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

Xiaoming Sun, Senior Member, IEEE, Guojing Tian, Shuai Yang, Pei Yuan, and Shengyu Zhang

Abstract—The quantum state preparation problem aims to prepare an n-qubit quantum state \(|\psi_0\rangle = \sum_{k=0}^{2^n-1} v_k |k\rangle\) 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 article, we study quantum circuits when there are \(m\) ancillary qubits available. We construct, for any \(m\), circuits that can prepare \(|\psi_0\rangle\) in depth \(O(2^n/(m+n) + n)\) and size \(O(2^n)\), achieving the optimal value for both measures simultaneously. These results also imply a depth complexity of \(\Theta(4^n/(m+n))\) 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 \(O(2^n)\). 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 tradeoff bounds, which is 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.

Index Terms—Circuit depth, depth-space tradeoff, quantum circuit, state preparation, unitary synthesis.

I. 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], [3], [4], [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 (Q SVM) [13]. One of the challenges to fully exploit quantum algorithms for these tasks, however, is to efficiently prepare a starting state, 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 2-norm, i.e., \(\sum_{k=0}^{2^n-1} |v_k|^2 = 1\). The task is to generate a corresponding n-qubit quantum 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 tradeoff for quantum circuits. The reason is that in the past decade, we have witnessed a rapid development in qubit number and qubit lifetime, 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 \(10^3 - 10^5\) 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

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

2Take superconducting qubits, for example, the qubit number jumped from 5 in 2014 to 127 in 2021 [14], [15], [16], [17], [18], [19].
efficiently trade space for time in a quantum circuit? In this article, we will address this question in the fundamental tasks of QSP and general unitary circuit synthesis.

Let us first fix a proper circuit model. If we aim to generate the target state \(|\psi_t\rangle\) 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 \(|\psi_s\rangle\) 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 the last qubit conditioned on the first \(j-1\) qubits being a certain computational basis state. This type of multiple-controlled \((2 \times 2)\)-unitary can be implemented in depth \(O(n)\) without ancillary qubit,\(^4\) yielding a depth upper bound of \(O(n^2)\) for the QSP problem. Bergholm et al. [25] 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 \((23/24)2^n - 2^{(n/2)+1} + (5/3)\) for even \(n\), and \((115/96)2^n\) 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 \((23/48)2^n\) for even \(n\) and \((115/192)2^n\) for odd \(n\). All these results are about the exact QSP 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 \(|\psi'_s\rangle\) can be implemented by an \(O(n^2)\)-depth quantum circuit with sufficiently many ancillary qubits,\(^5\) where \(|\|\psi'_s\rangle - |\psi_s\rangle|| \leq \epsilon\). Though QSP is only used as a tool for their main topic of the parallel quantum walk, their concluding section did call for studies on the tradeoff 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 \(\sum_{k=0}^{2^n-1} v_k |k\rangle\), but in the unary encoding \(\sum_{k=0}^{2^n-1} v_k |\bar{e}_k\rangle\), where \(e_j \in \{0,1\}\) is the \(k\)th bit being 1 and all other bits being 0. This article 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], [32], [33], [34]. Johri et al. [30] 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 \(O((n/d)2^{n-n/d})\) 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 article, we tightly characterize the depth and size complexities of the QSP 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 is available.

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

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

These depth bounds improve the depth of \(O(2^n)\) in [25] and [26] by a factor of \(m\) for any \(m \in [2n, O(2^n/\log n)]\), 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 \(|\psi_t\rangle\) 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 \(\Omega(2^n/(n+m))\) for circuits with \(m\) ancillary qubits. This bound deteriorates to 0 as \(m\) grows to infinity. Aharonov and Touati [35] gave a depth lower bound of \(\Omega(\log n)\) for a 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 \(\Omega(\max(n, 2^n/(m+n)))\), for circuits using arbitrary single-qubit and 2-qubit gates.

The proof of Theorem 3 is shown in [36, Appendix A].

---

\(^3\)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.
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 1 and illustrated in Fig. 1.

**Corollary 1:** For a circuit preparing an \( n \)-qubit quantum state with \( m \) ancillary qubits, the minimum size is \( \Theta(2^n) \), and the minimum depth \( D_{\text{QSP}}(n, m) \) for different ranges of \( m \) are characterized as follows:

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

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. [37] gave an upper bound \( O(n^3 4^n) \) for the number of CNOT gates for arbitrary \( n \)-qubit unitary matrix. Knill [38] improved the upper bound to \( O(n4^n) \). Vertaainen et al. [39] constructed a quantum circuit for an \( n \)-qubit unitary matrix with \( O(4^n) \) CNOT gates. Mottonen and Vertaainen [27] designed a quantum circuit of depth \( O(4^n) \) using \((23/48)4^n\) CNOT gates. The best known lower bound for \textit{number} of CNOT gates is \([(1/4)(4^n - 3n - 1)] \) [40], 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 4:** 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 \( O(n2^n + (4^n/(m + n))) \) 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 2:** For any \( n \)-qubit target state \( |\psi_v\rangle \), one can prepare a state \( |\psi_v'\rangle \) which is \( \epsilon \)-close to \( |\psi_v\rangle \) in \( \ell_2 \)-distance, by a circuit consisting of \{CNOT, H, S, T\} gates of depth

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

using \( m \) ancillary qubits.

1) **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 [41] 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 an 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 \( 2^n - 1 \) suffixes as in the previous case, where \( r_1 \approx n/2 \) 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 \( F_2 \). 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 \(|e_i\rangle \rightarrow |e_i\rangle |e_i\rangle \), and then to the binary encoding. This gives the optimal depth and size for the regime \( m \geq 2^n \). For \( m \in \omega\left(\frac{2^n}{n^2}\right), o(2^n) \), 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 $m \in [O(2^n/n^2), O(2^n/(n \log n))]$, the overall depth is asymptotically optimal, leaving a gap $[\Omega(n), O(n \log n)]$ only when $m$ is in a small range $[O(2^n/(n \log n)), O(2^n)]$.

2) Other Related Work: Besides the standard QSP, researchers have also studied some relaxed versions. Araujo et al. [42] have given a depth upper bound of $O(n^2)$ to prepare a state $\sum_{k=0}^{2^n-1} |v_k| |\text{garbage}_k\rangle$, 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 $[H, S, T, \text{CNOT}]$ gates to generate $|\psi_i\rangle$ satisfying $\||\psi_i\rangle - |\psi_t\rangle\| \leq \epsilon$ for various distance measures $\|\cdot\|$. Previous attention was paid to minimizing the number and depth of $T$ gates [43], [44], which is non-Clifford and usually thought to be hard to realize experimentally [43]. They have applied ancillary qubits to implement a circuit such that the number of $T$ gates can be optimized to $O(2^n/\lambda) + \lambda \log^2(2^\lambda/\epsilon)$ [43], 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 $O(n + \log(1/\epsilon))$.

3) Subsequent Work: After this work appeared on arXiv [36], Rosenthal [45] constructed a QSP circuit of depth $O(n)$, using $O(n^{2n})$ ancillary qubits, as opposed to ours that only uses $O(2^n)$ ancillary qubits. Rosenthal also presented a circuit for the general unitary synthesis of depth $O(2^n/\epsilon^2)$ using $O(4^n)$ 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 case of our results. This line of research culminating at [47], which completely solves the circuit depth and size complexities even for a generalized problem of QSP, where an arbitrary number of controlled qubits are allowed. In a different vein, [48] takes qubit connectivity into consideration, and studies circuit complexity for QSP and general unitary synthesis problems with various families of graphs as constraints on what pairs of qubits a two-qubit gate can apply on.

