framework for QSP in the parameter regime $m = \Omega(2^n/n^2)$ is shown in Figure 6. Let $t = \lfloor \log(m/3) \rfloor$. In previous framework, the first t UCGs are a QSP circuit to prepare a t-qubit quantum state $|\psi_v^{(t)}\rangle = \sum_{k=0}^{2^t} v'_k |k\rangle$, where $v'_k = \sqrt{\sum_{j=0}^{2^{n-t}-1} |v_{2^{n-t}k+j}|^2}$. In the new framework, we introduce a new t-qubit QSP circuit to replace the first t UCGs. The new QSP circuit consists of the following steps.

1. Generate a 2^t-qubit quantum state $|\psi'_v\rangle = \sum_{k=0}^{2^t-1} v'_k |e_k\rangle$, where $e_k \in \{0,1\}^{2^t}$ and by the quantum circuit in [30].

2. Applying U_t to $|\psi'_v\rangle$, we can obtain $|\psi_v^{(t)}\rangle$ with $2m/3$ ancillary qubits, where U_t is the unitary transformation $U_t : |e_i\rangle \to |i\rangle |0^{2^t-t}\rangle$ for all $i \in \{0\} \cup [2^t - 1]$.

3. Realize the last $n - t$ UCGs by Eq. (4) and Lemma 10.

6.2 Implementation and analysis

Now we give a more detailed implementation and analyze the correctness and cost of the algorithm. First, in [30] it is shown that QSP with the unary encoding can be implemented efficiently.

Lemma 27. *Given a vector $v = (v_0, v_1, \ldots, v_{2^n-1})^T \in \mathbb{C}^{2^n}$ with unit ℓ_2-norm, any 2^n-qubit quantum state $|\psi'_v\rangle = \sum_{k=0}^{2^n-1} v_k |e_k\rangle$ can be prepared from the initial state $|0\rangle^{\otimes 2^n}$ by a quantum circuit using single-qubit gates and CNOT gates of depth $O(n)$ and size $O(2^n)$ without ancillary qubits.*

Next we consider the encoding transformation.

Lemma 28. *The following unitary transformation on 2^n qubits*

$$|e_i\rangle \to |i\rangle |0^{2^n-n}\rangle, \forall i \in \{0\} \cup [2^n - 1], e_i \in \{0,1\}^{2^n}, \tag{24}$$

can be implemented by a quantum circuit using single-qubit gates and CNOT gate with 2^{n+1} ancillary qubits, of depth $O(n)$ and size $O(2^n)$.

The proof of Lemma 28 is shown in Appendix K. Now we are ready to give the hybrid algorithm and cost analysis.

Lemma 29. *For any $m \in [\Omega(2^n/n^2), 3 \cdot 2^n]$, any n-qubit quantum state $|\psi_v\rangle$ can be generated by a quantum circuit, using single-qubit gates and CNOT gates, of depth $O\Big(n(n - \log(m/3) + 1) + \frac{2^n}{m}\Big)$ and size $O(2^n)$ with m ancillary qubits.*

Proof. Let $t = \lfloor \log \frac{m}{3}) \rfloor$. Define a quantum state $|\psi_v^{(t)}\rangle = \sum_{i=0}^{2^t-1} v'_i |i\rangle$, where $v'_i = \sqrt{\sum_{j=0}^{2^{n-t}-1} |v_{i \cdot 2^{n-t}+j}|^2}$. Note that $|\psi_v^{(t)}\rangle = V_t V_{t-1} \cdots V_1 |0\rangle^{\otimes n}$, the state after we apply the first t UCGs in Figure 2(a).

According to Lemma 27, given the unit vector $v' = (v'_0, \ldots, v'_{2^t-1})$, we can prepare a 2^t-qubit quantum state $|\psi'_v\rangle = \sum_{i=0}^{2^t-1} v'_i |e_i\rangle$ by a quantum circuit of depth $O(t) = O(n)$ and size $O(2^t) = O(2^n)$. The resulting state is on 2^t qubits. Then we apply the unitary transform Eq. (24) in Lemma 28 to transform the unary encoding to a binary encoding and obtain $|\psi_v^{(t)}\rangle = \sum_{i=0}^{2^t-1} v'_i |i\rangle$. This transformation has depth $O(t)$ and size $O(2^t)$, and need 2^{t+1} ancillary qubits. The whole process can be carried out in a work space of $2^t + 2^{t+1} \leq m$ qubits.

To change $|\psi_v^{(t)}\rangle$ to the final target state $|\psi_v\rangle$, what is left is to apply $V_{t+1}, \ldots, V_n$ to $|\psi_v^{(t)}\rangle$. By Lemma 12, each V_j can be implemented by a circuit of depth $O(j + \frac{2^j}{m})$ and size $O(2^j)$ by m ancillary qubits. Hence $V_n \cdots V_{t+1}$ can be realized by a quantum circuit of depth $\sum_{j=t+1}^{n} O\Big(j + \frac{2^j}{m}\Big) = O\Big(n(n - \lfloor \log(m/3) \rfloor) + \frac{2^n}{m}\Big)$, and size $\sum_{j=t+1}^{n} O(2^j) = O(2^n)$, with m ancillary qubits.

Combining the two steps, we see that the total depth and size of this quantum state preparation circuit are $O\Big(n(n - \log(m/3) + 1) + \frac{2^n}{m}\Big)$ and $O(2^n)$, respectively. □

Note that when $m = 3 \cdot 2^n$, the depth bound becomes $O(n)$. And if m is even larger, then we can choose to only use $3 \cdot 2^n$ of them. Thus we have the following result, which is Theorem 1 in the parameter regime $m = \Omega(2^n/n^2)$.

Corollary 30. *For a circuit preparing an n-qubit quantum state with $m = \Omega(2^n/n^2)$ ancillary qubits, the minimum depth $D_{\mathrm{QSP}}(n,m)$ for different ranges of m are characterized as follows:*

$$\begin{cases} O(2^n/m), & \text{if } m \in [\Omega(2^n/n^2), O(2^n/(n\log n))], \\ O(n\log n), & \text{if } m \in [\omega(2^n/(n\log n)), o(2^n)], \\ O(n), & \text{if } m = \Omega(2^n). \end{cases}$$

7 Extensions and implications

7.1 Implications on optimality of unitary depth compression

In this section, we will show that our results for QSP can be applied to general unitary synthesis. The proofs of Theorem 31 and Corollary 32 are shown in Appendix L.

Theorem 31. *Any unitary matrix $U \in \mathbb{C}^{2^n \times 2^n}$ can be implemented by a quantum circuit of depth $O\Big(n2^n + \frac{4^n}{m+n}\Big)$ and size $O(4^n)$ with $m \le 2^n$ ancillary qubits.*

In [39], it was shown that one needs at least $\Omega\left(4^n\right)$ CNOT gates to implement an arbitrary n-qubit unitary matrix without ancillary qubits. In the proof, the authors first put the circuit in a form that all single-qubit gates are immediately before either a CNOT gate or the output. It is known that such a CNOT gate together with its two single-qubit incoming neighbor gates can be specified by 4 free real parameters, and that each single-qubit gate right before the output has 3 free real parameters. Thus overall the circuit has $4k + 3n$ parameters where k is the number of CNOT gates. To generate all n-qubit states, the set of which is known to have dimension $4^n - 1$, we need $4k + 3n \ge 4^n - 1$. Thus the bound follows. This argument basically applies to quantum circuits with ancillary qubits as well, as stated in the next corollary, which shows that our circuit construction for general unitary matrices is asymptotically optimal for $m = O(2^n/n)$.

Corollary 32. *The minimum circuit depth $D_{\mathrm{UNITARY}}(n,m)$ for an arbitrary n-qubit unitary with m ancillary qubits satisfies*

$$\begin{cases} D_{\mathrm{UNITARY}}(n,m) = \Theta\Big(\frac{4^n}{m+n}\Big), & \text{if } m = O(2^n/n), \\ D_{\mathrm{UNITARY}}(n,m) \in \left[\Omega(n), O(n2^n)\right], & \text{if } m = \omega(2^n/n). \end{cases}$$

7.2 Decomposition with Clifford + T gate set

The quantum gate set $\{CNOT, H, S, T\}$, sometimes called Clifford+T gate set, is a universal gate set in that any unitary matrix can be approximately implemented using these gates only. The gates in this set all have a fault-tolerant implementation, thus the gate set is considered as one of the most promising candidates for practical quantum computing. In this section we consider the circuits using only the gates in this set.

Definition 33 (ϵ-approximation)**.** *For any $\epsilon > 0$, a unitary matrix U is ϵ-approximated by another unitary matrix V if*

$$\|U - V\|_2 \overset{\text{def}}{=} \max_{\||\psi\rangle\|_2 = 1} \|(U - V)|\psi\rangle\|_2 < \epsilon.$$

We can extend our results on the exact implementations of state preparation and unitary to their approximate versions.

The following two corollaries are circuit implementations for quantum state preparation (Corollary 34) and unitary synthesis (Corollary 35). The Corollary 34 is a restatement of Corollary 6. The proofs are shown in Appendix M.

Corollary 34. *For any n-qubit target state $|\psi_v\rangle$ and $\epsilon > 0$, one can prepare a state $|\psi'_v\rangle$ which is ϵ-close to $|\psi_v\rangle$ in ℓ_2-distance, by a quantum circuit consisting of $\{CNOT, H, S, T\}$ gates of depth*

$$\begin{cases} O\Big(\frac{2^n \log(2^n/\epsilon)}{m+n}\Big) & \text{if } m = O(2^n/(n\log n)), \\ O(n\log n \log(2^n/\epsilon)) & \text{if } m \in [\omega(2^n/(n\log n)), o(2^n)], \\ O(n\log(2^n/\epsilon)) & \text{if } m = \Omega(2^n), \end{cases}$$

where m is the number of ancillary qubits.

The following is an implementation of a unitary matrix.

Corollary 35. *Any n-qubit general unitary matrix can be implemented by a circuit, using the $\{CNOT, H, S, T\}$ gate set, of depth $O\Big(n2^n + \frac{4^n \log(4^n/\epsilon)}{m+n}\Big)$ with m ancillary qubits.*

Remark. Our circuits for general states can be also extended to circuits for sparse states. See details in Appendix N.

8 Conclusion

In this paper, we have shown that an arbitrary n-qubit quantum state can be prepared by a quantum circuit consisting of single-qubit gates and CNOT gates with $m = O(2^n)$ ancillary qubits, of depth $O\Big(n\log n + \frac{2^n}{n+m}\Big)$ and size $O(2^n)$. The bound is improved to $O(n)$ if we have more ancillary qubits, and all these bounds are tight (up to a logarithmic factor in a small range of m). These results can be applied to reduce the depth of the circuit of general unitary to $O\Big(n2^n + \frac{4^n}{m+n}\Big)$ with m ancillary qubits, which is optimal when $m = O(2^n/n)$. The results can be extended to approximate state preparation by circuit using the Clifford+T gate set.

Many questions are left open for future studies. An immediate one is to close the gap for unitary synthesis for large m in Corollary 32. One can also put more practical restrictions into consideration. For instance, we assume that two-qubit gates can be applied on any two qubits. Though this all-to-all connection is indeed the case for certain quantum computer implementations (such qubits made of trapped ions), some others (such as superconducting qubits) can only support nearest neighbor interactions, and it is interesting to study QSP for that case. Another direction is to take various noises into account, and see how much that affects the complexity. We call for more studies of state preparation and circuit synthesis, and hope that methods and techniques developed in this paper can be used to design efficient circuits in those extended models.

References

[1] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[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(9):090502, 2015.

[3] Guang Hao Low and Isaac L Chuang. Optimal hamiltonian simulation by quantum signal processing. *Physical review letters*, 118(1):010501, 2017.

[4] G. H. Low and I. L. Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, 2019.

[5] Dominic W Berry, Andrew M Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 792–809. IEEE, 2015.

[6] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.

[7] Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[8] Patrick Rebentrost, Adrian Steffens, Iman Marvian, and Seth Lloyd. Quantum singular-value decomposition of nonsparse low-rank matrices. *Physical review A*, 97(1):012327, 2018.

[9] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[10] Leonard Wossnig, Zhikuan Zhao, and Anupam Prakash. Quantum linear system algorithm for dense matrices. *Physical review letters*, 120(5):050502, 2018.

[11] Iordanis Kerenidis, Jonas Landman, Alessandro Luongo, and Anupam Prakash. q-means: A quantum algorithm for unsupervised machine learning. *arXiv preprint arXiv:1812.03584*, 2018.

[12] Iordanis Kerenidis and Jonas Landman. Quantum spectral clustering. *arXiv preprint arXiv:2007.00280*, 2020.

[13] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.

[14] Rami Barends, Julian Kelly, Anthony Megrant, Andrzej Veitia, Daniel Sank, Evan Jeffrey, Ted C White, Josh Mutus, Austin G Fowler, Brooks Campbell, et al. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature*, 508(7497):500–503, 2014.

[15] Julian Kelly, Rami Barends, Austin G Fowler, Anthony Megrant, Evan Jeffrey, Theodore C White, Daniel Sank, Josh Y Mutus, Brooks Campbell, Yu Chen, et al. State preservation by repetitive error detection in a superconducting quantum circuit. *Nature*, 519(7541):66–69, 2015.

