A quantum algorithm for solving eigenproblem of the Laplacian matrix of a fully connected graph

Hai-Ling Liu 1,2, Su-Juan Qin 1,† Lin-Chun Wan 1, Chao-Hua Yu 3, Shi-Jie Pan 1, Fei Gao 1,‡ and Qiao-Yan Wen 1
1 State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
2 State Key Laboratory of Cryptology, P.O. Box 5159, Beijing, 100878, China and
3 School of Information Management, Jiangzi University of Finance and Economics, Nanchang 330032, China

(Dated: March 29, 2022)

Solving eigenproblem of the Laplacian matrix of a fully connected graph has wide applications in data science, machine learning, and image processing, etc. However, it is very challenging because it involves expensive matrix operations. Here, we propose an efficient quantum algorithm to solve it based on a general assumption that the element of each vertex and its norms can be accessed via a quantum random access data structure. Specifically, we propose in detail how to construct the block-encodings of operators containing the information of the weight matrix and the degree matrix respectively, and further obtain the block-encoding of the Laplacian matrix. This enables us to implement the quantum simulation of the Laplacian matrix by employing the optimal Hamiltonian simulation technique based on the block-encoding framework. Afterwards, the eigenvalues and eigenvectors of the Laplacian matrix are extracted by the quantum phase estimation algorithm. Compared with its classical counterpart, our algorithm has a polynomial speedup on the number of vertices and an exponential speedup on the dimension of each vertex over its classical counterpart. We also show that our algorithm can be extended to solve the eigenproblem of symmetric (non-symmetric) normalized Laplacian matrix.

I. INTRODUCTION

In the era of big data, graph learning [1] has attracted considerable attention owing to its wide applications in data science, machine learning, and image processing, etc. In the process of dealing with problems related to graph learning, such as graph networks [2, 3], image processing [4, 5], and reinforcement learning [6], it is often necessary to solve the eigenproblem of the Laplacian matrix of a fully connected graph to avoid dropping crucial nonlocal information. However, it is very challenging because it involves expensive matrix operations. Therefore, it is imperative for us to design an efficient algorithm to solve this problem.

Recently, quantum computing has exhibited potential acceleration advantages over classical computing by exploiting the unique properties of supposition and entanglement in quantum mechanics in solving certain problems, such as factoring integers [7], unstructured database searching [8], solving equations [9, 11], regression [12, 14], dimensionality reduction [15, 17], anomaly detection [18, 19] and neural network [20]. Therefore, some scholars have proposed employing quantum algorithm to solve eigenproblem of the Laplacian matrix.

In 2016, Huang et al. proposed a quantum Laplacian eigenmap algorithm, which has a significant speedup compared with its classical counterparts [21]. However, this algorithm relies on a assumption that the classically stored elements of the weight matrix and the degree matrix can be superposition accessed by a quantum random access memory [22]. In fact, in most scenarios [2, 5, 23, 24], usually given is the vertex set of the graph, and the elements of the weight matrix and the degree matrix require complex computations to get. In 2021, Kerenidis et al. [25] proposed a quantum algorithm for solving the eigenproblem of the symmetric normalized Laplacian matrix, and successfully applied it to the spectral clustering algorithm. However, their algorithm cannot be directly used to solve the eigenproblem of the Laplacian matrix of a fully connected graph.

In this paper, we design an efficient quantum algorithm to solve eigenproblem of the Laplacian matrix of a fully connected graph, which starts from accessing the vertex set of the graph. Specifically, we assume that the element of each vertex and its norms can be accessed via a quantum random access data structure [26]. This makes our proposed algorithm more suitable for practical scenarios.

We adopt the optimal Hamiltonian simulation technique based on the block-encoding framework [27, 29] to implement the quantum simulation of the Laplacian matrix and then employ the quantum phase estimation algorithm [30] to extract the eigenvalues and eigenvectors of the Laplacian matrix. The core of our entire algorithm is to construct the block-encoding of the Laplacian matrix. To achieve it, we design the specific controlled unitary operators to prepare the quantum states to construct the block-encodings of operators containing the information of the weight matrix and the degree matrix respectively, and further obtain the block-encoding of the Laplacian matrix. Compared with its classical counterpart, our algorithm has a polynomial speedup on the number of vertices and an exponential speedup on the dimension of each vertex over its classical counterpart. In particular, our algorithm can also be used to solve eigen-
problem of the weighted matrix, which is also of great significance [31, 34]. We also show that our algorithm can be extended to solve the eigenproblem of symmetric (non-symmetric) normalized Laplacian matrix.