4) Organization: The remainder of this article is organized as follows. In Section II, we will review notations and a framework of QSP. Then, we will present how to decompose the uniformly controlled gate (UCG) to diagonal unitary matrices and show the depth of QSP when the number of ancillary qubits $m = O(2^n/n^2)$ in Section III. Next, we will show two quantum circuits for diagonal unitary matrices used in the previous section, with and without ancillary qubits in Section IV and Section V, respectively. Furthermore, we present a new circuit framework for QSP when $m = \Omega(2^n/n^2)$ in Section VI. In Section VII, we will show some extensions and implications of the above bounds. Finally, we conclude in Section VIII. Due to the page limit, some proofs are omitted. They can be found in the appendices in the arxiv version of this work [36].

II. PRELIMINARIES

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

1) Notation: Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. All logarithms $\log(\cdot)$ are base 2 in this article. Let $I_n \in \mathbb{R}^{2^n \times 2^n}$ be the $n$-qubit identity operator. Denote by $\mathbb{F}_2$ the field with two elements, with multiplication $\cdot$ and addition $\oplus$, which can be overloaded to vectors: $x \oplus y = (x_1 \oplus y_1, x_2 \oplus y_2, \ldots, x_n \oplus y_n)^T$ for any $x, y \in \mathbb{F}_2^n$. The inner product of two vectors $s, x \in \mathbb{F}_2^n$ is $(s, x) := \oplus_{i=1}^n s_i \cdot x_i$ 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 $U_1, \ldots, U_k \in \mathbb{C}^{2^{(k)}}$, $\text{diag}(U_1, U_2, \ldots, U_k)$ is defined as $\text{diag}(U_1, \ldots, U_k) = \begin{bmatrix} U_1 & \cdots & U_k \end{bmatrix} \in \mathbb{C}^{2^k \times 2^k}$.

2) Elementary Gates: We will use the following $R_\gamma(\theta), R_x(\theta)$ and $R(\theta)$ to denote 1-qubit rotation (about y-axis and z-axis) gates and phase-shift gate, i.e.,

$$R_\gamma(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix},$$

$$R_x(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta/2} \end{bmatrix},$$

$$R(\theta) = \begin{bmatrix} 1 \\ e^{i\theta/4} \end{bmatrix},$$

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

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. $$

The 2-qubit controlled-NOT gate is

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

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

3) Single-Qubit Gate Decomposition: Any single-qubit operator $U \in \mathbb{C}^{2 \times 2}$ can be decomposed as $U = e^{i\gamma} R_x(\beta) R_y(\gamma) R_z(\delta)$ 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 $R_y(\gamma) = \text{SHR}_x(\gamma) H^T$, 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

$$U = e^{i\gamma} R_x(\beta) \text{SHR}_x(\gamma) H^T R_z(\delta). \quad (1)$$

4) 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 [49], [50], [51], 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 $x^1, x^2, \ldots, x^{2^n}$.
and we will construct them one by one. Take \( x^i = 0^\theta \). For each 
\( i = 1, 2, \ldots, 2^n - 1 \), the next string \( x^{i+1} \) is obtained from \( x^i \) by 
flipping the \( \theta \)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 1:** The RBC 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 \( i \) appears 
\( 2^{n-i} \) 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 a 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 \( k, k+1, \ldots, n, 1, 2, \ldots, k-1 \) to appear \( 2^i, 
2^{i-1}, 2^{i-2}, \ldots, 2^2, 2 \) 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.

### III. Quantum State Preparation With \( O(2^n/n^2) \) 
Ancillary Qubits

In this section, we will review a natural framework of algo-
rithm for QSP, first appeared in [22]. Our results presented in 
Sections IV and V, which achieve the optimal circuit depth, 
also fall into this framework. The framework to prepare an \( n \)-
qubit quantum state is depicted in Fig. 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

\[
V_j = \text{diag}(U_1, U_2, \ldots, U_{2^{j-1}}) \in \mathbb{C}^{2^j \times 2^j}
\]

(2)

where each \( U_l \) is a \( 2 \times 2 \) 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 Fig. 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 UCG. We give a specific example for 
illustration in [36, Appendix B].

Thus the depth of the circuit for QSP in the above 
framework crucially depends on the circuit depth of the 
implementation of \( V_j \)’s.

**Lemma 2:** 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 I, if we implement each \( V_j \) directly 
as in [22], then the whole QSP circuit has a depth of \( \Theta(n2^n) \), 
which is suboptimal 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 the efficient implemen-
tation 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 VI.

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

\[
\Lambda_n = \text{diag}(1, e^{i\theta_1}, e^{i\theta_2}, \ldots, e^{i\theta_{2^n-1}}) \in \mathbb{C}^{2^n \times 2^n}.
\]

(3)

**Lemma 3:** If one can implement \( \Lambda_n \) in (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 (1), each unitary matrix \( U_k \in \mathbb{C}^{2^k \times 2^k} \) 
can be decomposed as \( U_k = e^{i\alpha_k} R_z(\beta_k) SHR_z(\gamma_k) HS R_z(\delta_k) \). Then, 
the UCG \( V_n \) can thus be decomposed to

\[
V_n = \text{diag}(e^{i\alpha_1}, \ldots, e^{i\alpha_{2^n-1}}) \otimes \mathbb{I}_1 \\
\cdot \text{diag}(R_z(\beta_1), \ldots, R_z(\beta_{2^n-1})) \\
\cdot \text{diag}(R_z(\gamma_1), \ldots, R_z(\gamma_{2^n-1})) \\
\cdot \text{diag}(R_z(\delta_1), \ldots, R_z(\delta_{2^n-1}))
\]

(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 (3). If \( \Lambda_n \) 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 (3). We will prove the following lemmas in 
Sections IV and V.

**Lemma 4:** For any \( m \in [2n, 2^n/n] \), any diagonal unitary 
matrix \( \Lambda_n \in \mathbb{C}^{2^n \times 2^n} \) as in (3) can be implemented by a quantum 
circuit of depth \( O(\log m + (2^n/m)) \) and size \( O(2^n) \), with 
m ancillary qubits.
Lemma 5: Any diagonal unitary matrix $\Lambda_n \in \mathbb{C}^{2^n \times 2^n}$ as in (3) can be implemented by a quantum circuit of depth $O(2^n/n)$ and size $O(2^n)$ without ancillary qubits.

Lemmas 4 and 5 imply Lemma 6.

Lemma 6: For $m \geq 0$, any UCG $V_n \in \mathbb{C}^{2^n \times 2^n}$ as in (2) can be implemented by a quantum circuit of depth $O(n + (2^n/[n + m]))$ and $O(2^n)$ with $m$ ancillary qubits.

Proof: According to (4), every $V_n$ can be decomposed into three $n$-qubit diagonal unitary matrices and four single-qubit gates. Combining with Lemma 4 and Lemma 5, $V_n$ can be realized by a quantum circuit of depth $O(n + (2^n/[n + m]))$ 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 7 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 3 and will be treated in Section VI.

Lemma 7: For any $m \geq 0$, any $n$-qubit quantum state $|\psi_n\rangle$ 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 $O(n^2 + (2^n/[m + n]))$.

Proof: We prove the case $m = 0$ first. Plugging Lemma 5 into Lemma 3, we get a circuit solving QSP in size $\sum_{j=1}^{n} O(2^j) + 2n + 1 = O(2^n)$ and depth $O(\sum_{j=1}^{n} (2^j/j) + n) = O(\sum_{j=1}^{n} [\log(j)] (2^j/j) + \sum_{j=n+1}^{n} [\log(j)] (2^j/j)) = O(\sum_{j=1}^{n} [\log(j)] 2^j + \sum_{j=n+1}^{n} [\log(j)] 1) = O(2^n/n)$, as desired.