[16] Chao Song, Kai Xu, Wuxin Liu, Chui-ping Yang, Shi-Biao Zheng, Hui Deng, Qiwei Xie, Keqiang Huang, Qiujiang Guo, Libo Zhang, et al. 10-qubit entanglement and parallel logic operations with a superconducting circuit. *Physical review letters*, 119(18):180511, 2017.

[17] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[18] `https://quantum-computing.ibm.com/`.

[19] Jerry Chow, Oliver Dial, and Jay Gambetta. Ibm quantum breaks the 100-qubit processor barrier. *IBM Research Blog*, 2021.

[20] Jay Gambetta. IBM's roadmap for scaling quantum technology. `https://www.ibm.com/blogs/research/2020/09/ibm-quantum-roadmap/`, 2020.

[21] Lucero Erik. Unveiling our new quantum ai campus. `https://blog.google/technology/ai/unveiling-our-new-quantum-ai-campus/`, 2021.

[22] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002.

[23] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[24] Craig Gidney. https://algassert.com/circuits/2015/06/22/using-quantum-gates-instead-of-ancilla-bits.html. 2015.

[25] Ville Bergholm, Juha J Vartiainen, Mikko Möttönen, and Martti M Salomaa. Quantum circuits with uniformly controlled one-qubit gates. *Physical Review A*, 71(5):052330, 2005.

[26] Martin Plesch and Časlav Brukner. Quantum-state preparation with universal gate decompositions. *Physical Review A*, 83(3):032302, 2011.

[27] M Mottonen and Juha J Vartiainen. Decompositions of general quantum gates. *arXiv preprint quant-ph/0504100*, 2005.

[28] Xiao-Ming Zhang, Man-Hong Yung, and Xiao Yuan. Low-depth quantum state preparation. *arXiv preprint arXiv:2102.07533*, 2021.

[29] Zhicheng Zhang, Qisheng Wang, and Mingsheng Ying. Parallel quantum algorithm for hamiltonian simulation, 2021.

[30] Sonika Johri, Shantanu Debnath, Avinash Mocherla, Alexandros Singk, Anupam Prakash, Jungsang Kim, and Iordanis Kerenidis. Nearest centroid classification on a trapped ion quantum computer. *npj Quantum Information*, 7(1):1–11, 2021.

[31] Iordanis Kerenidis and Alessandro Luongo. Classification of the mnist data set with quantum slow feature analysis. *Physical Review A*, 101(6):062327, 2020.

[32] Danial Dervovic, Mark Herbster, Peter Mountney, Simone Severini, Naïri Usher, and Leonard Wossnig. Quantum linear systems algorithms: a primer. *arXiv preprint arXiv:1802.08227*, 2018.

[33] Zhikuan Zhao, Jack K Fitzsimons, Patrick Rebentrost, Vedran Dunjko, and Joseph F Fitzsimons. Smooth input preparation for quantum and quantum-inspired machine learning. *Quantum Machine Intelligence*, 3(1):1–6, 2021.

[34] Ryan LaRose and Brian Coyle. Robust data encodings for quantum classifiers. *Physical Review A*, 102(3):032420, 2020.

[35] Dorit Aharonov and Yonathan Touati. Quantum circuit depth lower bounds for homological codes. *arXiv preprint arXiv:1810.03912*, 2018.

[36] 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(5):3457, 1995.

[37] Emanuel Knill. Approximation by quantum circuits. *arXiv preprint quant-ph/9508006*, 1995.

[38] Juha J Vartiainen, Mikko Möttönen, and Martti M Salomaa. Efficient decomposition of quantum gates. *Physical review letters*, 92(17):177902, 2004.

[39] Vivek V Shende, Igor L Markov, and Stephen S Bullock. Minimal universal two-qubit controlled-not-based circuits. *Physical Review A*, 69(6):062321, 2004.

[40] Stephen S Bullock and Igor L Markov. Asymptotically optimal circuits for arbitrary n-qubit diagonal comutations. *Quantum Information & Computation*, 4(1):27–47, 2004.

[41] Israel F Araujo, Daniel K Park, Francesco Petruccione, and Adenilton J da Silva. A divide-and-conquer algorithm for quantum state preparation. *arXiv preprint arXiv:2008.01511*, 2020.

[42] Guang Hao Low, Vadym Kliuchnikov, and Luke Schaeffer. Trading t-gates for dirty qubits in state preparation and unitary synthesis. *arXiv preprint arXiv:1812.00954*, 2018.

[43] 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(4):041015, 2018.

[44] Xiaoming Sun, Guojing Tian, Shuai Yang, Pei Yuan, and Shengyu Zhang. Asymptotically optimal circuit depth for quantum state preparation and general unitary synthesis. *arXiv preprint arXiv:2108.06150v2*, 2021.

[45] Gregory Rosenthal. Query and depth upper bounds for quantum unitaries via grover search. *arXiv preprint arXiv:2111.07992*, 2021.

[46] Xiao-Ming Zhang, Tongyang Li, and Xiao Yuan. Quantum state preparation with optimal circuit depth: Implementations and applications. *arXiv preprint arXiv:2201.11495*, 2022.

[47] Gray Frank. Pulse code communication, March 17 1953. US Patent 2,632,058.

[48] Carla Savage. A survey of combinatorial gray codes. *SIAM review*, 39(4):605–629, 1997.

[49] Edgard N Gilbert. Gray codes and paths on the n-cube. *The bell system technical journal*, 37(3):815–826, 1958.

[50] Jonathan Welch, Alex Bocharov, and Krysta M Svore. Efficient approximation of diagonal unitaries over the clifford+ t basis. *arXiv preprint arXiv:1412.5608*, 2014.

[51] Jonathan Welch, Daniel Greenbaum, Sarah Mostame, and Alán Aspuru-Guzik. Efficient quantum circuits for diagonal unitaries without ancillas. *New Journal of Physics*, 16(3):033040, 2014.

[52] Jiaqing Jiang, Xiaoming Sun, Shang-Hua Teng, Bujiao Wu, Kewen Wu, and Jialin Zhang. Optimal space-depth trade-off of cnot circuits in quantum logic synthesis. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 213–229. SIAM, 2020.

[53] Ketan N Patel, Igor L Markov, and John P Hayes. Optimal synthesis of linear reversible circuits. *Quantum Inf. Comput.*, 8(3):282–294, 2008.

[54] Fino and Algazi. Unified matrix treatment of the fast walsh-hadamard transform. *IEEE Transactions on Computers*, C-25(11):1142 – 1146, 1976.

[55] Cristopher Moore and Martin Nilsson. Parallel quantum computation and quantum codes. *SIAM Journal on Computing*, 31(3):799–815, 2001.

[56] Neil J Ross. Optimal ancilla-free clifford+ v approximation of z-rotations. *Quantum Information & Computation*, 15(11-12):932–950, 2015.

[57] Emanuel Malvetti, Raban Iten, and Roger Colbeck. Quantum circuits for sparse isometries. *Quantum*, 5:412, 2021.

A Circuit depth lower bound

In this section we prove Theorem 3. In [26], the authors presented a depth lower bound of $\Omega\left(\frac{2^n}{n}\right)$ for quantum circuits without ancillary qubits. This can be extended to a lower bound of $\Omega\left(\frac{2^n}{n+m}\right)$ for circuits with m ancillary qubits. Next we prove the linear lower bound.

Lemma 36. *Almost all n-qubit quantum states need a quantum circuit of depth at least* $n - \log n - O(1)$ *to prepare, even if the circuit uses arbitrary single- and double-qubit gates, regardless of the number of ancillary qubits.*

Proof. As two adjacent single-qubit gates on the same qubit can be compressed into one, we can assume that a circuit with minimum depth has double-qubit gates and single-qubit gates appearing in alternative layers. Without loss of generality, assume that the odd layers contain only double-qubit gates and the even layers contain only single-qubit gates. Suppose the circuit depth is D, i.e. it has D layers of gates. Note that a single- or double-qubit gate can be represented by $O(1)$ real parameters.

Consider a time-space directed graph $G = (V, E)$ with $D + 1$ layers $L_1, \dots, L_{D+1}$ of nodes, corresponding to the $D + 1$ time steps separated by the D layers $U_1, \dots, U_D$ of gates. There are $n + m$ nodes in each layer of G, corresponding to the $n + m$ qubits in the circuit with n input qubits and m ancillary qubits. Edges appear only between nodes in adjacent layers L_i, L_{i+1}, and two nodes $(v_i, v_{i+1}) \in E$ if $v_i \in L_i$, $v_{i+1} \in L_{i+1}$, and the two corresponding qubits of v_i and v_{i+1} are among the input and output qubits for some gate in U_i, the i-th layer of gates in the circuit. (Thus each single-qubit gate induces one edge, and each double-qubit gate induces 4 edges.) All edges are in the direction from input to output of the gate.

Since the circuit generates the state, we have $U |0^{n+m}\rangle = |\psi\rangle \otimes |\phi\rangle$, where $|\psi\rangle$ is the target n-qubit state. Define the *light cone* of $|\psi\rangle$ to be the nodes in G that can reach, by walking along the directed edges, the nodes in L_{D+1} corresponding to the qubits in $|\psi\rangle$. Intuitively, only gates within this region contributes to the generation of $|\psi\rangle$. Indeed, we can remove gates outside the light cone, from the last layer to the first, one by one. Each removal of a gate U is equivalent to applying $U^\dagger$ at the end, which only affects $|\phi\rangle$ (and $|\psi\rangle$ remains unchanged.) Thus we can remove all gates outside the light cone yet the remaining circuit still generates $|\psi\rangle$.

As each node v_{i+1} in the graph G connects to at most 2 nodes in the previous layer L_i, the region contains at most $O(n \cdot 2^D)$ nodes, thereby also at most $O(n \cdot 2^D)$ gates in the circuit. As each gate can be fully specified by $O(1)$ real parameters, the circuit has at most $O(n \cdot 2^D)$ parameters. If $D \leq n - \log n - \omega(1)$, the number of parameters is strictly smaller than $2^n - 1$, the dimension of unit sphere S for all possible $|\psi\rangle$. Then the circuit as a map from the parameters to the generated state, has its image a measure-zero subset of S. Since there are only finitely many layouts in a D-layer circuit, the union of these images still has measure 0, thus almost all n-qubit quantum states cannot be generated by circuits of depth $n - \log n - \omega(1)$. □

B Quantum state preparation via Binary search tree

The framework of quantum state preparation is illustrated by a vector

$$v = \left(\sqrt{0.03}, \sqrt{0.07}, \sqrt{0.15}, \sqrt{0.05}, \sqrt{0.1}, \ \sqrt{0.3}, \sqrt{0.2}, \sqrt{0.1}\right)^T \in \mathbb{C}^8.$$

The corresponding quantum state is a 3-qubit quantum state

$$\begin{aligned} |\psi_v\rangle = &\sqrt{0.03}|000\rangle + \sqrt{0.07}|001\rangle + \sqrt{0.15}|010\rangle + \sqrt{0.05}|011\rangle \\ &+ \sqrt{0.1}|100\rangle + \sqrt{0.3}|101\rangle + \sqrt{0.2}|110\rangle + \sqrt{0.1}|111\rangle. \end{aligned}$$

The amplitudes of $|\psi_v\rangle$ are stored in the leaf nodes of the corresponding Binary Search Tree. Every internal node stores the square root of sum of squares of its child nodes. The root node stores the ℓ_2-norm of the vector.

Based on the Binary Search Tree in Figure 7(a), the QSP circuit can be designed layer-by-layer and branch-by-branch, as in Figure 7(b).

C Implementations of tasks in Eq. (5) and Eq. (6)

The first task (Eq. (5)) can be completed by the combination of the circuit in Figure 8, a fact formalized as Lemma 37, which can be easily verified.

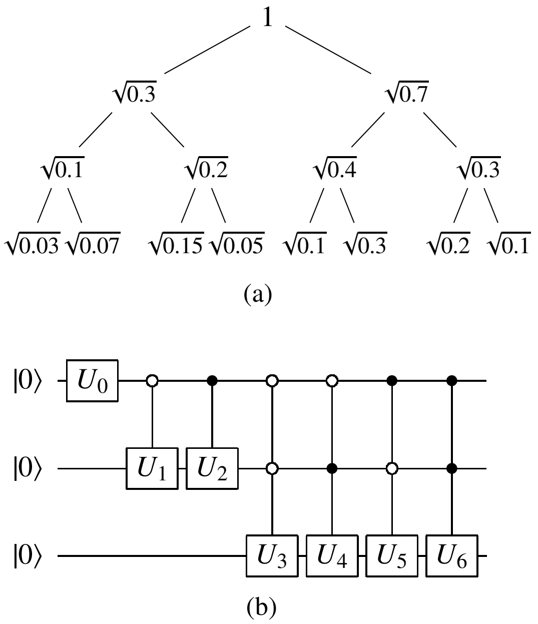


Figure 7: (a) Binanry search tree for vector $v = \left(\sqrt{0.03}, \sqrt{0.07}, \sqrt{0.15}, \sqrt{0.05}, \sqrt{0.1}, \sqrt{0.3}, \sqrt{0.2}, \sqrt{0.1}\right)^T \in \mathbb{C}^8$. (b) The quantum circuit to prepare the 3-qubit state $|\psi_v\rangle$. The single-qubit gates used in this circuit are defined as $U_i = R_y(2\theta_i)$ for $i \in \{0, 1, \ldots, 6\}$. The value of θ_i is shown as follow : $\theta_0 = \arccos\left(\sqrt{0.3/1}\right)$, $\theta_1 = \arccos\left(\sqrt{0.1/0.3}\right)$, $\theta_2 = \arccos\left(\sqrt{0.4/0.7}\right)$, $\theta_3 = \arccos\left(\sqrt{0.03/0.1}\right)$, $\theta_4 = \arccos\left(\sqrt{0.15/0.2}\right)$, $\theta_5 = \arccos\left(\sqrt{0.1/0.4}\right)$, $\theta_6 = \arccos\left(\sqrt{0.2/0.3}\right)$.