The remainder of the paper is organized as follows. In Sec. II, we give a brief overview of the graph Laplacian matrix. In Sec. III, we propose a quantum algorithm to solve eigenproblem of the Laplacian matrix of a fully connected graph. In Sec. IV, we generalized the algorithm to solve eigenproblem of symmetric (non-symmetric) normalized Laplacian matrix. In Sec. V, we give some discussions. Finally, we present our conclusion in Sec. VI.

II. REVIEW OF THE GRAPH LAPLACIAN MATRIX

Given a weighted undirected graph $G = (V, E)$ with the vertex set $V = \{x_i | x_i \in R^n\}_{i=1}^n$ and the edge set $E$. The weighted matrix $W \in R^{n \times n}$ of the graph $G$ is defined as follows:

$$w_{ij} = w_{ji} = \begin{cases} \geq 0, & i \neq j, \\ 0, & i = j. \end{cases}$$

(1)

Specifically, the element $w_{ij} > 0$ represents that vertex $x_i$ is connected to vertex $x_j$, otherwise $w_{ij} = 0$. For $w_{ij}$, we take the Gaussian similarity function $w_{ij} = \exp(-\lambda \|x_i - x_j\|^2)$ as an example which has a wide range of applications, where $\lambda > 0$ is any given real number [24, 35, 36]. It is worth noting that we can also choose other forms of nonlinear similar functions.

Next, the degree matrix $D$ of the graph $G$ is defined as a diagonal matrix $D = diag(d_{ii}) \in R^{n \times n}$, where

$$d_{ii} = \sum_{j=1}^{n} w_{ij}.$$  

Given $W$ and $D$, the graph Laplacian matrix $L$ is defined as

$$L = D - W \in R^{n \times n},$$  

(2)

where $L$ is a symmetric positive semi-definite matrix. In addition, there are two common normalized Laplacian matrices as follows:

$$L_s = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}W D^{-\frac{1}{2}},$$

$$L_r = D^{-1}L = I - D^{-1}W,$$  

(3)

where $L_s$ ($L_r$) is a symmetric (non-symmetric) matrix. For more information about the Laplacian matrix $L$, see Ref. [23, 24].

Solving eigenproblem of the Laplacian matrix $L$ of a fully connected graph has wide applications in graph networks [2, 3], image processing [4, 5], and reinforcement learning [6], etc. For a fully connected graph, $L$ is a dense matrix, and the multiplicity of the zero eigenvalue of $L$ is equal to 1 and the eigenvector corresponding to zero eigenvalue is $1 = (1, \cdots , 1)^T \in R^n$ [24]. It provides no extra information. Therefore, we consider extracting $1 \leq d \leq (n - 1)$ non-zero eigenvalues and the corresponding eigenvectors of $L$, and its classical complexity is $O(mn^2 + dn^3)$ [37]. This is quite time consuming when the size of the vertex set is large. Therefore, it is imperative for us to design an efficient algorithm to solve it.

III. A QUANTUM ALGORITHM FOR SOLVING EIGENPROBLEM OF THE LAPLACIAN MATRIX OF A FULLY CONNECTED GRAPH

To design an efficient quantum algorithm to solve eigenproblem of the Laplacian matrix $L$, we construct the block-encodings of operators containing the information of $W$ and $D$ respectively, and further obtain the block-encoding of $L$. Then, we adopt the optimal Hamiltonian simulation technology based on the block-encoding framework [27, 29] to realize the quantum simulation of $L$. Finally, we perform quantum phase estimation algorithm [30] to extract the eigenvalues and eigenvectors of $L$.

The entire section consists of five subroutines: we review the optimal Hamiltonian simulation technology based on the block-encoding framework in Sec. III A, quantum algorithms for preparing the quantum states to construct the block-encodings of operators containing the information of $W$ and $D$ in Sec. III B and Sec. III C respectively, a quantum algorithm for obtaining the block-encoding of $L$ to implement the quantum simulation of $L$ in Sec. III D, and a quantum algorithm to extract the eigeninformation of $L$ in Sec. III E. For convenience, we define the base of the logarithm function as 2, which can be abbreviated as log $x$.