Now we prove the case $m > 0$. If $1 \leq m \leq 2n$, we will not use the ancillary qubits—we just invoke Theorem 2 to obtain a circuit of depth $O(2^n/n)$. If $2n \leq m \leq 2n^2$, we can combine Lemmas 3 and 4 to give a circuit of size $3\sum_{j=1}^{n} O(2^j) + 2n + 1 = O(2^n)$ and depth $O(\sum_{j=1}^{n} (\log(m + (2^j/m)) + n) = O(n^2 + (2^n/m))$. If $m > 2n^2$, we only use the first $2n^2$ ancillary qubits, then the upper bound of $O(n^2)$ 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 $O(n^2 + (2^n/[m + n]))$.

Next let us consider how to efficiently implement $\Lambda_n$, 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 [41], [52], [53]. Bullock and Markov [41] used 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 IV). Interestingly, this approach turns out to inspire our construction for circuits without ancillary qubits (Section V), 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 $s \in [0,1]^n = \{0^n, 1^n\}$, make a phase shift of $\alpha_s$ on each basis $|x\rangle$ when $\langle s, x \rangle = 1$ (recall that $\langle s, x \rangle$ is over $\mathbb{Z}_2$), i.e.,

$$|x\rangle \rightarrow e^{i\alpha_s(s, x)} |x\rangle.$$  \hspace{1cm} (5)

2) Find $\{\alpha_s : s \in [0,1]^n - \{0^n\}\}$ s.t.

$$\sum_{s \in [0,1]^n - \{0^n\}} \alpha_s(s, s) = \theta(x) \ \forall x \in [0,1]^n - \{0^n\}.$$  \hspace{1cm} (6)

Then, we get

$$|x\rangle \rightarrow \prod_{s \in [0,1]^n - \{0^n\}} e^{i\alpha_s(s, x)} |x\rangle = e^{i\Sigma_{s} \alpha_s(s, x)} |x\rangle = e^{i\theta(x)} |x\rangle$$ as required in $\Lambda_n$. For notational convenience, we define $\alpha_0^n = 0$. The implementations of above two tasks in (5) and (6) are accomplished in [36, Appendix C].

IV. DIAGONAL UNITARY IMPLEMENTATION WITH ANCILLARY QUBITS

In this section, we prove Lemma 4. 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 (3) can be implemented by a quantum circuit of depth $O(\log m + (2^n/m))$ 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 $|s, x\rangle$ 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 $\Lambda_4$ by using eight ancillary qubits is shown in [36, Appendix D].

We will show how to implement $\Lambda_n$ 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 Fig. 3. Our framework consists of three registers and five stages. The first $n$ qubits labeled as $x_1, x_2, \ldots, x_n$ form the input register, the next $(m/2)$ qubits are the copy register, and the last $(m/2)$ 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 $\langle s_1 0 \ldots 0, x \rangle$ on each qubit in the phase register, and iterate other nonzero 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 $|0\rangle$.

1) Stage 1 (Prefix Copy): In this stage, we make $|\text{m/2}\rangle$ copies of each qubit $x_1, x_2, \ldots, x_n$ in the input register, where $t = |\log (m/2)| < n$. More formally, the circuit implements the unitary $U_{\text{copy,1}}$ which operates on the input and copy registers only. Its effect is

$$|x\rangle |0\rangle^{2n} \mapsto U_{\text{copy,1}} |x\rangle |\text{x_pre}\rangle$$  \hspace{1cm} (7)

where the two parts in the ket notation are for the input and copy register, respectively, and $|x\rangle = |x_1 x_2 \ldots x_n\rangle$, $|\text{x_pre}\rangle =$
Fig. 3. Framework for the circuit of $\Lambda_n$ with $m$ ancillary qubits. The first $n$ qubits $\{x_1, \ldots, x_n\}$ form the input register, the next $(m/2)$ qubits the copy register and the last $(m/2)$ 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 $O(\log m)$, $O(\log m)$, $O(\log m)$, $O(2^{\ell}/m)$, and $O(\log m + (2^\ell/m))$, respectively.

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

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

Lemma 10: The Gray Initial Stage can be implemented in depth at most $2\log m$ and in size at most $((n+1)m)/2$ such that its unitary $U_{\text{GrayInit}}$ satisfies

$$|x\rangle|x_{\text{prec}}\rangle|0^{m/2}\rangle \xrightarrow{U_{\text{GrayInit}}} e^{i\sum_{j(t)} f_j(x) \alpha_{s(j,1)}} |x\rangle|\tilde{x}_{\text{prec}}\rangle |f_{(\ell,1)}\rangle,$$

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

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

$$|x\rangle|x_{\text{prec}}\rangle|0^{m/2}\rangle \xrightarrow{U_1} |x\rangle|x_{\text{prec}}\rangle |f_{(\ell,1)}\rangle,$$

$$R_1 \overset{\text{def}}{=} e^{i\sum_{j(t)} f_j(x) \alpha_{s(j,1)}} |x\rangle|x_{\text{prec}}\rangle |f_{(\ell,1)}\rangle.$$ 

Now let us construct a shallow circuit for the first step $U_1$. Recall that we have $\ell = 2^\ell$ 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 $x_1 \oplus \cdots \oplus x_t$ ($t' \leq t$), we will use CNOT gates to copy the qubits $x_1, \ldots, x_t$ 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 $[2^\ell/[t|m/2(t')]]$ mini-steps, each mini-step handling $t|m/2(t')$ qubits $j$ by assigning the state $(s(j,1),x)$ to it. Since we have $\ell = 2^\ell$ qubits to handle, it needs $[2^\ell/[t|m/2(t')]]$ mini-steps.

For all positions $i \in [t]$ with $s(i,1) = 1$, we use CNOT gate to copy $x_i$ to qubit $j$. We have $t$ variables $x_1, \ldots, x_t$, each with $|m/2(t)|$ copies. To utilize these copies for parallelization, we break the $t|m/2(t')$ target qubits into $t$ blocks of size $|m/2(t')|$ each. Each mini-step gives all needed variables for $t(|m/2(t')| + 1)$ qubits $j$ in depth $t$. In the first layer, we use the $|m/2(t)|$ copies of $x_1$ as control qubits in CNOT to copy $x_1$ to the first block of target qubits $j$, use the $|m/2(t)|$ 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 $t|m/2(t')$ many qubits $j$ get their needed variables.

Since there are $[2^\ell/[t|m/2(t')]]$ mini-steps, each of depth $t$, the total depth for $U_1$ is $[2^\ell/[t|m/2(t')]] \cdot t \leq (m/2)/|m/2(t')| + t = 2t = 2\log(m/2) \leq 2\log m - 2$. 

[36, Appendix E]
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 \(R_k\) gate on it.

3) Stage 3 (Suffix Copy): In this stage, we first undo \(U_{\text{copy},1}\) and then make \([(m/2)[n-(n-1)])]\ copies of each suffix variable, namely \(x_{t+1}, \ldots, x_n\). The next lemma is similar to Lemma 8 and we omit the proof.

Lemma 11: We can make \([(m/2)[n-(n-1)])]\ copies of each qubit \(x_{t+1}, x_{t+2}, \ldots, x_n\) in the input register and the copy register, by applying on \(|\psi\rangle|0\rangle^m/2\) an \(m\)-size circuit \(U_{\text{copy},2}\) of CNOT gates only, in depth at most \(\log m\).