Lemma 37. *Let $x = x_1x_2 \ldots x_n$, $s = s_1s_2...s_n \in \{0, 1\}^n$, and $S = \{i_1, \ldots, i_k\} = \{i : s_i = 1\} \subseteq [n]$. The circuits in Figure 8 realize the following transformation:*

$$|x_1x_2 \cdots x_n\rangle \to e^{i\langle s,x\rangle\alpha} |x_1x_2 \cdots x_n\rangle .$$

The second task (Eq. (6)) is accomplished as follows. Based on Lemma 37, one can implement transformation Eq. (5) by using $2^n - 1$ circuits with parameters α_s for all $s \in \{0, 1\}^n - \{0^n\}$ in Figure 8. To determine parameters α_s in Eq. (6), the second task is essentially asking whether the $(2^n - 1) \times (2^n - 1)$ matrix A defined by

$$A(x, s) = \langle x, s\rangle, \quad x, s \in \{0, 1\}^n - \{0^n\} \tag{25}$$

is invertible. The answer is affirmative and the inverse is given by the following lemma, which can be easily verified [51].

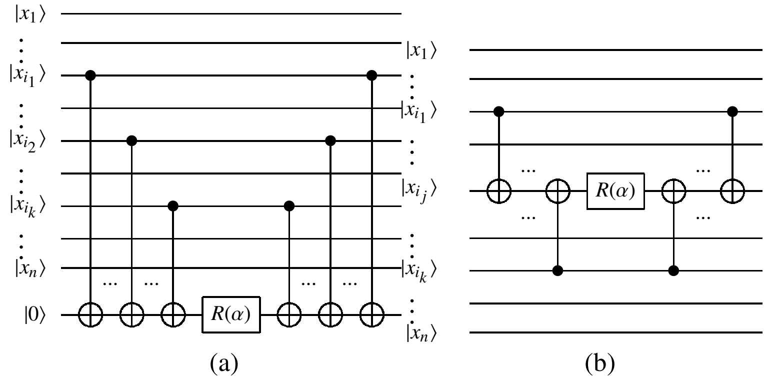
Lemma 38. *The matrix A defined as in Eq. (25) is invertible, and its inverse is $2^{1-n}(2A - \mathbf{J})$, where $\mathbf{J} \in \mathbb{R}^{(2^n-1)\times(2^n-1)}$ is the all-one matrix.*

This gives a way to compute the parameters α_s efficiently on a classical computer.

Lemma 39. *For QSP problem, given a unit vector $v = (v_0, v_1, \cdots, v_{2^n-1})^T \in \mathbb{C}^{2^n}$, the values of $\{\alpha_s : s \in \{0, 1\}^n - \{0^n\}\}$ in Eq. (6) can be calculated on a classical computer using $O(n2^n)$ time and $O(n2^n)$ space.*

Proof. We calculate these α_s in three steps.

1. Our QSP circuit consists of n UCGs $V_1, V_2, \cdots, V_n$ as in Figure 2(a). We calculate all parameters of $V_1, \ldots, V_n$ in time $O(2^n)$ and in space $O(2^n)$ using binary trees [7,22].



(a)

(b)

Figure 8: A quantum circuit to implement transformation $|x_1 x_2 \cdots x_n\rangle \rightarrow e^{i\langle s,x\rangle\alpha} |x_1 x_2 \cdots x_n\rangle$ with string $s = s_1 s_2 \cdots s_n \in \{0,1\}^n$ being the indicator vector of set $S = \{i_1, \ldots, i_k\} \subseteq [n]$, i.e. $s_j = 1$ if $j \in S$ and $s_j = 0$ otherwise. (a) A quantum circuit with an ancillary qubit initialized as $|0\rangle$. The index set of controlled qubit of CNOT gates is S. (b) A quantum circuit without ancillary qubits, where i_j is an arbitrary element in S. The index set of the controlled qubit of CNOT gates is $S - \{i_j\}$ and the index of target qubit is i_j.

2. Secondly, we decompose all UCGs into diagonal unitary matrices and some single-qubit operations according to Eq. (4). As in the proof of Lemma 9, we decompose every single-qubit gate in V_j into R_z gates, S gates and H gates in time $O(1)$ and space $O(1)$, and UCG V_j can be decomposed into 3 diagonal unitary matrices and two H gates and S gates in time $O(2^j)$ and space $O(2^j)$. Hence, the total time and space of this step are $\sum_{j=1}^{n} O(2^j) = O(2^n)$.

3. Thirdly, for every diagonal unitary matrix Λ_j with diagonal element $e^{i\theta(x)}$ for all $x \in \{0,1\}^j - \{0^j\}$, we calculate parameters α_s in Eq. (6). Let

$$\boldsymbol{\alpha} \overset{\text{def}}{=} (\alpha_{0\cdots01}, \alpha_{0\cdots10}, \ldots, \alpha_{1\cdots11})^T \in \mathbb{R}^{2^n-1},$$
$$\boldsymbol{\theta} \overset{\text{def}}{=} (\theta(0\cdots01), \theta(0\cdots10), \ldots, \theta(1\cdots11))^T \in \mathbb{R}^{2^n-1}.$$

Based on Eq. (6) and lemma 38, we have $\boldsymbol{\alpha} = 2^{1-n}(2A - \mathbf{J})\boldsymbol{\theta}$. Notice that $(2A - \mathbf{J})\boldsymbol{\theta}$ is just the Walsh-Hadamard transform on $\boldsymbol{\theta}$. Thus the vector $\boldsymbol{\alpha}$ can be calculated by fast Walsh-Hadamard transform algorithm [54], which costs $O(n2^n)$ time and $O(n2^n)$ space.

Adding these costs up, we see that the values of α_s can be calculated in $O(n2^n)$ time and $O(n2^n)$ space.

□

D Warm-up example: Implement Λ_4 using 8 ancillary qubits

In this section, we will show how to implement Λ_4 with 8 ancillary qubits based on a Gray code, which can help to understand the general case. Our construction of quantum circuit for Λ_4 is shown in Figure 9. All the parameters α_s for $s \in \{0,1\}^4$ are defined in Eq. (6). In Figure 9, the first four qubits initialized as $|x\rangle := |x_1 x_2 x_3 x_4\rangle$ constitute the input register; the next four qubits form the copy register; and the last four qubits form the phase register. All qubits in the copy and the phase register are initialized to $|0\rangle$.

For any 4-bit string $s = s_1 s_2 s_3 s_4 \in \{0,1\}^4$, $s_1 s_2$ is the prefix and $s_3 s_4$ is the suffix. In Step 1-2, we make two copies of prefix $|x_1\rangle, |x_2\rangle$ into the copy register, i.e.,

$$\underbrace{|x_1 x_2 x_3 x_4\rangle}_{\text{input register}} \underbrace{|0000\rangle}_{\text{copy register}} \rightarrow |x_1 x_2 x_3 x_4\rangle |x_1 x_2 x_1 x_2\rangle .\text{(Step 1-2)}$$

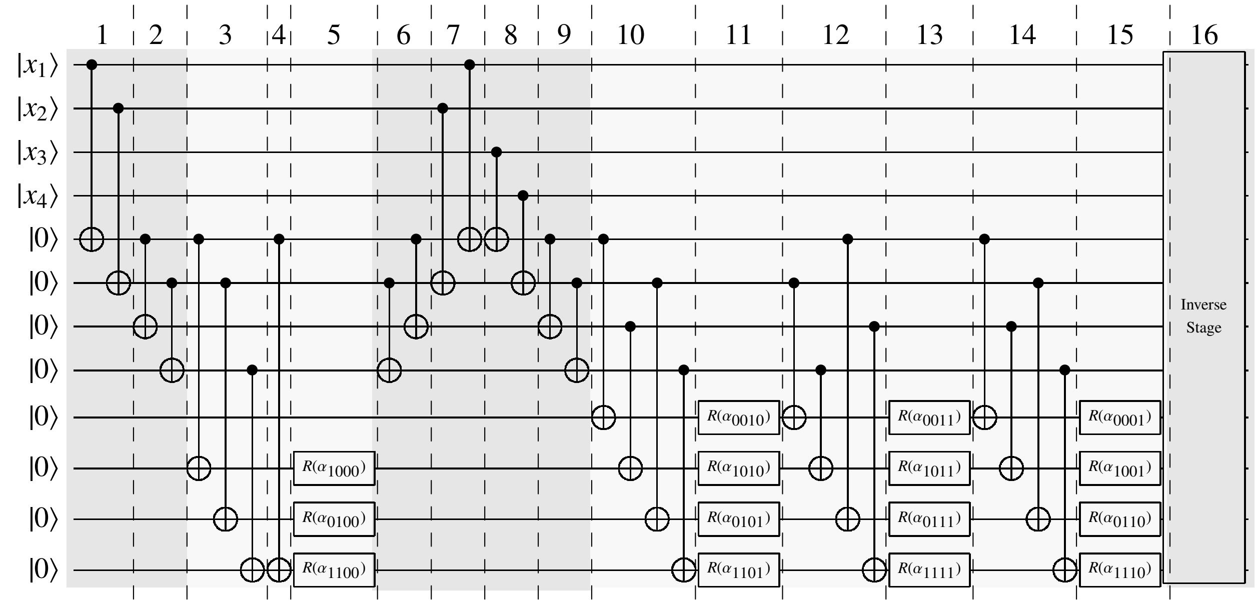


Figure 9: Implementation of Λ_4 with 8 ancillary qubits. The first 4 qubits form the input register, the next 4 qubits form the copy register and the last 4 qubits form the phase register. Step 1-2 are Prefix Copy Stage; Step 3-5 are Gray Initial Stage; Step 6-9 are Suffix Copy Stage; Step 10-15 are Gray Path Stage; Step 16 is Inverse stage. All parameters α_s of phase gate $R(\alpha_s)$ for $s \in \{0,1\}^4$ are determined by Eq. (6). Step 16 consists of inverse quantum circuits of Step 14,12,10,9,8,7,6,4,3,2 and 1 in that order. Step 1-15 are quantum circuits of depth 1 and Step 16 is quantum circuits of depth 11.

In Step 3-5, we generate all 4-bit strings whose suffixes are all 00 in the phase register by CNOT gates. Namely, we realize the following transformation

$$
\begin{aligned}
&\overbrace{|x_1x_2x_3x_4\rangle}^{\text{input register}}\overbrace{|x_1x_2x_1x_2\rangle}^{\text{copy register}}\ \overbrace{|0000\rangle}^{\text{phase register}} \\
\to& |x_1x_2x_3x_4\rangle|x_1x_2x_1x_2\rangle \\
& |\langle 0000, x\rangle, \langle 1000, x\rangle, \langle 0100, x\rangle, \langle 1100, x\rangle\rangle && \text{(Step 3-4)} \\
\to& e^{i\sum_{s\in\{0,1\}^2}\langle s00, x\rangle\alpha_{s00}} |x_1x_2x_3x_4\rangle|x_1x_2x_1x_2\rangle \\
& |\langle 0000, x\rangle, \langle 1000, x\rangle, \langle 0100, x\rangle, \langle 1100, x\rangle\rangle\,. && \text{(Step 5)}
\end{aligned}
$$

In Step 6-9, we transform the copy register to initial state and make two copies of suffix x_3x_4:

$$
\begin{aligned}
&\overbrace{|x_1x_2x_3x_4\rangle}^{\text{input register}}\overbrace{|x_1x_2x_1x_2\rangle}^{\text{copy register}} \\
\to& |x_1x_2x_3x_4\rangle|0000\rangle && \text{(Step 6-7)} \\
\to& |x_1x_2x_3x_4\rangle|x_3x_4x_3x_4\rangle\,. && \text{(Step 8-9)}
\end{aligned}
$$