Assume that the vertex set $V = \{x_i | x_i \in R^n\}_{i=1}^n$ is stored in a quantum random access memory data structure [29], i.e., the element $x_{ij}$ of each vertex $x_i$ is stored in the $i$th leaf of the binary tree, and the internal node of the tree stores the modulo sum of the elements in the subtree rooted in it. Then there exists a quantum algorithm that can perform the following map with $\varepsilon_x$-precision in $O(\log(mn/\varepsilon_x))$ time:

$$U|i\rangle|0\rangle \rightarrow |i\rangle|x_i\rangle = |i\rangle \frac{1}{\|x_i\|} \sum_{j=1}^{m} x_{ij}|j\rangle.$$  

(4)

In addition, this structure can perform the unitary operator $O$ in time $O(\log(mn))$:

$$O|i\rangle|0\rangle \rightarrow |i\rangle||x_i\|.$$  

(5)

This data structure has also been successfully applied to quantum data compression [38], quantum linear systems with displacement structures [39] and so on.

A. Review of the optimal Hamiltonian simulation technology based on the block-encoding framework

In this section, we review the optimal Hamiltonian simulation technology based on the block-encoding frame-
work \cite{27,29}. We first give the framework of block-encoding.

**Definition1** (Block-encoding). Assume that $A$ is an $s$-qubits operator, $\alpha, \epsilon_A \in \mathbb{R}^+$, and $a \in \mathbb{N}$, then we say that the $(s+a)$-qubits unitary $U$ is an $(\alpha, a, \epsilon_A)$ block-encoding of $A$ if it satisfies

$$\|A - \alpha(\langle 0 \rangle^{\otimes a} \otimes I)U(\langle 0 \rangle^{\otimes a} \otimes I)\| \leq \epsilon_A. \quad (6)$$

Meanwhile, Low and Chuang \cite{27} also proposed a block-encoding framework of a purified density operator, as follows:

**Lemma1** (Block-encoding of density operators). Suppose that $\rho$ is an $s$-qubits density operator and $G$ is an $(a+s)$-qubits unitary such that for all $\|\rho\|_1 \leq \beta$. The pair of unitaries $(P_L, P_R)$ is called a $(\beta, b, \epsilon_y)$-state-preparation-pair if $P_L|0\rangle^{\otimes b} = \sum_{j=1}^{2^b} c_j |j\rangle$ and $P_R|0\rangle^{\otimes b} = \sum_{j=1}^{2^b} d_j |j\rangle$ such that $\sum_{j=1}^{2^b} |\beta| (c_j^* d_j) - |y_j| \leq \epsilon_y$ and for all $j \in m + 1, \ldots, 2^b$, we have $c_j^* d_j = 0$.

**Lemma2** (Linear combination of block-encoding matrices). Let $A = \sum_{j=1}^{m} y_j A_j$ be an $s$-qubits operator and $\epsilon_A \in \mathbb{R}^+$. Assume that $(P_L, P_R)$ is a $(\beta, b, \epsilon_y)$-state-preparation-pair for all $y \in \mathbb{C}^m$, $W = \sum_{j=1}^{m} |j\rangle\langle j| \otimes U_j + ((I - \sum_{j=1}^{m} |j\rangle\langle j|) \otimes I_a \otimes I_s)$ is an $(s+a+b)$-qubits unitary such that for all $j = 1, \ldots, m$, we have that $U_j$ is an $(\alpha, a, \epsilon_A)$-block-encoding of $A_j$. Then we can implement a $(\alpha \beta, a, a \epsilon_y + \alpha \beta \epsilon_A)$-block-encoding of $A$, with a single use of $W, P_R, \text{and } P^L_R$.

Based on the above block-encoding framework, the optimal Hamiltonian simulation technique is proposed as follows:

**Theorem1** (Optimal Block-Hamiltonian simulation). Suppose that $U$ is an $(a, a, \epsilon/|2t|)$-block-encoding of the Hamiltonian $H$. Then we can implement an $\epsilon$-precise Hamiltonian simulation unitary $V$ which is an $(1, s+1, \epsilon)$-block-encoding of $exp(\alpha H t)$, with $O(|\alpha| t + \log(1/\epsilon) / \log \log(1/\epsilon))$ uses of controlled-$U$ or its inverse and with $O(a|\alpha t | + a \log(1/\epsilon) / \log \log(1/\epsilon))$ two-qubit gates.