Define

\[
|x_{\text{surf}}\rangle \overset{\text{def}}{=} |x_{t+1} \cdots x_{t+n} 0 \cdots 0\rangle,
\]

then the effect of \(U_{\text{copy},2}\) is \(|\psi\rangle|0\rangle^m/2 \xrightarrow{U_{\text{copy},2}} |x\rangle|x_{\text{surf}}\rangle\). The operator of this stage is \(U_{\text{copy},2}U_{\text{copy},1}\), and the depth is at most \(2 \log m\) and the size is at most \(m\). The effect of this stage \(U_{\text{copy},2}U_{\text{copy},1}\) is

\[
|x\rangle|x_{\text{surf}}\rangle \xrightarrow{U_{\text{copy},1}} |x\rangle|0\rangle^m/2 \xrightarrow{U_{\text{copy},2}} |x\rangle|x_{\text{surf}}\rangle.
\]

4) 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 \(x_{r(t-1)}, \ldots, x_{r(0)}\), 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 12: The phase \(k\) of the Gray Path Stage implements

\[
|x\rangle|x_{\text{surf}}\rangle \xrightarrow{U_k} |x\rangle|x_{\text{surf}}\rangle \xrightarrow{R_k} e^{\sum_{j \in \ell} \overline{f_{j,k}}(x) \alpha_{s(j,k)}} |x\rangle|x_{\text{surf}}\rangle
\]

where \(f_{j,k}(x) = \langle s(j, k), x \rangle\) and \(|\overline{f_{j,k}}(x)\rangle = \otimes_{j \in \ell} \overline{f_{j,k}(x)}\). The phase \(k\) of the Gray Path Stage are at most \(2 \log m\) and \(2^n+1\).

Proof: The operation can be easily seen in a similar way as that for Lemma 10. 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 9, one CNOT gate suffices to implement the function \((x, s(j,k))\) from \((x, s(j,k-1))\) in the previous phase: The control qubit is \(x_{t+j-1}\) and the target qubit is \(j\). Moreover, the third property in Lemma 9 shows that each variables \(x_{t+1}\) is used as a control qubit for at most \([(m/2[n-(n-1)])] + 1\) different \(j \in \ell\). Since we have \([(m/2[n-(n-1)])] + 1\) copies in the input register and the copy register, these CNOT gates in step k.1 can be implemented in depth 1.

The phase 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 \(2^n/\ell \cdot (1 + 1) \leq 2 \cdot 2^n/\ell\).

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 \(2^n+1\).

5) Stage 5 (Inverse): In this stage, the circuit applies \(U_{\text{copy},1}^\dagger U_{\text{copy},1}U_{\text{copy},2}^\dagger U_{\text{copy},2} \cdots U_{\text{copy},t}^\dagger U_{\text{copy},t}^\dagger\).

Lemma 13: The depth and size of the Inverse Stage are at most \(O(\log m + 2^n/m)\) and \(m^2 + mn + m + 2^n = 2^n + (3m + mn)/2\). The effect of this stage is

\[
|x\rangle|x_{\text{surf}}\rangle \xrightarrow{U_{\text{copy},t}^\dagger \cdots U_{\text{copy},1}^\dagger} |x\rangle|0\rangle^m/2 \xrightarrow{U_{\text{copy},t} \cdots U_{\text{copy},1}} |x\rangle|0\rangle^m/2.
\]

The proof of Lemma 13 is shown in [36, 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 14: The circuit implements the operation in (3) in depth \(O(\log m + 2^n/m)\) and in size \(3 \cdot 2^n + mn + (7/2)m\).

The proof of Lemma 14 is shown in [36, Appendix G]. In summary, \(\Lambda_n\) can be implemented in \(O(\log m + (2^n/m))\) depth and size \(3 \cdot 2^n + mn + (7/2)m\) with \(m \in [2n, 2^n/n]\) ancillary qubits, proving Lemma 4.

V. DIAGONAL UNITARY IMPLEMENTATION WITHOUT ANCILLARY QUBITS

In this section, we prove Lemma 5. That is, any diagonal unitary \(\Lambda_n \in \mathbb{C}^{2^n \times 2^n}\) as in (3) can be implemented by a quantum circuit of depth \(O(2^n/n)\) and size \(O(2^n)\) without ancillary qubits. In Section V-A, 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 V-B. Finally, we give the detailed construction of some operators in Section V-C.

A. Framework and Functionalities

The framework of our circuit implementing \(\Lambda_n\) is a recursive procedure shown in Fig. 4.

The \(n\)-qubit work register is divided into two registers: 1) a control register consisting of the first \(r_t\) qubits and 2) a target register consisting of the last \(r_t\) qubits. The circuit has the following components.
1. A sequence of $n$-qubit unitary operators $G_1, \ldots, G_r$, the detailed construction of which will be given in Section V-C.
2. An $r_1$-qubit unitary operator $R$, which resets the state in the target register to the input value $|x_{r_1+1}, \ldots, x_n\rangle$.
3. An $r_c$-qubit diagonal unitary operator $\Lambda_{r_c}$, which is implemented recursively.

The parameters are set as follows. $r_1 = \lfloor n/2 \rfloor \approx n/2$, $r_c = n - r_1 \approx n/2$, and $\ell \leq \lfloor (2^{n+2})/(r_1 + 1) \rfloor - 1 \approx (2^{n+2}/3)/n$.

Next we describe the function of each operator in Fig. 4, for which it suffices to specify their effects on an arbitrary computational basis state $|x\rangle = |x_1x_2 \cdots x_{r_c}x_{r_c+1} \cdots x_n\rangle = |\chi_{\text{control}}\rangle |x_{\text{target}}\rangle$, 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 IV, an $n$-bit string $s \in \{0, 1\}^n - \{0^n\}$ is broken into two parts, a $\lfloor \log(m/2) \rfloor$-bit prefix and an $(n - \lfloor \log(m/2) \rfloor)$-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 IV, $s \in \{0, 1\}^n - \{0^n\}$ is generated in the phase register, which is initialized to $|0\rangle$. In this section, $s = ct$, in which $c$ is the $r_1$-bit prefix and $t$ is the $r_2$-bit suffix. The state $|c, x\rangle$ 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 $t \neq 0^r$ and then generate $c0^r$ recursively.

2. In Section IV, there are $2^{\lfloor \log(m/2) \rfloor} \leq (m/2)$ prefixes which can be enumerated in $(m/2)$ qubits in phase register exactly. In this section, $2^{r_1} - 1 \approx 2^{n/2}$ suffixes should be generated in $r_1$ qubits in target register. As we only have $r_1$ qubits, the small space is insufficient to enumerate all $2^{r_1} - 1$ suffixes. Thus, we need to enumerate them in many stages, and $r_2$ 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 $\{0, 1\}^{r_2}$. Thus, we need carefully divide $2^{r_1} - 1$ suffixes into $\ell$ sets $T^{(1)}$, $\ldots$, $T^{(\ell)}$ with $T^{(k)} = \{t^{(k)}_1, t^{(k)}_2, \ldots, t^{(k)}_{r_2}\}$ each $t^{(k)}_a \neq 0^r$ for $a \in [r_1]$ and $k \in [\ell]$, and the strings in each $T^{(k)}$ linearly independent. We allow overlap between these sets, but maintain the total number $\ell$ of sets only a constant times of $(2^{r_1} - 1)/r_1$, 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 $T^{(1)}, T^{(2)}, \ldots, T^{(\ell)}$ satisfying the following two key properties.