Up to now, we have generated all prefixes for suffix "00", and next we need to generate all the other 4-bit strings. In order to reduce the number of CNOT gates, we consider two forms of 2-bit Gray code. The 1-Gray code and 2-Gray code starting from 00 are

$$00, 10, 11, 01 \text{ and } 00, 01, 11, 10.$$

If in 1-Gray code and 2-Gray code, we list all the bits changed between adjacent strings, we get two lists: $1, 2, 1$ and $2, 1, 2$. To obtain parallelism, in the first and second qubit in the phase register we generate suffixes $00, 10, 11, 01$ (1-Gray code) in that order. In the third and fourth qubit in the phase register we generate suffixes $00, 01, 11, 10$ (2-Gray code) in that order. In Step 10-15, we implement the following transformation

$$
\begin{aligned}
&e^{i\sum_{s\in\{0,1\}^2}\langle s00,x\rangle\alpha_{s00}}\,|x_1x_2x_3x_4\rangle\,|x_3x_4x_3x_4\rangle\\
&|\langle 0000,x\rangle,\langle 1000,x\rangle,\langle 0100,x\rangle,\langle 1100,x\rangle\rangle\\
\to&e^{i\sum_{s\in\{0,1\}^4-\{0^4\}}\langle s,x\rangle\alpha_s}\,|x_1x_2x_3x_4\rangle\,|x_3x_4x_3x_4\rangle\\
&|\langle 0001,x\rangle,\langle 1001,x\rangle,\langle 0110,x\rangle,\langle 1110,x\rangle\rangle \qquad \text{(Step 10-15)}\\
=&e^{i\theta(x)}\,|x_1x_2x_3x_4\rangle\,|x_3x_4x_3x_4\rangle\\
&|\langle 0001,x\rangle,\langle 1001,x\rangle,\langle 0110,x\rangle,\langle 1110,x\rangle\rangle\,.
\end{aligned}
$$

Every step can be realized by a quantum circuit in depth 1. Step 16 consists of inverse quantum circuits of Step 14,12,10,9,8,7,6,4,3,2 and 1 in order. The total depth of Step 16 is 11. It transforms copy register and phase register to their initial states. Therefore, it implements the following transformation

$$
\begin{aligned}
&e^{i\theta(x)}\,|x_1x_2x_3x_4\rangle\,|x_3x_4x_3x_4\rangle\\
&|\langle 0001,x\rangle,\langle 1001,x\rangle,\langle 0110,x\rangle,\langle 1110,x\rangle\rangle\\
\to&e^{i\theta(x)}\,|x\rangle\,|0000\rangle\,|0000\rangle \qquad \text{(Step 16)}
\end{aligned}
$$

As discussed above, the quantum circuit in Figure 9 is an implementation of Λ_4 with 8 ancillary qubits.

E Proof of Lemma 15

Lemma 15 *Let $t = \lfloor \log \frac{m}{2} \rfloor$ and $\ell = 2^t$. The set $\{0,1\}^n$ can be partitioned into a 2-dimensional array $\{s(j,k) : j \in [\ell], k \in [2^n/\ell]\}$ of n-bit strings, satisfying that*

1. *Strings in the first column $\{s(j,1) : j \in [\ell]\}$ have the last $(n-t)$ bits being all 0, and strings in each row $\{s(j,k) : k \in [2^n/\ell]\}$ share the same first t bits.*
2. *$\forall j \in [\ell], \forall k \in [2^n/\ell - 1]$, $s(j,k)$ and $s(j,k+1)$ differ by 1 bit.*
3. *For any fixed $k \in [2^n/\ell - 1]$, and any $t' \in \{t+1,...,n\}$, there are at most $\left(\frac{m}{2(n-t)}+1\right)$ many $j \in [\ell]$ s.t. $s(j,k)$ and $s(j,k+1)$ differ by the t'-th bit.*

Proof. Consider each n-bit string as two parts, a t-bit prefix followed by an $(n-t)$-bit suffix. We let $\{s(j,1) : j \in [\ell]\}$ run over all ℓ possible prefixes, and for each fixed $j \in [\ell]$, the collection of $\{s(j,k) : k \in [2^n/\ell]\}$ run over all possible suffixes. Thus $\{s(j,k) : j \in [\ell], k \in [2^n/\ell]\}$ form a partition of $\{0,1\}^n$, and the first condition is satisfied.

Now for the j-th set of suffixes $\{s(j,k) : k \in [2^n/\ell]\}$, we identify it with $\{0,1\}^{n-t}$, and apply the $(j', n-t)$-Gray code and Lemma 7 to it, where $j' = ((j-1) \mathit{ mod } (n-t)) + 1 \in \{1,...,n-t\}$. For any $k \in [2^n/\ell - 1]$ and any $t' \in \{t+1,...,n\}$, let us see how many $j \in [\ell]$ have that $s(j,k)$ and $s(j,k+1)$ differ by bit t'. When j runs over $[n-t]$, $s(j,k)$ and $s(j,k+1)$ differ by bit t' exactly once. When j runs over $\{n-t+1,...,2(n-t)$, $s(j,k)$ and $s(j,k+1)$ differ by bit t' again exactly once. Repeating this we can see that when j runs over all $[\ell]$, $s(j,k)$ and $s(j,k+1)$ differ by bit t' for at most $\lceil \ell/(n-t) \rceil \leq m/2(n-t)+1$ times. □

F Proof of Lemma 19

Lemma 19 *The depth and size of the Inverse Stage are at most $O(\log m + 2^n/m)$ and $\frac{m}{2} + \frac{nm}{2} + m + 2^n = 2^n + \frac{3m+nm}{2}$. The effect of this stage is*

$$|x\rangle\,|x_{suf}\rangle\,|f_{[\ell],2^n/\ell}\rangle \xrightarrow{U_{Inverse}} |x\rangle\,|0^{m/2}\rangle\,|0^{m/2}\rangle\,.$$

Proof. The depth is just the summation of the CNOT depth of the first four stages, which is $O\Big(\log m + 2\log m + 2\log m + 2^n/\ell\Big) = O\Big(\log m + 2^n/m\Big)$. The analyze of size is similar by adding up the sizes in previous stages. The effect is shown as follows, which holds by Lemma 18, Eq. (11), Eq. (9) and Eq. (7).

$$|x\rangle\,|x_{suf}\rangle\,|f_{[\ell],2^n/\ell}\rangle \xrightarrow{U_2^\dagger\cdots U_{2^n/\ell}^\dagger} |x\rangle\,|x_{suf}\rangle\,|f_{[\ell],1}\rangle \xrightarrow{U_{copy,1}U_{copy,2}^\dagger}$$

$$|x\rangle\,|x_{pre}\rangle\,|f_{[\ell],1}\rangle \xrightarrow{U_1^\dagger} |x\rangle\,|x_{pre}\rangle\,|0^{\frac{m}{2}}\rangle \xrightarrow{U_{copy,1}^\dagger} |x\rangle\,|0^{\frac{m}{2}}\rangle\,|0^{\frac{m}{2}}\rangle$$

□

G Proof of Lemma 20

Lemma 20 *The circuit implements the operation in Eq.* (3) *in depth* $O(\log m + 2^n/m)$ *and in size* $3\cdot 2^n + nm + \frac{7}{2}m$.

Proof. For the depth, simply adding up the depth and the size of the five stages gives the bound. Next we analyze the operation step by step as follows. The three registers are the input, copy and phase registers, respectively.

$$\begin{aligned}
&|x\rangle\,|0^{m/2}\rangle\,|0^{m/2}\rangle \\
&\xrightarrow{U_{copy,1}} |x\rangle\,|x_{pre}\rangle\,|0^{m/2}\rangle && \text{(Eq. (7))}\\
&\xrightarrow{U_{GrayInit}} e^{i\sum\limits_{j\in[\ell]} f_{j,1}(x)\alpha_{s(j,1)}}\,|x\rangle\,|x_{pre}\rangle\,|f_{[\ell],1}\rangle && \text{(Eq. (8))}\\
&\xrightarrow{U_{copy,2}U_{copy,1}^\dagger} e^{i\sum\limits_{j\in[\ell]} f_{j,1}(x)\alpha_{s(j,1)}}\,|x\rangle\,|x_{suf}\rangle\,|f_{[\ell],1}\rangle && \text{(Eq. (11))}\\
&\xrightarrow{R_2U_2} e^{i\sum\limits_{\substack{j\in[\ell]\\ k\in[2]}} f_{j,k}(x)\alpha_{s(j,k)}}\,|x\rangle\,|x_{suf}\rangle\,|f_{[\ell],2}\rangle && \text{(Eq. (12))}\\
&\vdots\\
&\xrightarrow{R_{\frac{2n}{\ell}}U_{\frac{2n}{\ell}}} e^{i\sum\limits_{\substack{j\in[\ell]\\ k\in[\frac{2n}{\ell}]}} f_{j,k}(x)\alpha_{s(j,k)}}\,|x\rangle\,|x_{suf}\rangle\,|f_{[\ell],\frac{2^n}{\ell}}\rangle && \text{(Eq. (12))}\\
&\quad= e^{i\sum\limits_{s\in\{0,1\}^n}\langle x,s\rangle\alpha_s}\,|x\rangle\,|x_{suf}\rangle\,|f_{[\ell],2^n/\ell}\rangle && \text{(Lem 15)}\\
&\quad= e^{i\theta(x)}\,|x\rangle\,|x_{suf}\rangle\,|f_{[\ell],2^n/\ell}\rangle && \text{(Eq. (6))}\\
&\xrightarrow{U_{Inverse}} e^{i\theta(x)}\,|x\rangle\,|0^{m/2}\rangle\,|0^{m/2}\rangle && \text{(Eq. (13))}\\
&\quad= \Lambda_n\,|x\rangle\,|0^{m/2}\rangle\,|0^{m/2}\rangle
\end{aligned}$$

□

H Construction of linearly independent sets

What remains for completing the Generate Stage is the construction of sets $T^{(1)}, T^{(2)}, \ldots, T^{(\ell)}$, which we will show next.

Lemma 40. *There exist sets* $T^{(1)}, T^{(2)}, \cdots, T^{(\ell)} \subseteq \{0,1\}^n - \{0^n\}$, *for some integer* $\ell \le \frac{2^{n+2}}{n+1} - 1$, *such that:*

1. *For any* $i \in [\ell]$, $|T^{(i)}| = n$;
2. *For any* $i \in [\ell]$, *the Boolean vectors in* $T^{(i)} = \{t_1^{(i)}, t_2^{(i)}, \cdots, t_n^{(i)}\}$ *are linearly independent over* $\mathbb{F}_2$;

3. $\bigcup_{i\in[\ell]} T^{(i)} = \{0,1\}^n - \{0^n\}$.

Proof. For any n-bit vector $x \in \{0,1\}^n$, let $S_x = \{x \oplus e_1, x \oplus e_2, \cdots, x \oplus e_n\}$. Firstly, we construct a set $L \subseteq \{0,1\}^n$ which satisfies $|L| \le \frac{2^n+1}{n+1}$ and $\{0,1\}^n = (\bigcup_{x\in L} S_x) \cup L$. Let $k = \lceil \log(n+1) \rceil$. For $t \in [n]$, denote the k-bit binary representation of integer t by $t_k \cdots t_2 t_1$, where $t_1, \ldots, t_k \in \{0,1\}$ and $t = \sum_{i=1}^{k} t_i 2^{i-1}$. We use a bar to denote the corresponding column vector, i.e.

$$\overline{t} = [t_1, t_2, \ldots, t_k]^T \in \{0,1\}^k.$$

Define a $k \times n$ Boolean matrix H by concatenating vectors $\overline{1}, \overline{2}, \cdots, \overline{n}$ together, i.e.

$$H = [\overline{1}, \overline{2}, \cdots, \overline{n}] \in \{0,1\}^{k\times n}.$$

Note that the k-dimensional identity matrix $I_k = [\overline{2^0}, \overline{2^1}, \ldots, \overline{2^{k-1}}]$ is a submatrix of H, therefore H is full row rank, i.e. $\mathrm{rank}(H) = k$. Define sets

$$\begin{aligned} L^{(0)} &= \{x \in \{0,1\}^n : Hx = 0^k\}, \\ L^{(1)} &= \{x \in \{0,1\}^n : Hx = 1^k\}, \end{aligned} \tag{26}$$

and

$$\begin{aligned} A^{(0)} &= \{x \in \{0,1\}^n : (Hx)_k = 0\}, \\ A^{(1)} &= \{x \in \{0,1\}^n : (Hx)_k = 1\}. \end{aligned} \tag{27}$$

For each $x \in \{0,1\}^n$, the last bit of Hx is either 0 or 1, thus $A^{(0)} \cup A^{(1)} = \{0,1\}^n$. Also note that for each $b \in \{0,1\}$, $L^{(b)}$ requires all bits being b and $A^{(b)}$ only requires the last bit being b, thus $L^{(b)} \subseteq A^{(b)}$.