The core of this technique is to construct the block-encoding of an operator. To realize the quantum simulation of $L = D - W$, next we will design quantum algorithms to construct the block-encodings of operators containing the information of $W$ and $D$, respectively.

**B. Prepare the quantum state to construct the block-encoding of an operator containing the information of $W$**

We know that the elements of $W$ are

$$w_{ij} = \exp(-\lambda |x_i - x_j|^2) = \exp(-\lambda (||x_i||^2 + ||x_j||^2)) \exp(2\lambda x_i \cdot x_j), \quad (7)$$

where $i, j = 1, 2, \ldots, n$.

Due to $\exp(-\lambda (||x||^2 + ||x_j||^2))$ is a scalar that depends on the size of $||x||, i = 1, \ldots, n$, we take the Taylor expansion of $\exp(2\lambda x_i \cdot x_j)$ to get

$$\exp(2\lambda x_i \cdot x_j) = \sum_{k=0}^{\infty} \frac{(2\lambda)^k}{k!} (x_i \cdot x_j)^k. \quad (8)$$

According to Taylor’s theorem \cite{20}, by keeping only the low-order terms of the Taylor expansion, we can get a finite-dimensional approximation of $\exp(2\lambda x_i \cdot x_j)$, i.e.,

$$\exp(2\lambda x_i \cdot x_j) = \sum_{k=0}^{p} \frac{(2\lambda)^k}{k!} (x_i \cdot x_j)^k. \quad (9)$$

A detailed analysis of the errors of low order approximation with Taylor expansion, see Ref. \cite{41}.

Combine Eq.(7) with Eq.(9), we can obtain

$$w_{ij} = \exp(-\lambda (||x_i||^2 + ||x_j||^2)) \sum_{k=0}^{p} a_k (x_i \cdot x_j)^k, \quad (10)$$

where $a_k = (2\lambda)^k / k!$, $k = 0, 1, \ldots, p$.

Note that when $||x|| = 1$, we have

$$\exp(-\lambda (||x||^2 + ||x_j||^2)) = \exp(-2\lambda). \quad (11)$$

Once $\lambda$ is given, $\exp(-2\lambda)$ is a constant that can be absorbed into $a_k$, namely $\tilde{a}_k = [\exp(-2\lambda)(2\lambda)^k] / k!$. However, when $||x|| \neq 1$, its value affects the value of $a_k$, which result in $\exp(-\lambda (||x||^2 + ||x_j||^2))$ not being absorbed into $a_k$.

Therefore, next we will design the corresponding quantum algorithms to construct the block-encodings of operators containing the information of $W$ in two cases, as follows:

**(B.I.) A algorithm in the case of $||x|| = 1$**

The specific steps of the quantum algorithm are as follows.

(I.1) Prepare the quantum state

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle \frac{1}{\sqrt{p}} \sum_{k=0}^{p} \sqrt{a_k} |k\rangle_3 \otimes \hat{a} \equiv \sum_{k=0}^{p} \hat{a}_k. \quad (12)$$

Here, we assume that $\tilde{a} = \frac{1}{\sqrt{a}} (\sqrt{a_0}, \ldots, \sqrt{a_p}) \in R^{p+1}$ is stored in a quantum random access memory data structure \cite{28}. Then there is a quantum algorithm that can generate an $\epsilon_{\tilde{a}}$-approximation of $|\tilde{a}\rangle = \frac{1}{\sqrt{a}} \sum_{k=0}^{p} \sqrt{a_k} |k\rangle$ with gate complexity $O\{\text{poly} \log (p+1)/\epsilon_{\tilde{a}}\}$.\footnote{In this case, one can use the quantum algorithm to convert to a state $|\tilde{a}\rangle$ which can be retrieved with a register of $p+1$ qubits.}
(I.2) Apply the controlled unitary operator $R_u := \sum_{k=0}^p |k\rangle_2 \langle k|_2 \otimes (p-k)! k_{1,3}$ to three registers, the system state will becomes

\[
\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle_1 \sum_{k=0}^{p} \sqrt{a_k} |0\rangle_3^\otimes |x_i\rangle_3^\otimes k := \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle_1 |\Phi(x_i)\rangle_{2,3} := |\Phi\rangle,
\]

where

\[
|\Phi(x_i)\rangle_{2,3} = \frac{1}{\sqrt{a}} \left( \sqrt{a_0} |0\rangle_3 \otimes \sqrt{a_1} |1\rangle_3 \otimes \cdots \right) |x_i\rangle_3 \otimes p,
\]