1. For each $k \in [\ell]$, the set $T^{(k)} = \{t^{(k)}_1, t^{(k)}_2, \ldots, t^{(k)}_{r_2}\}$ contains $r_1$ vectors from $\{0, 1\}^{r_1}$ that are linearly independent over the field $\mathbb{F}_2$.
2. The construction of sets $T^{(1)}, \ldots, T^{(\ell)}$ are shown in [36, Appendix H]. For each $k \in [\ell] \cup \{0\}$, define an $r_1$-qubit state

$$|y^{(k)}\rangle = |y_1^{(k)}, y_2^{(k)}, \ldots, y_{r_1}^{(k)}\rangle$$

where $y_j^{(k)} = \begin{cases} x_{r_c+j}, & \text{if } k = 0 \\ 0^r & , & \text{if } k \in [\ell]. \end{cases}$ (14)

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

$$F_1 = \{ct : t \in T^{(1)}, c \in \{0, 1\}^{r_1}\}$$

$$F_k = \{ct : t \in T^{(k)}, c \in \{0, 1\}^{r_1}\} - \bigcup_{d \in [k-1]} F_d, \quad 2 \leq k \leq \ell. \quad (15)$$

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

$$\bigcup_{k \in [\ell]} F_k = \{0, 1\}^{r_1} \times \bigcup_{k \in [\ell]} T^{(k)} = \{0, 1\}^{r_1} \times \{0^{r_1} : c \in \{0, 1\}^{r_1}\}.$$ (16)

With the above concepts, we can now show the desired effect of the operators $G_k$, $R$, and $\Lambda_{r_c}$.

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

$$G_k |\chi_{\text{control}}\rangle |y^{(k-1)}\rangle = e^{i \sum_{x \in [0,1]^{r_1}} \alpha_x |\chi_{\text{control}}\rangle |y^{(k)}\rangle} \quad (17)$$

where $\alpha_x$ is determined by (6). In words, $G_k$ has two effects: a) it puts a phase and b) it transits from the stage $k - 1$ to the stage $k$.

2. The transformation $R$ acts on the target register and resets the suffix state as follows:

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

As a map on $\{0, 1\}^{r_1}$ (instead of $|x\rangle : x \in \{0, 1\}^{r_1}\}$, $R$ is an invertible linear transformation over $\mathbb{F}_2$.

3. The operator $\Lambda_{r_c}$ is an $r_c$-qubit diagonal matrix satisfying that

$$\Lambda_{r_c} |\chi_{\text{control}}\rangle = e^{i \sum_{x \in [0,1]^{r_c}, y \neq [0^r]} (\alpha \circ \beta)_{x,y} |\chi_{\text{control}}\rangle}\quad (19)$$

and will be implemented recursively.

We will define these operators and show these properties in Section V-C.
B. Correctness and Depth

In this section, we will prove the correctness and analyze the depth of invertible linear transformation from [54] (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 [55]. We thus have the following result.

**Lemma 15:** Suppose that \( U \in \mathbb{F}_2^{2^n \times 2^n} \) is an invertible linear transformation over \( \mathbb{F}_2 \). Then as a \( 2^n \times 2^n \) unitary matrix which permutes computational basis \( \{|x\} : x \in \{0,1\}^n \} \), the map \( U \) can be realized by a CNOT circuit of depth at most \( O(n/\log n) \) and size at most \( O(n^2/\log n) \) without ancillary qubits.

As mentioned in Section V-A, \( R \) is an invertible linear transformation on the computational basis variables, thus the above lemma immediately implies the following depth upper bounds for \( R \).

**Lemma 16:** The operator \( R \) can be realized by an \( O((r_\ell/\log r_\ell) \cdot \log n) \)-depth and \( O(r_\ell^2/\log r_\ell) \)-size CNOT circuit without ancillary qubits.

The depth of \( G_0 \) will be easily seen from its construction in Section V-C.

**Lemma 17:** The operator \( G_0 \) can be realized by an \( O(2^r) \)-depth and \( O(r \cdot 2^{r+1}) \)-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 Fig. 4 is shown in [36, Appendix I].

**Lemma 18:** Any diagonal unitary matrix \( \Lambda_n \) can be realized by the quantum circuit \( (\Lambda_n \otimes R)G_0G_{\ell_1}G_{\ell_1-1} \cdots G_{\ell_1} \) as in Fig. 4, which has depth \( O(\log n) \) and size \( 2n^3 + O(n^2/\log n) \) and uses no ancillary qubits.

**Proof:** We prove that the circuit has depth \( D(n) = O(2^n/n) \). Lemma 17 shows \( G_0 \) can be realized in depth at most \( 2^r \) for a constant \( \lambda_1 > 0 \) and Lemma 16 shows \( R \) can be implemented in depth at most \( 2\lambda_2 \cdot r_\ell/\log r_\ell \) without ancillary qubits for a constant \( \lambda_2 > 0 \). Therefore, \( D(n) \) satisfies the following recurrence:

\[
D(n) \leq \max \left \{ D(r_\ell), \lambda_2 \cdot \frac{r_\ell}{\log r_\ell}, \lambda_1 \cdot 2^r \cdot \ell \right \} + \lambda_2 \cdot \frac{n^2}{\log n^2} + \lambda_1 \cdot \frac{2^n}{2^n} \cdot \ell \\
\leq D(n/2) + \lambda_2 \cdot \frac{n^2}{2^n} + \lambda_1 \cdot \frac{2^n}{2^n} + O(n^2/\log n) \\
= D(n/2) + O(n^2/\log n).
\]

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 \( S(n) \leq S(n/2) + 2^{n^3} - 2^n/2^n + O(n^2/\log n) \leq 2n^3 + O(n^2/\log n) \). \( \square \)

C. Construction of \( G_k \) and \( R \)

In this section, we will show how to construct operator \( G_k \), which consists of two stages: 1) Generate Stage and 2) Gray Path Stage, see Fig. 5. Along the way, we will also show the construction of \( R \).

1) **Generate Stage:** In this stage, we implement operator \( U^{(k)} \) such that

\[
|y^{(k-1)}\rangle \overset{U^{(k)}}{\longrightarrow} |y^{(k)}\rangle, \quad k \in [\ell] \tag{20}
\]

where \( y^{(k-1)} \) and \( y^{(k)} \) are defined in (14) and determined by \( T^{(k-1)} \) and \( T^{(k)} \), respectively. For \( k \in [\ell] \), recall that \( t^{(k)} = \{ t^{(k)}_1, \ldots, t^{(k)}_r \} \). Fix this ordering, view each \( t^{(k)}_i \) as a column vector, and define a matrix \( \tilde{T}^{(k)} = [ t^{(k)}_1, \ldots, t^{(k)}_r ]^{T} \in [0,1]^{r_\ell \times r_\ell} \) for \( k \in [\ell] \), with special case \( \tilde{T}^{(0)} \) defined to be \( I_r \). Then, the vectors \( y^{(k)} \) can be rewritten as follows:

\[
y^{(k)} = \tilde{T}^{(k)} \hat{y}_{\text{target}} \quad \forall k \in [r_\ell] \cup \{0\}. \tag{21}
\]

Since \( t^{(k)}_1, t^{(k)}_2, \ldots, t^{(k)}_r \) are linearly independent over \( \mathbb{F}_2 \), \( \tilde{T}^{(k)} \) is an invertible linear transformation over \( \mathbb{F}_2 \). Now define a unitary \( U^{(k)}_G \) by \( U^{(k)}_G |y\rangle = |\tilde{T}^{(k)} (\tilde{T}^{(k)} - 1)^{-1} y\rangle \), where the matrix-vector multiplication at the right-hand side is over \( \mathbb{F}_2 \). From (21), we see that

\[
U^{(k)}_G |y^{(k-1)}\rangle = |\tilde{T}^{(k)} (\tilde{T}^{(k)} - 1)^{-1} y^{(k-1)}\rangle = |\tilde{T}^{(k)} x_{\text{target}}\rangle = |y^{(k)}\rangle
\]

satisfying (20). Also note that when viewed as a linear transformation over \( \mathbb{F}_2 \), \( U^{(k)}_G \) is invertible. Thus, according to Lemma 15, the following depth upper bound applies.