Now we will show

$$A^{(0)} \subseteq L^{(0)} \cup \left(\cup_{x\in L^{(0)}} S_x\right) \quad \text{and} \quad A^{(1)} \subseteq L^{(1)} \cup \left(\cup_{x\in L^{(1)}} S_x\right).$$

For any $y \in A^{(0)} - L^{(0)}$, consider $\overline{t} = Hy$: Since it satisfies $\overline{t}_k = 0$ and $t_{k-1} \cdots t_1 \neq 0^{k-1}$, we have that

$$1 \le t \le \sum_{i=1}^{k-1} 2^{i-1} < 2^{k-1} = 2^{\lceil \log(n+1) \rceil - 1} < 2^{\log(n+1)} = n + 1.$$

Therefore, $1 \le t \le n$, and we can thus use $He_t = \overline{t}$ to obtain the following equality

$$H(y \oplus e_t) = Hy \oplus He_t = \overline{t} \oplus \overline{t} = 0^k.$$

Therefore, $y \oplus e_t \in L^{(0)}$. That is, for any $y \in A^{(0)} - L^{(0)}$, there exists an $x \in L^{(0)}$ s.t. $y = x \oplus e_t$ for some $t \in [n]$. Hence,

$$A^{(0)} \subseteq L^{(0)} \cup \left(\cup_{x\in L^{(0)},\ t\in[n]} \{x \oplus e_t\}\right) = L^{(0)} \cup \left(\cup_{x\in L^{(0)}} S_x\right).$$

For any $y \in A^{(1)} - L^{(1)}$, $\overline{t} = Hy$ satisfies $\overline{t}_k = 1$ and $t_{k-1}...t_1 \neq 1^{k-1}$. It looks symmetric to the $A^{(0)} - L^{(0)}$ case but there is a technicality that the corresponding integer t may be outside the range $[n]$. To remedy this, define $\overline{t'} = \overline{t} \oplus 1^k$ (and let t' be the integer corresponding to vector $\overline{t'}$). Now that $\overline{t'}_k = 0$ and $t'_{k-1}...t'_1 \neq 0^{k-1}$, and we know $t' \in [n]$. Thus we can again get $He_{t'} = \overline{t'}$ and, in turn,

$$H(y \oplus e_{t'}) = Hy \oplus He_{t'} = \overline{t} \oplus \overline{t'} = \overline{t} \oplus \overline{t} \oplus 1^k = 1^k.$$

Therefore, $y \oplus e_{t'} \in L^{(1)}$, and

$$A^{(1)} \subseteq L^{(1)} \cup \left(\cup_{x\in L^{(1)},\ t'\in[n]} \{x \oplus e_{t'}\}\right) = L^{(1)} \cup \left(\cup_{x\in L^{(1)}} S_x\right).$$

Let $L = L^{(0)} \cup L^{(1)}$. We have

$$\{0,1\}^n = A^{(0)} \cup A^{(1)} \subseteq L^{(0)} \cup \left(\cup_{x \in L^{(0)}} S_x\right) \cup L^{(1)} \cup \left(\cup_{x \in L^{(1)}} S_x\right) = L \cup (\cup_{x \in L} S_x) \subseteq \{0,1\}^n.$$

Recall that $\mathrm{rank}(H) = k$ over field $\mathbb{F}_2$, the size of solution set $L^{(b)}$ is $|L^{(b)}| = 2^{n-k} = 2^{n-\lceil \log(n+1) \rceil} \leq \frac{2^n}{n+1}$, for each $b \in \{0,1\}$. Thus $|L| = |L^{(0)}| + |L^{(1)}| \leq \frac{2^{n+1}}{n+1}$.

We have constructed a set $L \subseteq \{0,1\}^n$ of size at most $\frac{2^{n+1}}{n+1}$ satisfying $L \cup (\cup_{x \in L} S_x) = \{0,1\}^n$. We will now use this set L to construct $\ell \leq \frac{2^{n+2}}{n+1} - 1$ sets $T^{(i)}$ which satisfy the three properties in the statement of the present lemma.

Since 0^n is a solution of $Hx = 0^k$, it holds that $0^n \in L^{(0)} \subseteq L$. Note that the vectors in $S_{0^n} = \{e_1, e_2, \ldots, e_n\}$ are linearly independent. For any $x \in L$ and $x \neq 0^n$, let us construct two sets of linearly independent vectors $S_x^{(0)}$ and $S_x^{(1)}$. Since $\mathrm{rank}[x \oplus e_1, x \oplus e_2, \cdots, x \oplus e_n] \geq n-1$ over field $\mathbb{F}_2$, we can select $n-1$ linearly independent vectors from S_x to form a set $S_x^{(0)} \subseteq S_x$. Let $S_x^{(1)} = (S_x - S_x^{(0)} - \{0^n\}) \cup \{x\}$. It is not hard to verify that if $x = e_j$ for some $j \in [n]$, then $S_x^{(1)} = \{x\} = \{e_j\}$; if $x \notin \{0^n, e_1, \ldots, e_n\}$, then $S_x^{(1)} = \{x, x \oplus e_j\}$ for some $j \in [n]$. In any case, the vector(s) in $S_x^{(1)}$ are linearly independent (the same for $S_x^{(0)}$), and it holds that $S_x^{(0)} \cup S_x^{(1)} = S_x \cup \{x\} - \{0^n\}$. Thus for each $b \in \{0,1\}$, we can always extend the set $S_x^{(b)}$ to $T_x^{(b)}$ of n linearly independent vectors by adding some vectors. Recalling $\{0,1\}^n = (\cup_{x \in L} S_x) \cup L$, we have

$$\begin{aligned}
&\{0,1\}^n - \{0^n\} \\
&= (\cup_{x \in L} S_x) \cup L - \{0^n\} \\
&= \cup_{x \in L} (S_x \cup \{x\}) - \{0^n\} \\
&= \left(\cup_{x \in L - \{0^n\}} (S_x \cup \{x\} - \{0^n\})\right) \cup S_{0^n} \\
&= \left(\cup_{x \in L - \{0^n\}} (S_x^{(0)} \cup S_x^{(1)})\right) \cup S_{0^n} \\
&= \left(\cup_{x \in L - \{0^n\}} S_x^{(0)}\right) \cup \left(\cup_{x \in L - \{0^n\}} S_x^{(1)}\right) \cup S_{0^n} \\
&\subseteq \left(\cup_{x \in L - \{0^n\}} T_x^{(0)}\right) \cup \left(\cup_{x \in L - \{0^n\}} T_x^{(1)}\right) \cup S_{0^n} \\
&\subseteq \{0,1\}^n - \{0^n\}.
\end{aligned}$$

Now collect $\{T_x^{(0)} : x \in L - \{0^n\}\}$, $\{T_x^{(1)} : x \in L - \{0^n\}\}$ and S_{0^n} as our sets $T^{(1)}, \ldots, T^{(\ell)}$. Since $|L| \leq \frac{2^{n+1}}{n+1}$ and $0^n \in L$, the collection contains $\ell \leq 2 \cdot (\frac{2^{n+1}}{n+1} - 1) + 1 = \frac{2^{n+2}}{n+1} - 1$ sets, each consisting of n linearly independent vectors, and the collection of all these vectors is exactly $\{0,1\}^n - \{0^n\}$. This completes the proof. □

I Correctness of circuit framework in Figure 4

In this section, we shown the correctness of circuit framework in Figure 4. For any input state $|x\rangle$, the quantum circuit $(\Lambda_{r_c} \otimes \mathcal{R})\mathcal{G}_\ell\mathcal{G}_{\ell-1}\cdots\mathcal{G}_1$ makes the following sequence of operations.

$$\begin{aligned}
|x\rangle &= |x_{control}\rangle\,|y^{(0)}\rangle \\
&\xrightarrow{\mathcal{G}_1} e^{i\sum_{s\in F_1}\langle s,x\rangle\alpha_s}\,|x_{control}\rangle\,|y^{(1)}\rangle \\
&\xrightarrow{\mathcal{G}_2} e^{i\sum_{s\in F_1\cup F_2}\langle s,x\rangle\alpha_s}\,|x_{control}\rangle\,|y^{(2)}\rangle \\
&\ \ \vdots \\
&\xrightarrow{\mathcal{G}_\ell} e^{i\sum_{s\in\bigcup_{k\in[\ell]}F_k}\langle s,x\rangle\alpha_s}\,|x_{control}\rangle\,|y^{(\ell)}\rangle \\
&\xrightarrow{\mathbb{I}_{r_c}\otimes\mathcal{R}} e^{i\sum_{s\in\bigcup_{k\in[\ell]}F_k}\langle s,x\rangle\alpha_s}\,|x_{control}\rangle\,|y^{(0)}\rangle \\
&\xrightarrow{\Lambda_{r_c}\otimes\mathbb{I}_{r_t}} e^{i\sum_{s\in\left(\bigcup_{k\in[\ell]}F_k\right)\cup\left(\{c0^{r_t}\}_{c\in\{0,1\}^{r_c}-\{0^{r_c}\}}\right)}\langle s,x\rangle\alpha_s} \\
&\qquad\quad |x_{control}\rangle\,|y^{(0)}\rangle \\
&= e^{i\sum_{s\in\{0,1\}^n-\{0^n\}}\langle s,x\rangle\alpha_s}\,|x_{control}\rangle\,|y^{(0)}\rangle \\
&= e^{i\theta(x)}\,|x_{control}\rangle\,|y^{(0)}\rangle \\
&= e^{i\theta(x)}\,|x\rangle
\end{aligned}$$

The first equation holds by Eq. (14). For arbitrary $k \in [\ell]$, unitary transformation $\mathcal{G}_k$ holds by Eq.(17) and $F_j \cap F_k = \emptyset$, for $j \in [k-1]$. Unitary transformation $\mathcal{R}$ holds by Eq. (18). Unitary transformation Λ_{r_c} holds by Eq. (19). The last two equations hold by Eq. (16) and Eq. (6)), respectively.

J Proof of Lemma 26

Lemma 26 *The quantum circuit defined above is of depth $O(2^{r_c})$ and of size $r_c 2^{r_c+1}$, and implements Gray Path Stage $U_{GrayPath}$ in Eq. (22).*

Proof. We first show the correctness. For each $p \in [2^{r_c}]$, let us define a set $F_k^{(p)}$ by

$$F_k^{(p)} = \left\{s:\ s \in F_k \text{ and } s = c_p^i t_i^{(k)} \text{ for some } i \in [r_t]\right\}. \tag{28}$$

By definition of F_k in Eq. (15), the collection of $F_k^{(p)}$'s satisfy

$$F_k^{(i)} \cap F_k^{(j)} = \emptyset \text{ for all } i \neq j \in [2^{r_c}], \tag{29}$$

$$F_k = \bigcup_{p\in[2^{r_c}]} F_k^{(p)}. \tag{30}$$

Now we can see how the Gray Path Stage $U_{GrayPath}$ in Eq. (22) is realized by the above quantum circuit $C_1, C_2, \ldots, C_{2^{r_c}+1}$ step by step as follows. For $j \in [2^{r_c}]$, define $|f_j\rangle := |\langle c_j^1 t_1^{(k)}, x\rangle, \langle c_j^2 t_2^{(k)}, x\rangle, \cdots, \langle c_j^{r_t} t_{r_t}^{(k)}, x\rangle\rangle$.

Note that $|f_1\rangle = |\langle 0^{r_c} t_1^{(k)}, x\rangle, \langle 0^{r_c} t_2^{(k)}, x\rangle, \cdots, \langle 0^{r_c} t_{r_t}^{(k)}, x\rangle\rangle$ since $c_1^i = 0^{r_c}$ for all $i \in [r_t]$.

$$\begin{aligned}
&|x_{control}\rangle |y^{(k)}\rangle \\
&= |x_{control}\rangle |f_1\rangle && \text{(Eq. (14))} \\
&\xrightarrow{C_1} e^{i\sum_{s\in F_k^{(1)}}\langle s,x\rangle\alpha_s} |x_{control}\rangle |f_1\rangle && \text{(Eq. (28))} \\
&\xrightarrow{C_2} e^{i\sum_{s\in F_k^{(1)}\cup F_k^{(2)}}\langle s,x\rangle\alpha_s} |x_{control}\rangle |f_2\rangle && \text{(Eq. (28),(29))} \\
&\ \vdots \\
&\xrightarrow{C_{2^{r_c}}} e^{i\sum_{s\in\bigcup_{p\in[2^{r_c}]} F_k^{(p)}}\langle s,x\rangle\alpha_s} |x_{control}\rangle |f_{2^{r_c}}\rangle && \text{(Eq. (28), (29))} \\
&= e^{i\sum_{s\in F_k}\langle s,x\rangle\alpha_s} |x_{control}\rangle |f_{2^{r_c}}\rangle && \text{(Eq. (30))} \\
&\xrightarrow{C_{2^{r_c}+1}} e^{i\sum_{s\in F_k}\langle s,x\rangle\alpha_s} |x_{control}\rangle |f_1\rangle \\
&= e^{i\sum_{s\in F_k}\langle s,x\rangle\alpha_s} |x_{control}\rangle |y^{(k)}\rangle && \text{(Eq. (14))}
\end{aligned}$$

Next we analyze the depth. Phase 1 consists of rotations applied on different qubits in the target register, thus can be made in a single depth. In each phase $p \in \{2, 3, \ldots, 2^{r_c}\}$, since c_{p-1}^i and c_p^i differ by only 1 bit, one CNOT gate suffices to implement the function $\langle c_p^i t_i^{(k)}, x\rangle$ from $\langle c_{p-1}^i t_i^{(k)}, x\rangle$ in the previous phase. The control and target qubit of this CNOT gate is the h_{ip}-th qubit in control register and the i-th qubit in target register. According to Eq. (23), indices $h_{1p}, h_{2p}, \ldots, h_{r_t p}$ of control qubits are all different, and therefore, all the CNOT gates in step $p.1$ can be implemented in depth 1. The rotations in step $p.2$ are on different qubits and thus fit in one depth as well. Similarly, phase $2^{r_c} + 1$ can also be implemented in depth 1. Thus the total depth of Gray Path Stage is at most $1 + 2 \cdot (2^{r_c} - 1) + 1 = 2 \cdot 2^{r_c}$. The size of Gray Path Stage is at most $r_c \cdot 2 \cdot 2^{r_c} = r_c 2^{r_c+1}$. □

K Proof of Lemma 28

The Toffoli gate is a 3-qubit CCNOT gate where we flip the basis $|0\rangle, |1\rangle$ of (i.e. apply X gate to) the third qubit conditioned on the first two qubits are both on $|1\rangle$. This can be extended to an n-qubit Toffoli gate, which applies the X gate to the last qubit conditioned on the first $(n-1)$ qubits all being on $|1\rangle$. An n-qubit Toffoli gate can be implemented by a circuit of $O(n)$ size and depth [24].