The quantum circuit of implementing $R_u$ is shown in Fig. 1. Note that

\[
\langle \Phi(x_i) | \Phi(x_j) \rangle = [\hat{a}_0 + \hat{a}_1 (x_i \otimes x_j) + \cdots + \hat{a}_p (x_i \otimes x_j)^p] / \hat{a} = u_{ij}/\hat{a}.
\]

(I.3) Take partial trace for the second and third registers, we have

\[
\rho_0 := \text{Tr}(|\Phi\rangle \langle \Phi|)_{2,3} = \frac{1}{n} \sum_{i=1}^{n} \langle \Phi(x_i) | \Phi(x_j) \rangle |i\rangle_1 \langle j|_1
\]

\[
= (W + \hat{a} I) / n\hat{a}.
\]

According to Lemma 1, we can design an $p + \log(n(p+1))$-qubits unitary operator $G_0$ which implement $G_0 |0\rangle_0 \otimes |p+\log(p+1)\rangle_0 \otimes \log n \rightarrow |\Phi\rangle$, s.t., $\text{Tr}_{2,3}(|\Phi\rangle \langle \Phi|) = \rho_0$. Then $V_0 := (G_0^\dagger \otimes I_1)(I_2 \otimes \text{SWAP}_1)(G_0 \otimes I_1)$ is an $(1, p + \log(p+1), 2\epsilon_0)$-block-encoding of $\rho_0$, where $\epsilon_0$ is the error that produces $|\Phi\rangle$. See (B.II.2) for a detailed analysis about $\epsilon_0$. For convenience, we use $\epsilon$ to represent the estimated value caused by the quantum algorithm in the following sections.

(B.II.2) The Complexity of algorithm in (I)

In step (I.1), the complexity is $O(\text{poly} \log[n(p+1)/\epsilon_0])$, which is derived from $\log n$ Hadamard gates generating the first register and $O(\text{poly} \log[(p+1)/\epsilon_0])$ with gate complexity generating the second register.

In step (I.2), according to the quantum circuit of $R_u$ in Fig. 2, it needs $O(p)$ calls of the unitary operator $U$. Thus the complexity of $R_u$ is $O(p \text{poly} \log(mn/\epsilon x))$. Next we analyze the error that produces $|\Phi\rangle$ as follows:

\[
|||\hat{\Phi}| - |\Phi||_2
\]

\[
= ||\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle |\Phi(x_i)\rangle - \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle |\Phi(x_i)\rangle||_2
\]

\[
\leq \frac{1}{\sqrt{n}} \sum_{i=1}^{n} ||\hat{\Phi}(x_i) - |\Phi(x_i)||_2
\]

\[
\leq \frac{1}{\sqrt{n}} \sum_{i=1}^{n} [\epsilon_\alpha + (1 + 2 + \cdots + p)x_\epsilon]
\]

\[
= \sqrt{n}[\epsilon_\alpha + (1 + 2 + \cdots + p)x_\epsilon],
\]

where the final inequality is proved in Appendix A.

To make the error to generate $|\Phi\rangle$ is $\epsilon_0$, we choose $\epsilon_\alpha = \epsilon x$; then

\[
|||\hat{\Phi}| - |\Phi||_2 \leq \sqrt{np^2} \epsilon x := \epsilon_0.
\]

We can get $1/\epsilon x = \sqrt{np^2}/\epsilon_0$.

In short, the complexity is $O(\text{poly} \log(mn^3/2p^3/\epsilon_0))$.

(B.II.1) A algorithm in the case of $||x_\epsilon|| \neq 1$

To facilitate the introduction of the algorithm, we first list the lemma required.