**Lemma 19:** The Generate Stage unitary \( U^{(k)}_G \) can be realized by an \( O((r_\ell/\log r_\ell) \cdot \log n) \)-depth and \( O(r_\ell^2/\log r_\ell) \)-size CNOT circuit without ancillary qubits.

Similar to the discussion of \( U^{(k)}_G \) operator \( R \) can be defined by \( R |y\rangle = |(\tilde{T}^{(\ell)} - 1)^{-1} y\rangle \), then \( R |y^{(\ell)}\rangle = |(\tilde{T}^{(0)} - 1)^{-1} y^{(0)}\rangle = |y^{(0)}\rangle \). Thus, \( R \) can be also viewed an invertible linear transformation over \( \mathbb{F}_2 \). Applying Lemma 15 gives the bound in Lemma 16.

2) **Gray Path Stage:** This stage implements the operator

\[
|x_{\text{control}}\rangle \xrightarrow{U^{(k)}_{\text{GrayPath}}} e^{i \sum_{\alpha} f_k(s,x) a_\alpha} |x_{\text{control}}\rangle |y^{(k)}\rangle
\]

where \( k \in [\ell] \) and \( F_k \) is defined in (15). The Gray Path Stage in this section is similar to the Gray Path Stage in Section IV, though we need to use a Gray code cycle here instead of a Gray code path. For every \( i \in [r_\ell] \), let \( c^{(i)}_1, c^{(i)}_2, \ldots, c^{(i)}_{2^{r_\ell}}, c^{(i)}_{2^{r_\ell}} = 0^{r_\ell} \) denote the i-Gray code of \( r_\ell \) bits starting at \( c^{(i)}_1 = 0^{r_\ell} \) for \( i \in [r_\ell] \). Let \( h_i \) denote the index of the bit that \( c^{(i)}_j \) differ for each \( j \in \{2,3,\ldots,2^r\} \) and \( h_1 \) the index of the bit that \( c^{(i)}_1 \) and \( c^{(i)}_{2^{r_\ell}} \) differ. For the i-Gray code cycle of \( r_\ell \) bits

\[
h_i = \begin{cases} \left( r_i + i - 2 \mod r_i \right) + 1, & \text{if } j = 1 \\ \left( \zeta (j-1) + i - 2 \mod r_i \right) + 1, & \text{if } j \neq 1. \end{cases} \tag{23}
\]

The exact form of \( h_i \) is not crucial; the important fact to be used later is that the indices \( h_{1p}, h_{2p}, \ldots, h_{i_{\text{rep}}} \) are all different.
This stage consists of $2^c + 1$ phases.

1) In phase 1, circuit $C_1$ applies a rotation $R(\alpha_{q_c,r_1}^{c_1})$ on the $i$th qubit in the target register for all $i \in [r_1]$ if the string $0^{r_c}t_i^{(k)} \in F_k$, where $\alpha_{q_c,r_1}^{c_1}$ is defined in (6).

2) In phase $p \in \{2, \ldots, 2^n\}$, circuit $C_p$ consists of two steps.
   a) Step $p.1$ is a unitary that, for all $i \in [r_1]$, applies a CNOT gate on the $i$th qubit in target register, controlled by the $h_i$th qubit in control register.
   b) Step $p.2$ is a unitary that, for all $i \in [r_1]$, applies a rotation $R(\alpha_{q_c,r_1}^{c_1'})$ on the $i$th qubit in target register if $c_i^{(k)} \in F_k$, where $\alpha_{q_c,r_1}^{c_1'}$ is defined in (6).

3) In phase $2^c + 1$, circuit $C_{2^c+1}$ implements a unitary that, for all $i \in [r_1]$, applies a CNOT gate on the $i$th qubit in target register, controlled by the $h_i$th qubit in control register.

The next lemma gives the correctness and depth of this constructed circuit. The proof of Lemma 20 is shown in [36, Appendix J].

Lemma 20: The quantum circuit defined above is of depth $O(2^c)$ and size $O(r_c2^{c+1})$, and implements Gray Path Stage $U_{\text{GrayPath}}$ in (22).

According to Lemmas 19 and 20, operator $G_k$ can be implemented in depth $O(2^c)+O(r_c/\log r_1)=O(2^c)$, and, the size of the circuit is at most $O(n^2/\log n)+r_c2^{c+1}=O(r_c2^{c+1}).$ This completes the proof of Lemma 17.

VI. 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 IV when the number of ancillary qubits $m = \Omega(2^n/n^2)$. In Section VI-A, we will present the framework, and in Section VI-B, 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 $i \in \{0, 1, \ldots, 2^n-1\}$, and $|e_i|$ is the corresponding $2^n$-qubit state. We use $n$-qubit state $|i\rangle = (|0\rangle, |1\rangle)^{\otimes n}$ to denote the binary encoding of $i$, where $i_0, \ldots, i_{n-1} \in \{0, 1\}$ and $i = \sum_{j=0}^{n-1} i_j \cdot 2^j$.

A. New Framework for Quantum State Preparation

The quantum circuit in Section II for QSP consists of $n$ UCGs $V_1, V_2, \ldots, V_n$ [Fig. 2(a)]. In Section IV, we showed that any $j$-qubit UCG $V_j$ can be implemented by a quantum circuit of depth $O((j + (2/\log m))^2)$ 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 $\sum |v_i| |e_i|$ using the result in [30], and then make an encoding transform $|e_i| \rightarrow |i\rangle$, 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 VI-B. 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 IV for the suffix part.

Our new circuit framework for QSP in the parameter regime $m = \Omega(2^n/n^2)$ is shown in Fig. 6. Let $t = \lfloor \log(m/3) \rfloor$. In the previous framework, the first $t$ UCGs are a QSP circuit to prepare a $t$-qubit quantum state $|\psi^{(t)}_t\rangle = \sum_{k=0}^{2^t} v_k' |k\rangle$, where $v_k' = \sqrt{\sum_{j=0}^{2^{n-t-1}} |v_{2t-k+j}|^2}$. 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 $2t$-qubit quantum state $|\psi'_t\rangle = \sum_{k=0}^{2^t-1} v_k' |e_k\rangle$, where $e_k \in \{0, 1\}^{2t}$ and by the quantum circuit in [30].

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

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

B. 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 21: Given a vector $v = (v_0, v_1, \ldots, v_{2^n-1})^T \in \mathbb{C}^{2^n}$ with unit $\ell_2$-norm, any $2^n$-qubit quantum state $|\psi^{(t)}_t\rangle = \sum_{k=0}^{2^n-1} v_k |e_k\rangle$ can be prepared from the initial state $|0\rangle \otimes\mathbb{C}^{2^n}$ 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 22: The following unitary transformation on $2^n$ qubits:

$$|e_i\rangle \rightarrow |i\rangle |0^{2^n-n}\rangle \forall i \in \{0\} \cup \{2^n-1\}, e_i \in \{0, 1\}^{2^n}$$

(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 22 is shown in [36, Appendix K]. Now we are ready to give the hybrid algorithm and cost analysis.
Lemma 23: For any \( m \in \Omega(2^n/n^2) \), any \( n\)-qubit quantum state \( |\psi_v\rangle \) can be generated by a quantum circuit, using single-qubit gates and CNOT gates, of depth \( O(n(n - \log(m/3)+1) + (2^m/m)) \) and size \( O(2^n) \) with \( n \) ancillary qubits.