Lemma 41. *An n-qubit Toffoli gate can be implemented by a quantum circuit of depth and size $O(n)$ without ancillary qubits.*

The next result we need says that cascading CNOT gates with the same target qubit can be exponentially compressed [55].

Lemma 42. *Let C be a quantum circuit consisting of n CNOT gates with the same target qubit and distinct controlled qubits. Then C can be compressed to $O(\log n)$ depth and $O(n)$ size without using ancillary qubits.*

Recall the description of Lemma 28.
Lemma 28 *The following unitary transformation on 2^n qubits*

$$|e_i\rangle \to |i\rangle |0^{2^n-n}\rangle, \textit{ for all } i \in \{0\} \cup [2^n - 1], e_i \in \{0,1\}^{2^n},$$

can be implemented by a quantum circuit using single-qubit gates and CNOT gate with 2^{n+1} ancillary qubits, of depth $O(n)$ and size $O(2^n)$.

Proof. We will implement Eq. (24) with 2^{n+1} ancillary qubits in three steps:

Step 1: $\underbrace{|e_i\rangle}_{\substack{2^n\\ \text{qubits}}} |0^{2^{n+1}}\rangle \to |0^{2^n}\rangle \underbrace{|e_s\rangle}_{\substack{2^{n/2}\\ \text{qubits}}} \underbrace{|e_t\rangle}_{\substack{2^{n/2}\\ \text{qubits}}} |0^{2^{n+1}-2\cdot 2^{n/2}}\rangle$ for all $s,t \in \{0\} \cup [2^{n/2}-1]$ and $i = s\cdot 2^{n/2}+t$.

Step 2: $|0^{2^n}\rangle |e_s\rangle |e_t\rangle |0^{2^{n+1}-2\cdot 2^{n/2}}\rangle \to |0^{2^n}\rangle |i\rangle |0^{2^{n+1}-n}\rangle$ for all $s,t \in \{0\} \cup [2^{n/2}-1]$ and $i = s\cdot 2^{n/2}+t$.

Step 3: $|0^{2^n}\rangle |i\rangle |0^{2^{n+1}-n}\rangle \to |i\rangle |0^{3\cdot 2^n - n}\rangle$ for all $i \in \{0\} \cup [2^n-1]$.

In these three steps, the first 2^n qubits are called register A. The last 2^{n+1} are ancillary qubits, which are initialized as $|0\rangle$ and called register B. Let the first $2^{n/2}$ qubits of register B be register B_1 and the second $2^{n/2}$ qubits of register B be register B_2.

Firstly, we implement Step 1 by a quantum circuit of depth $O(n)$ and size $O(2^n)$ with 2^{n+1} ancillary qubits. Step 1 consists of two phases.

- Step 1a: $|e_i\rangle |0^{2^{n+1}}\rangle \to |e_i\rangle |e_s\rangle |e_t\rangle |0^{2^{n+1}-2\cdot 2^{n/2}}\rangle$ for all $s,t \in \{0\} \cup [2^{n/2}-1]$ and $i = s\cdot 2^{n/2}+t$.

  Let $CNOT^i_{j,(k)}$ denote a CNOT gate whose controlled qubit is the i-th qubit of register A, and target qubit is the j-th qubit of register B_k for $k \in [2]$. Let $CNOT^i_{s,t} = CNOT^i_{s,(1)} CNOT^i_{t,(2)}$. Therefore, Step 1a can be realized by a CNOT circuit as follows:

$$
\begin{aligned}
&\prod_{s,t=0}^{2^{n/2}-1} CNOT^{s\cdot 2^{n/2}+t}_{s,t}\\
=&\left[\prod_{s,t=0}^{2^{n/2}-1} CNOT^{s\cdot 2^{n/2}+t}_{s,(1)}\right]\left[\prod_{s,t=0}^{2^{n/2}-1} CNOT^{s\cdot 2^{n/2}+t}_{t,(2)}\right]\\
=&\left[\prod_{s=0}^{2^{n/2}-1}\left(\prod_{t=0}^{2^{n/2}-1} CNOT^{s\cdot 2^{n/2}+t}_{s,(1)}\right)\right]\cdot\left[\prod_{t=0}^{2^{n/2}-1}\left(\prod_{s=0}^{2^{n/2}-1} CNOT^{s\cdot 2^{n/2}+t}_{t,(2)}\right)\right].
\end{aligned}
$$

  For every $s \in \{0\} \cup [2^{n/2}-1]$, all CNOT gates in $C'_s \overset{\text{def}}{=} \prod_{t=0}^{2^{n/2}-1} CNOT^{s\cdot 2^{n/2}+t}_{s,(1)}$ have different controlled qubits and the same target qubit. According to Lemma 42, C'_s can be implemented by a circuit of depth $O(n)$ and size $O(2^{n/2})$ without ancillary qubits. For all $s \in \{0\} \cup [2^{n/2}-1]$, C'_s act on different qubits. Therefore, they can be paralleled and $\prod_s C'_s$ can be implemented by a circuit of depth $O(n)$ and size $O(2^n)$ without ancillary qubits. By similar discussion, $\prod_t C''_t$ can be implemented by a circuit of depth $O(n)$ and size $O(2^n)$ without ancillary qubits, where $C''_t \overset{\text{def}}{=} \prod_{t=0}^{2^{n/2}-1} CNOT^{s\cdot 2^{n/2}+t}_{t,(2)}$. So Step 1a can be implemented by a quantum circuit of depth $O(n)$ and size $O(2^n)$ without ancillary qubits.

- Step 1b: $|e_i\rangle |e_s\rangle |e_t\rangle |0^{2^{n+1}-2\cdot 2^{n/2}}\rangle \to |0^{2^n}\rangle |e_s\rangle |e_t\rangle |0^{2^{n+1}-2\cdot 2^{n/2}}\rangle$ for all $s,t \in \{0\} \cup [2^{n/2}-1]$ and $i = s\cdot 2^{n/2}+t$. Let $\mathsf{T}^{s,t}_i$ denotes a 3-qubit Toffoli gate, whose controlled qubits are the s-th qubit in register B_1 and t-th qubit in register B_2, and the target qubit is the i-th qubit in register A. The unitary transform of Step 1b is realized by applying all Toffoli gates $\mathsf{T}^{s,t}_{s\cdot 2^{n/2}+t}$. To reduce the circuit depth, we make $2^{n/2}-1$ copies of register B_1, B_2 in last 2^{n+1} qubits of register B:

$$
|e_i\rangle |e_s\rangle |e_t\rangle |0^{2^{n+1}-2\cdot 2^{n/2}}\rangle \to |e_i\rangle \underbrace{|e_s\rangle |e_t\rangle \cdots |e_s\rangle |e_t\rangle}_{2^{n/2}\text{ copies of }|e_s\rangle|e_t\rangle}
$$

  for all $s,t \in \{0\} \cup [2^{n/2}-1]$ and $i = s\cdot 2^{n/2}+t$. This transformation can be parallelized to depth $O(n)$ (by Lemma 14) and size $O(2^n)$. Because there are $2^{n/2}$ copies of $|e_s\rangle |e_t\rangle$, all Toffoli gates $\mathsf{T}^{s,t}_{s\cdot 2^{n/2}+t}$ are on distinct control and target qubits, thus can be executed in parallel in depth $O(1)$. Finally, restore the last $2^{n+1}-2\cdot 2^{n/2}$ qubits of register B to all-zero state in $O(n)$ depth. Overall, Step 1b can be implemented by a quantum circuit of depth $O(n)$ and size $O(2^n)$ with 2^{n+1} ancillary qubits.

Secondly, Step 2 can be realized by a circuit of depth $O(n)$ and size $O(n2^{n/2})$ with $O(n2^{n/2})$ ancillary qubits. Now, we rewrite the transformation of Step 2:

$$|0^{2^n}\rangle\,|e_s\rangle\,|e_t\rangle\,|0^{2^{n+1}-2\cdot 2^{n/2}}\rangle \to |0^{2^n}\rangle\,\underbrace{|s\rangle}_{\frac{n}{2}\text{ qubits}}\,\underbrace{|t\rangle}_{\frac{n}{2}\text{ qubits}}\,|0^{2^{n+1}-n}\rangle$$

for all $s, t \in \{0\} \cup [2^{n/2} - 1]$. If we can realize transformation

$$|e_s\rangle\,|0^k\rangle \to |s\rangle\,|0^{2^k}\rangle\,, \text{for } e_s \in \{0,1\}^{2^k},\;\; s \in \{0\} \cup [2^k - 1] \tag{31}$$

by a quantum circuit of depth $O(k)$ and size $O(k2^k)$ with $k2^k$ ancillary qubits, then we can implement Step 2 by a quantum circuit of depth $O(n)$ and size $O(n2^{n/2})$ with $(n/2)2^{n/2}$ ancillary qubits.

We will implement Eq. (31) with $k2^k$ ancillary qubits in three steps:

Step 2a: $|e_s\rangle\,|0^k\rangle\,|0^{k2^k}\rangle \to |e_s\rangle\,|s\rangle\,|0^{k2^k}\rangle$ for all $s \in \{0\} \cup [2^k - 1]$;

Step 2b: $|e_s\rangle\,|s\rangle\,|0^{k2^k}\rangle \to |0^{2^k}\rangle\,|s\rangle\,|0^{k2^k}\rangle$ for all $s \in \{0\} \cup [2^k - 1]$;

Step 2c: $|0^{2^k}\rangle\,|s\rangle\,|0^{k2^k}\rangle \to |s\rangle\,|0^{2^k}\rangle\,|0^{k2^k}\rangle$ for all $s \in \{0\} \cup [2^k - 1]$.

The first 2^k qubits are called register A and the second n qubits are called register B. The last $k2^k$ qubits are ancillary qubits, which are called register C. Let $CNOT^s_j$ denote a CNOT gate, whose controlled qubit is the s-th qubit in register A and target qubit is the j-th qubit in register B for all $i \in \{0\} \cup [2^k - 1]$ and $j \in \{0\} \cup [n - 1]$. Define $CNOT^s_{S_s} \overset{\text{def}}{=} \prod_{j\in S_s} CNOT^s_j$, where $S_s \overset{\text{def}}{=} \left\{j | s_j = 1 \text{ for } j \in \{0\} \cup [k-1]\right\}$.

- Step 2a: Firstly, we implement Step 2a by a quantum circuit with $k2^k$ ancillary qubits of depth $O(k)$. It can be easily verified that Step 2a can be implemented by a CNOT circuit

$$\prod_{s\in\{0\}\cup[2^k-1]} CNOT^s_{S_s} = \prod_{s\in\{0\}\cup[2^k-1]} \prod_{j\in S_s} CNOT^s_j = \prod_{s\in\{0\}\cup[k-1]} \prod_{s:s\in\{0\}\cup[2^k-1],s_j=1} CNOT^s_j.$$

For every $j \in \{0\} \cup [k-1]$, CNOT circuit $C_j \overset{\text{def}}{=} \prod_{s:s\in\{0\}\cup[2^k-1],s_j=1} CNOT^s_j$ consists of 2^{k-1} CNOT gates. All these CNOT gates have distinct control qubits and the same target qubit. According to Lemma 42, C_j can be parallelized to depth $O(k)$ and size $O(2^k)$ without ancillary qubits. Step 2a consists of $C_0, C_1, \ldots, C_{k-1}$ and the target qubits of CNOT gates in C_t and C_ℓ are different if $t \neq \ell$. In order to reduce the depth of Step 2a, we make k copies of register A in ancillary qubits (register C). Then $C_0, C_1, \ldots, C_{k-1}$ can be implemented simultaneously using the k copies of register A. Thus $\prod_{j=0}^{k-1} C_j$ can be implemented simultaneously in depth $O(k)$ and size $O(k2^k)$. Finally, we reset register C back to 0 in depth $O(k)$ and size $O(2^k)$. Step 1 is summarized as follows:

$$\begin{aligned}
&|e_s\rangle\,|0^k\rangle\,|0^{k2^k}\rangle \\
\to\, &|e_s\rangle\,|0^k\rangle\,\underbrace{|e_s\rangle\cdots|e_s\rangle}_{k \text{ copies of } |e_s\rangle} && \text{(Lemma 14, depth } O(\log k)\text{, size } O(k2^k)) \\
\to\, &|e_s\rangle\,|s\rangle\,|e_s\rangle\cdots|e_s\rangle && \text{(Lemma 42, depth } O(k)\text{, size } O(k2^k)) \\
\to\, &|e_s\rangle\,|s\rangle\,|0^{k2^k}\rangle && \text{(Lemma 14, depth } O(\log k)\text{, size } O(k2^k))
\end{aligned}$$

The total depth and size of Step 2a are $O(\log k) + O(k) + O(\log k) = O(k)$ and $O(k2^k)$, respectively.