Proof: Let \( t = \lceil \log(m/3) \rceil \). Define a quantum state 
\[
|\psi_v^{(t)}\rangle = \sum_{i=0}^{2^t-1} v_i|i\rangle,
\]
where 
\[
v_i = \sqrt{\frac{2^{m-t}-1}{|v_{2^t-1-i}|^2}}.
\]
Note that \( |\psi_v^{(t)}\rangle = V_1 V_{2t-1} \cdots V_1 |0\rangle^{\otimes n} \), the state after we apply the first \( t \) UCGs in Fig. 2(a).

According to Lemma 21, given the unit vector \( v_i = (v_0, \ldots, v_{2^t-1}) \), we can prepare a \( 2^t \)-qubit quantum state 
\[
|\psi'_v\rangle = \sum_{i=0}^{2^t-1} v'_i|i\rangle
\]
by a quantum circuit of depth \( O(t) \) and size \( O(2^t) \). 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 workspace of \( 2^t + 2^t \leq m \) qubits.

To change \( |\psi_v^{(t)}\rangle \) to the final target state \( |\psi_v\rangle \), what is left is to apply \( V_{2t+1}, \ldots, V_n \) to \( |\psi'_v\rangle \). By Lemma 6, each \( V_j \) can be implemented by a circuit of depth \( O((2^m/m)) \) and size \( O(2^m) \) by \( n \) ancillary qubits. Hence, \( V_{2t+1}, \ldots, V_n \) can be realized by a quantum circuit of depth \( \sum_{j=t+1}^{n} O(j + (2^m/m)) = O((n - \lceil \log(m/3) \rceil) + 2^m/m) \), 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 QSP circuit are \( O(n(n - \log(m/3) + 1) + (2^m/m)) \) and \( O(2^n) \), respectively.

Note that when \( m = \Omega(2^n/n^2) \), 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 3: For a circuit preparing an \( n \)-qubit quantum state with \( m = \Omega(2^n/n^2) \) ancillary qubits, the minimum depth \( D_{\text{QSP}}(n, m) \) for different ranges of \( m \) are characterized as follows:

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

VII. EXTENSIONS AND IMPLICATIONS

A. 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 5 and Corollary 4 are shown in [36, Appendix L].

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

In [40], it was shown that one needs at least \( \Omega(4^n) \) 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 four free real parameters, and that each single-qubit gate right before the output has three 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 \( 4k + 3n \geq 4^n - 1 \). 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 4: The minimum circuit depth \( D_{\text{UNITARY}}(n, m) \) for an arbitrary \( n \)-qubit unitary with \( m \) ancillary qubits satisfies

\[
D_{\text{UNITARY}}(n, m) = \Theta\left(\frac{4^n}{m + n}\right), \quad \text{if } m = O(2^n/n),
\]

\[
D_{\text{UNITARY}}(n, m) \in [\Omega(n), (n2^n)], \quad \text{if } m = o(2^n/n).
\]

B. DECOMPOSITION WITH CLIFFORD + T GATE SET

The 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 1 (\( \epsilon \)-Approximation): For any \( \epsilon > 0 \), a unitary matrix \( U \) is \( \epsilon \)-approximated by another unitary matrix \( V \) if

\[
\|U - V\|_2 \leq \max_{\|\psi\|_2=1} \|\langle U - V|\psi\rangle\| < \epsilon.
\]

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 QSP (Corollary 5) and unitary synthesis (Corollary 6). The Corollary 5 is a restatement of Corollary 2. The proofs are shown in [36, Appendix M].

Corollary 5: For any \( n \)-qubit target state \( |\psi_v\rangle \) and \( \epsilon > 0 \), one can prepare a state \( |\psi'_v\rangle \) which is \( \epsilon \)-close to \( |\psi_v\rangle \) in \( \ell_2 \)-distance, by a quantum circuit consisting of \{CNOT, H, S, T\} gates of depth

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

where \( m \) is the number of ancillary qubits.

The following is an implementation of a unitary matrix.

Corollary 6: Any \( n \)-qubit general unitary matrix can be implemented by a circuit, using the \{CNOT, H, S, T\} gate set, of depth \( O(n2^n + ([4^n \log(4^n/\epsilon)]/[m + n])) \) with \( m \) ancillary qubits.

Remark 1: Our circuits for general states can be also extended to circuits for sparse states. See details in [36, Appendix N].

VIII. CONCLUSION

In this article, 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 \( O(n \log n + (2^n/[n + m])) \) 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 $O(n^2 + (4^n / (m + n)))$ 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 4. 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] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, “Quantum machine learning,” Nature, vol. 549, no. 7671, pp. 195–202, 2017.
[2] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, “Simulating hamiltonian dynamics with a truncated Taylor series,” Phys. Rev. Lett., vol. 114, no. 9, 2015, Art. no. 90502.
[3] G. H. Low and I. L. Chuang, “Optimal hamiltonian simulation by quantum signal processing,” Phys. Rev. Lett., vol. 118, no. 1, 2017, Art. no. 10501.
[4] G. H. Low and I. L. Chuang, “Hamiltonian simulation by qubitization,” Quantum, vol. 3, p. 163, Jul. 2019.
[5] D. W. Berry, A. M. Childs, and R. Kothari, “Hamiltonian simulation with nearly optimal dependence on all parameters,” in Proc. IEEE 56th Annu. Symp. Found. Comput. Sci., 2015, pp. 792–809.
[6] S. Lloyd, M. Mohseni, and P. Rebentrost, “Quantum principal component analysis,” Nat. Phys., vol. 10, no. 9, pp. 631–633, 2014.
[7] I. Kerenidis and A. Prakash, “Quantum recommendation systems,” in Proc. 8th Innov. Theor. Comput. Sci. Conf. (ITCS), 2017, pp. 1–22.
[8] P. Rebentrost, A. Steffens, I. Marvian, and S. Lloyd, “Quantum singular-value decomposition of nonsparse low-rank matrices,” Phys. Rev. A, vol. 97, no. 1, 2018, Art. no. 12327.
[9] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” Phys. Rev. Lett., vol. 103, no. 15, 2009, Art. no. 150502.
[10] L. Wossnig, Z. Hao, and A. Prakash, “Quantum linear system algorithm for dense matrices,” Phys. Rev. Lett., vol. 120, no. 5, 2018, Art. no. 050502.
[11] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, “Q-means: A quantum algorithm for unsupervised machine learning,” 2018, arXiv:1812.03584.
[12] I. Kerenidis and J. Landman, “Quantum spectral clustering,” 2020, arXiv:2007.00280.
[13] P. Rebentrost, M. Mohseni, and S. Lloyd, “Quantum support vector machine for big data classification,” Phys. Rev. Lett., vol. 113, no. 13, 2014, Art. no. 130503.
[14] R. Barends et al., “Superconducting quantum circuits at the surface code threshold for fault tolerance,” Nature, vol. 508, no. 7497, pp. 500–503, 2014.
[15] J. Kelly et al., “State preservation by repetitive error detection in a superconducting quantum circuit,” Nature, vol. 519, no. 7541, pp. 66–69, 2015.
[16] C. Song et al., “10-Qubit entanglement and parallel logic operations with a superconducting circuit,” Phys. Rev. Lett., vol. 119, no. 18, 2017, Art. no. 180511.
[17] F. Arute et al., “Quantum supremacy using a programmable superconducting processor,” Nature, vol. 574, no. 7779, pp. 505–510, 2019.
[18] “IBM Quantum,” 2021. [Online]. Available: https://quantum-computing.ibm.com/.
[19] J. Chow, O. Dial, and J. Gambetta, “IBM quantum breaks the 100-Qubit processor barrier.” IBM Research Blog, 2021. [Online]. Available: https://research.ibm.com/blog/127-qubit-quantum-processor-eagle/pageStart.
[20] J. Gambetta, “IBM’s roadmap for scaling quantum technology,” 2020. [Online]. Available: https://www.ibm.com/blogs/research/2020/09/ibm-quantum-roadmap/
[21] L. Grover, “Unveiling our new quantum AI campus,” 2021. [Online]. Available: https://blog.google/technology/ai/unveiling-our-new-quantum-ai-campus/.
[22] Z. Zhang, Q. Wang, and M. Ying, “Parallel quantum algorithm for hamiltonian simulation,” 2021, arXiv:2105.11889.
[23] S. Johri et al., “Nearest centroid classification on a trapped ion quantum computer,” NPJ Quantum Inf., vol. 7, no. 1, pp. 1–11, 2021.
[24] I. Kerenidis and A. Luongo, “Classification of the MNIST data set with quantum slow feature analysis,” Phys. Rev. A, vol. 101, no. 6, 2020, Art. no. 62327.
[25] D. Dervovic, M. Herbster, P. Mountney, S. Severini, N. Usher, and L. Wossnig, “Quantum linear systems algorithms: A primer,” 2018, arXiv:1802.08227.
[26] Z. Zhao, J. K. Fitzsimons, P. Rebentrost, V. Danjko, and J. F. Fitzsimons, “Smooth input preparation for quantum and quantum-inspired machine learning,” Quantum Mach. Intell., vol. 3, no. 1, pp. 1–6, 2021.
[27] R. LaRose and B. Coyle, “Robust data encodings for quantum classifiers,” Phys. Rev. A, vol. 102, no. 3, 2020, Art. no. 32420.
[28] D. Aharonov and Y. Touati, “Quantum circuit depth lower bounds for homological codes,” 2018, arXiv:1810.03912.
[29] X. Sun, G. Tian, S. Yang, P. Yuan, and Z. Zhang, “Asymptotically optimal circuit depth for quantum state preparation and general unitary synthesis,” 2021, arXiv:2108.06150v3.
[30] A. Barenco et al., “Elementary gates for quantum computation,” Phys. Rev. A, vol. 52, no. 5, pp. 3457–3467, 1995.
[31] E. Knill, “Approximation by quantum circuits,” 1995, arXiv:quant-ph/9508006.
[32] J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, “Efficient decomposition of quantum gates,” Phys. Rev. Lett., vol. 92, no. 17, 2004, Art. no. 177902.
[33] V. Shende, I. L. Markov, and S. S. Bullock, “Minimal universal two-qubit controlled-NOT-based circuits,” Phys. Rev. A, vol. 69, no. 6, 2004, Art. no. 62321.
[34] S. S. Bullock and I. L. Markov, “Asymptotically optimal circuits for arbitrary n-qubit diagonal computations,” Quantum Inf. Comput., vol. 4, no. 1, pp. 27–47, 2004.
[35] I. F. Araujo, D. K. Park, F. Petruccione, and A. J. da Silva, “A divide-and-conquer algorithm for quantum state preparation,” 2020, arXiv:2008.01511.
[36] G. H. Low, V. Kliuchnikov, and L. Schaeffer, “Trading T-gates for dirty qubits in state preparation and unitary synthesis,” 2018, arXiv:1812.00954.
[37] R. Babbush et al., “Encoding electronic spectra in quantum circuits with linear T complexity,” Phys. Rev. X, vol. 8, no. 4, 2018, Art. no. 41015.
[38] G. Rosenthal, “Query and depth upper bounds for quantum circuits,” 2021, arXiv:2111.07992.
[39] X.-M. Zhang, Z. Li, and X. Yuan, “Quantum state preparation with optimal circuit depth: Implementations and applications,” 2022, arXiv:2201.11495.
[47] P. Yuan and S. Zhang, “Optimal QRAM and improved unitary synthesis by quantum circuits with any number of ancillary qubits,” 2022, arXiv:2202.11302.

[48] J. Alcock, P. Yuan, and S. Zhang, “Does qubit connectivity impact quantum circuit complexity?” 2022, arXiv:2211.05413.

[49] G. Frank, “Pulse code communication,” U.S. Patent 2 632 058, Jul. 1953.

[50] C. Savage, “A survey of combinatorial gray codes,” SIAM Rev., vol. 39, no. 4, pp. 605–629, 1997.

[51] E. N. Gilbert, “Gray codes and paths on the N-cube,” Bell Syst. Tech. J., vol. 37, no. 3, pp. 815–826, May 1958.

[52] J. Welch, A. Bocharov, and K. M. Svore, “Efficient approximation of diagonal unitaries over the Clifford+T basis,” 2014, arXiv:1412.5608.

[53] J. Welch, D. Greenbaum, S. Mostame, and A. Aspuru-Guzik, “Efficient quantum circuits for diagonal unitaries without ancillas,” New J. Phys., vol. 16, no. 3, 2014, Art. no. 33040.

[54] J. Jiang, X. Sun, S.-H. Teng, B. Wu, K. Wu, and J. Zhang, “Optimal space-depth trade-off of CNOT circuits in quantum logic synthesis,” in Proc. 14th Annu. ACM-SIAM Symp. Discr. Algorithms, 2020, pp. 213–229.

[55] K. N. Patel, I. L. Markov, and J. P. Hayes, “Optimal synthesis of linear reversible circuits,” Quantum Inf. Comput., vol. 8, no. 3, pp. 282–294, 2008.

Xiaoming Sun (Senior Member, IEEE) received the Ph.D. degree from Tsinghua University, Beijing, China, in 2005. He is a Professor with the Institute of Computing Technology (ICT), Chinese Academy of Sciences, Beijing, and the Director of the Laboratory for Quantum Computation and Theoretical Computer Science, ICT. He is also a Professor with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing. His research interests include quantum computing, computational complexity, approximation algorithms, and combinatorics.

Prof. Sun is currently the Director of CCF Technical Committee on Theoretical Computer Science.

Guojing Tian received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2017. She is an Associate Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, and also with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing. She worked with the Center for Quantum Technologies, National University of Singapore, Singapore, as a joint Ph.D. from 2014 to 2015. Her research interests include quantum computation, quantum algorithm, quantum circuit, quantum nonlocality, and quantum simulation.

Shuai Yang received the bachelor’s degree in computer science from Sichuan University, Chengdu, China, in 2018. He is currently pursuing the Ph.D. degree with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, and the University of Chinese Academy of Sciences, Beijing, advised by Prof. J. Zhang. He is interested in problems at the quantum computation. He currently focuses on the quantum circuit synthesis and quantum circuit optimization.

Pei Yuan received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2015, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2021. She is currently with Tencent Quantum Laboratory, Shenzhen, China. Her research interests include quantum circuit optimization and quantum algorithms.

Shengyu Zhang received the Ph.D. degree in computer science from Princeton University, Princeton, NJ, USA, in 2006. He is a Distinguished Scientist with Tencent, Shenzhen, China, and the Director of Tencent Quantum Laboratory, Shenzhen. His research interest lies in quantum computing theory and systems, algorithm designing, computational complexity, foundation of artificial intelligence, and AI for sciences.