- Step 2b: Secondly, we implement Step 2b by a quantum circuit with $k2^k$ ancillary qubits of depth $O(k)$ and size $O(k2^k)$. Define an $(k+1)$-qubit quantum gate Tof_s acting on register B and the s-th qubit in register A:

$$\text{Tof}_s\,|x\rangle\,|j\rangle = |x \oplus \delta_{sj}\rangle\,|j\rangle\,, \;\; \text{for } s, j \in \{0\} \cup [2^k - 1],$$

where $|x\rangle$ is the s-th qubit in register A and $|j\rangle$ is in register B. That is, conditioned on the state in register B is $|s\rangle$, Tof_s flips the s-th qubit in register A. Step 2b is just $\prod_{s=0}^{2^k-1}\text{Tof}_s$.

Any Tof_s can be implemented by $[I\otimes(\otimes_{j=0}^{k-1}X^{s_j})]\text{Tof}_{2^k-1}[I\otimes(\otimes_{j=0}^{k-1}X^{s_j})]$, where Toffoli gate Tof_{2^k-1} can be implemented in depth $O(k)$ and size $O(k)$ without ancillary qubits (Lemma 41). Therefore Tof_s can be realized by an $O(k)$-depth and $O(k)$-size quantum circuit without ancillary qubits. To realize $\text{Tof}_0,\dots,\text{Tof}_{2^k-1}$ simultaneously, we make 2^k copies of register B in register C depth $O(k)$ and $O(k2^k)$. Then $\text{Tof}_0,\dots,\text{Tof}_{2^k-1}$ can be implemented simultaneously by using these copies. Finally, we reset register C back to 0 in depth $O(k)$ and size $O(k2^k)$. Step 2b can be summarized as follows:

$$\begin{aligned}
&|e_s\rangle|s\rangle|0^{k2^k}\rangle\\
\to\ &|e_s\rangle|s\rangle\underbrace{|s\rangle\cdots|s\rangle}_{2^k\text{ copies of }|s\rangle} && \text{(Lemma 14, depth } O(k)\text{, size } O(k2^k))\\
\to\ &|0^{2^k}\rangle|s\rangle|s\rangle\cdots|s\rangle && \text{(depth } O(k)\text{, size } O(k2^k))\\
\to\ &|0^{2^k}\rangle|s\rangle|0^{k2^k}\rangle && \text{(Lemma 14, depth } O(k)\text{, size } O(k2^k))
\end{aligned}$$

The total depth and size of Step 2b are $O(k)+O(k)+O(k)=O(k)$ and $O(k2^k)$, respectively.

- Step 2c: Thirdly, for Step 2c, we swap the first k qubits in register A and register B by k swap gates. Hence, Step 2c can be implemented in depth $O(1)$ and size $O(k)$.

Thirdly, for Step 3, we swap the first n qubit in register A and register B in depth $O(1)$ and size $O(n)$ without ancillary qubits by swap gates. □

L Circuit depth for general unitary synthesis

In Eq. (2), we called a (2×2)-block diagonal matrix V_j a j-qubit UCG. In the view of a circuit, this is a multiple controlled gate where the target qubit is the last one and the conditions are on the first $j-1$ qubits. But this target qubit can actually be any one, and all the implementations in Section 4 and Section 5 still apply. Let V_k^n denote an n-qubit UCG whose index of target qubit is k. By repeatedly applying cosine-sine decomposition, one can factor an arbitrary unitary matrix U into a sequence of UCGs as follows [27]. Recall that the Ruler function $\zeta(n)$ is defined as $\zeta(n)=\max\{k:2^{k-1}|n\}$.

Lemma 43. *Any n-qubit unitary matrix $U\in\mathbb{C}^{2^n\times 2^n}$ can be decomposed as $U=V_n^n(0)\cdot\prod_{i=1}^{2^{n-1}-1}V_{n-\zeta(i)}^n(i)\cdot V_n^n(2^{n-1})$, where different i in $V_k^n(i)$ denote different forms of n-qubit UCGs despite the same target qubit k.*

The proofs of Theorem 31 and Corollary 32 in Section 7 are shown as follows.

Theorem 31 *Any unitary matrix $U\in\mathbb{C}^{2^n\times 2^n}$ can be implemented by a quantum circuit of depth $O\Big(n2^n+\frac{4^n}{m+n}\Big)$ and size $O(4^n)$ with $m\le 2^n$ ancillary qubits.*

Proof. Based on Eq. (4), Lemma 10 and Lemma 11, given $m\le 2^n$ ancillary qubits, any n-qubit UCG $V_k^n(i)$ can be implemented by a circuit of size $O(2^n)$ and depth $O\Big(n+\frac{2^n}{n+m}\Big)$. Since Lemma 43 shows that any U can be decomposed into $O(2^n)$ many n-qubit UCGs, a circuit can simply implement them sequentially to realize U, yielding a circuit of size $O(2^n)\cdot O(2^n)=O(4^n)$ and depth $O(2^n)\cdot O\Big(n+\frac{2^n}{m+n}\Big)=O\Big(n2^n+\frac{4^n}{m+n}\Big)$. □

Corollary 32 *The minimum circuit depth $D_{\text{Unitary}}(n,m)$ for an arbitrary n-qubit unitary with m ancillary qubits satisfies*

$$\begin{cases}D_{\text{Unitary}}(n,m)=\Theta\Big(\frac{4^n}{m+n}\Big), & \textit{if } m=O(2^n/n),\\ D_{\text{Unitary}}(n,m)\in\big[\Omega(n),O(n2^n)\big], & \textit{if } m=\omega(2^n/n).\end{cases}$$

Proof. The lower bound for the number of CNOT gates is $\Omega(4^n)$ by a similar argument. The only difference is that with m ancillary qubits, the number of single-qubit gates right before the output is at most $n+m$ instead of n. We thus have $4k+3(n+m) \geq 4^n - 1$. When $m = O\big(2^n/n\big)$, this still gives $k = \Omega(4^n)$. Since each layer can have at most $(m+n)/2$ CNOT gates, we know that it needs at least $\Omega\big(\frac{4^n}{m+n}\big)$ depth for any circuit of n input qubits and m ancillary qubits. Theorem 3 shows a lower bound $\Omega(n)$ for an n-qubit QSP circuit, which is a special case of a circuit for an n-qubit unitary matrix. Therefore, we get a depth lower bound of $\Omega\big(\max\big\{n, \frac{4^n}{n+m}\big\}\big)$ for an n-qubit unitary matrix. Putting the depth upper bound $O\big(n2^n + \frac{4^n}{n+m}\big)$ and lower bound $\Omega\big(\max\big\{n, \frac{4^n}{n+m}\big\}\big)$ together, we complete the proof. □

M Decomposition with Clifford + T gate set

Lemma 44 ([56]). *For $\epsilon > 0$, any rotation $R_z(\theta) \in \mathbb{C}^{2\times 2}$ can be ϵ-approximated by a quantum circuit consisting of $O(\log(1/\epsilon))$ many H and T gates, without ancillary qubits.*

Based on this lemma, it is not hard to extend our results on the exact implementation of diagonal unitary matrix Λ_n to its approximate version (Lemma 45). This in turn gives approximate realization of UCGs V_n (Lemma 46), state preparation (Corollary 34), and unitary operation (Corollary 35).

Lemma 45. *Any n-qubit diagonal unitary matrix Λ_n can be ϵ-approximated by a quantum circuit of depth $O\left(n + \frac{2^n \log(2^n/\epsilon)}{m+n}\right)$, using the Clifford+T gate set with m ancillary qubits.*

Proof. In Section 4 and Section 5, our quantum circuits for Λ_n consist of only CNOT gates and $2^n - 1$ rotation gates $R(\alpha)$ for $\alpha \in \mathbb{R}$. By Lemma 44, every $R(\alpha)$ can be $(\epsilon/2^n)$-approximated by $O(\log(2^n/\epsilon))$ H and T gates up to a global phase. The overall accuracy of the circuit can then be seen from a union bound.

If $m \in [2n, 2^n]$, the total circuit depth of Λ_n is $O(\log m) + O(\log m + \log(2^n/\epsilon)) + O(\log m) + O(2^n \log(2^n/\epsilon)/m) = O(\log m + 2^n \log(2^n/\epsilon)/m) = O(n + 2^n \log(2^n/\epsilon)/m)$. If $m \leq 2n$, the depth of the circuit implementing the diagonal unitary matrix is $O(2^n \log(2^n/\epsilon)/n)$. Putting these two results together, we complete the proof. □

Corollary 46. *Any n-qubit UCG V_n can be ϵ-approximated by a quantum circuit using the Clifford+T gate set, of depth $O\left(n + \frac{2^n \log(2^n/\epsilon)}{m+n}\right)$ with m ancillary qubits.*

Proof. From Eq. (4), we can see that any n-qubit UCG can be decomposed into three n-qubit diagonal unitary matrices, two S gates and two H gates. By Lemma 45, every Λ_n can be $(\epsilon/3)$-approximated by a quantum circuit of depth $O\left(n + \frac{2^n \log(3\cdot 2^n/\epsilon)}{m+n}\right)$ with m ancillary qubits. Hence, V_n can be ϵ-approximated by a circuit of depth $3 \times O\left(n + \frac{2^n \log(3\cdot 2^n/\epsilon)}{m+n}\right) + 2 + 2 = O\left(n + \frac{2^n \log(2^n/\epsilon)}{m+n}\right)$. □

The approximate implementations of state preparation and general unitary matrix are shown in Corollary 34 and Corollary 35.

Corollary 34 *For any n-qubit target state $|\psi_v\rangle$ and $\epsilon > 0$, one can prepare a state $|\psi'_v\rangle$ which is ϵ-close to $|\psi_v\rangle$ in ℓ_2-distance, by a quantum circuit consisting of $\{CNOT, H, S, T\}$ gates of depth*

$$\begin{cases} O\left(\frac{2^n \log(2^n/\epsilon)}{m+n}\right) & \text{if } m = O(2^n/(n\log n)), \\ O(n \log n \log(2^n/\epsilon)) & \text{if } m \in [\omega(2^n/(n\log n)), o(2^n)], \\ O(n\log(2^n/\epsilon)) & \text{if } m = \Omega(2^n), \end{cases}$$

where m is the number of ancillary qubits.

Proof. According to Theorem 1, any n-qubit quantum state $|\psi_v\rangle$ can be prepared by a quantum circuit QSP, using single-gates and CNOT gates, of size $c \cdot 2^n$ for some constant $c > 0$ and depth

$$d = \begin{cases} O\left(\frac{2^n}{m+n}\right) & \text{if } m = O(2^n/(n\log n)), \\ O(n\log n) & \text{if } m \in [\omega(2^n/(n\log n), o(2^n)], \\ O(n) & \text{if } m = \Omega(2^n), \end{cases}$$

Based on Eq. (1) and Lemma 44, every single-qubit gate can be $(\epsilon/c2^n)$-approximated by a quantum circuit consisting of $O(\log((2^n)/\epsilon))$ Clifford+T gates. Approximate all single-qubit gates in this way. The depth of the new quantum circuit QSP' consisting of Clifford+T gates is $d \times O(\log(2^n/\epsilon)) = O(d\log(2^n/\epsilon))$. And circuit QSP' prepare a quantum state $|\psi'_v\rangle$ satisfying

$$\| |\psi_v\rangle - |\psi'_v\rangle \|_2 = \|(QSP - QSP')|0\rangle^{\otimes n}\|_2 \le \frac{\epsilon}{c2^n} \times c2^n = \epsilon.$$

□

Corollary 35 *Any n-qubit general unitary matrix can be implemented by a quantum circuit, using the* $\{CNOT, H, S, T\}$ *gate set, of depth* $O\left(n2^n + \frac{4^n\log(4^n/\epsilon)}{m+n}\right)$ *with m ancillary qubits.*

Proof. Lemma 43 shows that any $U \in \mathbb{C}^{2^n\times 2^n}$ can be decomposed into $2^n - 1$ n-qubit UCGs. According to Lemma 46, an n-qubit UCG V_n can be $\epsilon/(2^n - 1)$-approximated by a quantum circuit V'_n consisting of $\{CNOT, H, S, T\}$ gates in depth $O\left(n + \frac{2^n\log(4^n/\epsilon)}{m+n}\right)$ with m ancillary qubits. Hence, U can be ϵ-approximated by a quantum circuit in depth $O\left(n + \frac{2^n\log(4^n/\epsilon)}{m+n}\right) \times (2^n + 1) = O\left(n2^n + \frac{4^n\log(4^n/\epsilon)}{m+n}\right)$. □

N Sparse quantum state preparation

A vector $v = (v_0, v_1, \ldots, v_{2^n-1}) \in \mathbb{C}^{2^n}$ is said to be s-sparse if there are at most s nonzero elements in v. In this section, we consider how to efficiently prepare s-sparse states.

Lemma 47. *The unitary transformation defined by*

$$|x_1x_2\cdots x_n\rangle|t\rangle \to |x_1x_2\cdots x_n\rangle|\bigoplus_{i=1}^{n} x_i \oplus t\rangle \tag{32}$$

$\forall x_1, \ldots, x_n, t \in \{0, 1\}$, *can be implemented in depth* $O(\log(n))$.

Proof. The circuit implementation of Eq. (32) is shown in Figure 10. □

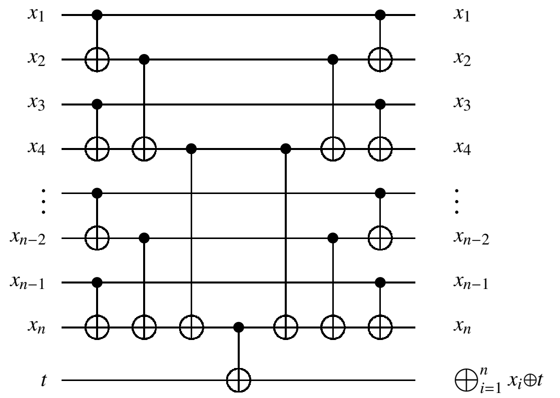


Figure 10: An $O(\log(n))$-depth circuit implementation of Eq. (32).

Lemma 48. *Suppose that we are given two sets $S_1 \subseteq \{0,1\}^{n_1}$ and $S_2 \subseteq \{0,1\}^{n_2}$, both of size s, and also given a bijection $P : S_1 \to S_2$. Then a unitary transformation satisfying*

$$|x\rangle\,|0^{n_2}\rangle \to |x\rangle\,|P(x)\rangle\,, \forall x \in S_1 \tag{33}$$

can be implemented by a quantum circuit of depth $O\Big(n_2\log(m) + \frac{(n_1+\log(m))sn_1n_2}{m}\Big)$, using m ($\geq 2n_1$) ancillary qubits.

Proof. Define an (n_1+1)-qubit unitary $\mathsf{Tof}_a^x\,|z\rangle\,|b\rangle := |z\rangle\,|a\cdot\delta_{xz}\oplus b\rangle$ for any $x,z\in\{0,1\}^{n_1}$ and $a,b\in\{0,1\}$, where $\delta_{xz}=1$ if $x=z$ and $\delta_{xz}=0$ if $x\neq z$. Unitary Tof_a^x can be implemented by an n_1-qubit Toffoli gate and at most $2n_1$ Pauli X gates. According to Lemma 41, Tof_a^x can be implemented in depth $O(n_1)$.

Let $\{x(1),x(2),\dots,x(s)\}$ be the elements of S_1, and $P(x(i))=y(i)$. Denote $y(i)=y_1(i)y_2(i)\cdots y_{n_2}(i)$, and we will compute $y(i)$ bit by bit.

We start by computing $y_1(i)$, the first bit of $y(i)$, and then all other bits can be computed similarly. More precisely, we aim to implement the following unitary transformation U_1:

$$|x(i)\rangle\,|0\rangle \xrightarrow{U_1} |x(i)\rangle\,|y_1(i)\rangle\,, \forall i\in[s].$$

Unitary U_1 can be implemented in 3 steps.

- Step 1: make $\frac{m}{2n_1}$ copies of $|x(i)\rangle$ using $\frac{m}{2}$ ancillary qubits, i.e., implement the following unitary transformation
$$|x(i)\rangle\,|0\rangle\,|0^{m/2}\rangle \to |x(i)\rangle\,|0\rangle\,|x(i)x(i)\cdots x(i)\rangle\,, \forall i\in[s].$$
This step can be implemented in depth $O(\log(m))$ by Lemma 14.

- Step 2: implement the following unitary transformation using $\frac{m}{2n_1}$ ancillary qubits
$$|x(i)\rangle\,|0\rangle\,|x(i)x(i)\cdots x(i)\rangle\,|0^{\frac{m}{2n_1}}\rangle \to |x(i)\rangle\,|y_1(i)\rangle\,|x(i)x(i)\cdots x(i)\rangle\,|0^{\frac{m}{2n_1}}\rangle\,, \forall i\in[s].$$
We divide set S_1 into $\frac{s}{m/2n_1}=\frac{2sn_1}{m}$ parts $S_1^{(1)},\dots,S_1^{(\frac{2sn_1}{m})}$. The size of each part is $\frac{m}{2n_1}$. For the first part $S_1^{(1)} := \{x(i) : i\in[\frac{m}{2n_1}]\}$, we implement the following unitary transformation:
$$|x(i)\rangle\,|0\rangle\,|x(i)x(i)\cdots x(i)\rangle\,|0^{\frac{m}{2n_1}}\rangle \to \begin{cases} |x(i)\rangle\,|y_1(i)\rangle\,|x(i)\cdots x(i)\rangle\,|0^{\frac{m}{2n_1}}\rangle\,, & \text{if } x(i)\in S_1^{(1)} \\ |x(i)\rangle\,|0\rangle\,|x(i)\cdots x(i)\rangle\,|0^{\frac{m}{2n_1}}\rangle\,, & \text{otherwise.}\end{cases} \tag{34}$$
Namely, we compute $y_1(i)$ for those $x(i)\in S_1^{(1)}$ (and keep other $x(i)$ untouched). This can be achieved by the following three sub-steps. In step 2.1, we apply unitaries $\mathsf{Tof}^{x(1)}_{y_1(1)}, \mathsf{Tof}^{x(2)}_{y_1(2)},\dots,\mathsf{Tof}^{x(\frac{m}{2n_1})}_{y_1(\frac{m}{2n_1})}$. For each $\mathsf{Tof}^{x(i)}_{y_1(i)}$, the control qubits are the i-th copy of $x(i)$ and the target qubit is the i-th qubit of ancillary qubits (those in $|0^{\frac{m}{2n_1}}\rangle$). Therefore, they can be realized in parallel by a circuit of depth $O(n_1)$. The effect of this step 2.1 is
$$|x(i)\rangle\,|0\rangle\,|x(i)\cdots x(i)\rangle\,|0^{\frac{m}{2n_1}}\rangle \to |x(i)\rangle\,|0\rangle\,|x(i)\cdots x(i)\rangle\,|0^{i-1}y_1(i)0^{\frac{m}{2n_1}-i}\rangle\,,\quad \forall x(i)\in S_1^{(1)}.$$
In step 2.2, we implement the following unitary transformation
$$|x(i)\rangle\,|0\rangle\,|x(i)x(i)\cdots x(i)\rangle\,|0^{i-1}y_1(i)0^{\frac{m}{2n_1}-i}\rangle \to |x(i)\rangle\,|y_1(i)\rangle\,|x(i)x(i)\cdots x(i)\rangle\,|0^{i-1}y_1(i)0^{\frac{m}{2n_1}-i}\rangle\,,\quad \forall x(i)\in S_1^{(1)}.$$
in depth $O(\log(m/(2n_1)))$ according to Lemma 47. In step 2.3, we restore the ancillary qubits by an inverse circuit of step 2.1 of depth $O(n_1)$, i.e., we realize the following unitary transformation:
$$|x(i)\rangle\,|y_1(i)\rangle\,|x(i)x(i)\cdots x(i)\rangle\,|0^{i-1}y_1(i)0^{\frac{m}{2n_1}-i}\rangle \to |x(i)\rangle\,|y_1(i)\rangle\,|x(i)x(i)\cdots x(i)\rangle\,|0^{\frac{m}{2n_1}}\rangle\,, \forall x(i)\in S_1^{(1)}.$$

We can verify that step 2.1, 2.2 and 2.3 realize unitary in Eq. (34) by circuit of depth $O(n_1 + \log(m/n_1)) = O(n_1 + \log(m))$. By the same discussion, for every $j \in [\frac{2sn_1}{m}]$,

$$|x(i)\rangle\,|0\rangle\,|x(i)x(i)\cdots x(i)\rangle\,|0^{\frac{m}{2n_1}}\rangle \to |x(i)\rangle\,|y_1(i)\rangle\,|x(i)x(i)\cdots x(i)\rangle\,|0^{\frac{m}{2n_1}}\rangle\,, \forall x(i) \in S_1^{(j)}. \tag{35}$$

can be implemented in depth $O(n_1 + \log(m))$. In summary, step 2 can be implemented in depth $O\Big(n_1 + \log(m)\Big) \cdot \frac{2sn_1}{m} = O\Big(\frac{(n_1+\log(m))sn_1}{m}\Big)$.

- Step 3: restore the ancillary qubits by an inverse circuit of step 1, of depth $O(\log(m))$.

In summary, unitary U_1 can be implement in depth $2 \cdot O(\log(m)) + O\Big(\frac{(n_1+\log(m))sn_1}{m}\Big) = O\Big(\log(m) + \frac{(n_1+\log(m))sn_1}{m}\Big)$. By similar discussion of U_1, for all $j \in [n_2]$, the following unitary U_j

$$|x(i)\rangle\,|0\rangle \xrightarrow{U_j} |x(i)\rangle\,|y_j(i)\rangle\,, \forall i \in [s].$$

can be also implemented in depth $O\Big(\log(m) + \frac{(n_1+\log(m))sn_1}{m}\Big)$. By applying $U_1, U_2, \ldots, U_{n_2}$, we compute all n_2 bits of $y(i)$. This implements unitary in Eq. (33) and the total depth is $n_2 \cdot O\Big(\log(m) + \frac{(n_1+\log(m))sn_1}{m}\Big) = O\Big(n_2\log(m) + \frac{(n_1+\log(m))sn_1n_2}{m}\Big)$. □

Lemma 49 ([57]). *Any n-qubit s-sparse quantum state can be prepared by a quantum circuit of size $O(ns)$, using no ancillary qubits.*

Theorem 50. *For any $m \geq 0$, any n-qubit s-sparse quantum state can be prepared by a quantum circuit of depth $O(n\log(sn) + \frac{s\log(s)n^2}{n+m})$, using m ancillary qubits.*

Proof. For simplicity, we assume $\log(s)$ is an integer and $n' = \log(s)$. Define $S_1 = \{0,1\}^{n'}$, and S_2 to be all s-sparse strings in $\{0,1\}^n$. Let P be any bijection from S_1 to S_2. Any n-qubit s-sparse quantum state can be represented as $|\psi\rangle = \sum_{x\in\{0,1\}^{n'}} v_x\,|P(x)\rangle$.

- **Case 1:** $m \geq 3n$**.** First, we prepare an n'-qubit quantum state

  $$|\psi'\rangle = \sum_{x\in\{0,1\}^{n'}} v_x\,|x\rangle\,,$$

  which can be implemented in depth $O((n')^2 + \frac{2^{n'}}{n'+m}) = O(\log^2(s) + \frac{s}{\log(s)+m})$ using m ancillary qubtis by Lemma 13.

  Second, we implement the following unitary transformations and then we complete preparing the state $|\psi\rangle$.

  $$\begin{aligned} |x0^{n-n'}\rangle\,|0^n\rangle &\to |x0^{n-n'}\rangle\,|P(x)\rangle & (36)\\ &\to |0^n\rangle\,|P(x)\rangle & (37)\\ &\to |P(x)\rangle\,|0^n\rangle\,, \forall x \in \{0,1\}^{n'}. & (38)\end{aligned}$$

  Based on Lemma 48, using m ancillary qubits, Eq. (36) can be implemented in depth $O(n\log(m) + \frac{(\log(s)+\log(m))s\log(s)n}{m})$. Eq. (37) can be viewed as a similar process by Lemma 48, though we need to swap S_1 and S_2 and reverse the direction of P. This can be implemented in depth $O(\log(s)\log(m) + \frac{(n+\log(m))s\log(s)n}{m})$, respectively. Eq. (38) can be implemented by n swap gates in depth 1. Therefore, if $m \leq \frac{s\log(s)n}{\log(s)+\log(n)}$, the total depth is $O(n\log(m) + \frac{(\log(s)+\log(m))s\log(s)n}{m}) + O(\log(s)\log(m) + \frac{(n+\log(m))s\log(s)n}{m}) + 1 = O(n\log(sn) + \frac{s\log(s)n^2}{m})$. If $m > \frac{s\log(s)n}{\log(s)+\log(n)}$, we use only $\frac{s\log(s)n}{\log(s)+\log(n)}$ ancillary qubits and the total depth is $O(n\log(sn))$. Combining these two cases, the total depth is $O(n\log(sn) + \frac{s\log(s)n^2}{m})$.

- **Case 2:** $m < 3n$**.** If $m < 3n$, we do not use ancillary qubits. According to Lemma 49, $|\psi\rangle$ can be prepared in depth $O(ns)$.

Combining the above two cases, the circuit depth for $|\psi\rangle$ is $O(n\log(sn) + \frac{s\log(s)n^2}{n+m})$. □