Lemma 3 [32] (Quantum Multiply-Adder (QMA)) Let the binary representations of integers $a$ and $b$ be shown as $a_1a_2 \cdots a_n$ and $b_1b_2 \cdots b_n$, respectively. Then there is a quantum algorithm that can realize

\[
|a\rangle|b\rangle \rightarrow |a\rangle|ab\rangle,
\]

\[
|a\rangle|b\rangle \rightarrow |a\rangle|a + b\rangle,
\]

where $|a\rangle = |a_1\rangle \otimes \cdots \otimes |a_n\rangle$ and $|b\rangle = |b_1\rangle \otimes \cdots \otimes |b_n\rangle$.

The time complexity is given by $O(\text{poly} \log(1/\epsilon_m))$ with accuracy $\epsilon_m$.

The specific algorithm proceeds as following steps:

(II.1) Prepare the quantum state

\[
\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle_1 \sum_{k=0}^{p} \sqrt{a_k} |k\rangle_2 \langle 0|_2 \langle 0|_2 \otimes |0\rangle_2 \otimes |0\rangle_2,
\]

where $a = \sum_{k=0}^{p} a_k$ and the quantum state $\sum_{k=0}^{p} \sqrt{a_k} |k\rangle$ can be prepared with precision $\epsilon_a$ in the gate complexity of $O(\text{poly} \log[(p+1)/\epsilon_a])$ which is similar to the step of (I.1).
The quantum circuit of implementing \( R_O \) on the fourth register to generate \( R \) and uncompute redundant registers, we obtain

\[
\rho_1 := \text{Tr}(|\Psi\rangle\langle\Psi|)_{2,5} = \frac{1}{T} \sum_{i=1}^{n} |\langle\Psi(x_i)|\Psi(x_j)\rangle| i \langle j | = (W + I) \text{Tr}(W + I).
\]

According to Lemma 1, the process of generating the quantum state \( |\Psi\rangle \) can be regard as an \((2p + 3 + 3 \log[n(p + 1)]\) qubits unitary operator \( G_1 \) which implement \( G_1 |0\rangle^{\otimes 2p+3+3 \log(n(p+1))} |0\rangle^{\otimes \log n} \rightarrow |\Psi\rangle\), s.t.,

\[
\text{Tr}_{2,5}(|\Psi\rangle\langle\Psi|) = \rho_1. \quad \text{Then } V_1 := (G_1 \otimes I_{1,3,4,0})(I_{2,5} \otimes \text{SWAP}_{1,3,4,0})(G_1 \otimes I_{1,3,4}) \text{ is a } (1, 2p + 3 + 3 \log(n(p + 1)), 2\epsilon_1)\text{-block-encoding of } \rho_1, \text{ where } \epsilon_1 \text{ is the error that produces } |\Phi\rangle. \text{ See (BII.2) for detailed analysis for } \epsilon_1. \]
FIG. 3. The quantum circuit that generates the quantum state $|\Psi\rangle$. Here the subscripts $0, 1, \cdots, 5$ represent the index of the registers. $R_O$ and $R_U$ denote the controlled unitary operator $R_O$ and $R_U$, respectively. QMA denotes the quantum multiply-adder in Lemma 1. $R$ denotes a controlled rotation operator, and QAA represents the quantum amplitude amplification algorithm. $\exp(-\lambda x)$ stands for the $\exp(-\lambda x)$ gate. The numbers (II.2) – (II.9) represent the sequential steps of algorithm.

Next we analyze the complexity of step (II.7). According to Eq. (25), we can get the probability amplitude of the auxiliary qubit $|0\rangle$ is

$$\sum_{k=0}^{n} \sum_{i=1}^{p} a_k |\exp(-\lambda \|x_i\|^2)|^2 \|x_i\|^2 = O\left(\frac{1}{nC^2}\right).$$

The equation above is established by the Taylor expansion of order $p$ of $|\exp(\lambda \|x_i\|^2)|^2$, i.e.,

$$\|\exp(\lambda \|x_i\|^2)\|^2 = \exp(2\lambda \|x_i\|^2) = \sum_{k=0}^{p} a_k \|x_i\|^{2k}. \quad (32)$$

Thus, we need perform $O(\sqrt{nC})$ repetitions of quantum amplitude amplification algorithm to obtain the quantum state of Eq. (26).

Besides, we can obtain

$$\sum_{k=0}^{n} \sum_{i=1}^{p} a_k |\exp(-\lambda \|x_i\|^2)|^2 \|x_i\|^{2k}/n = O(1). \quad (33)$$

That means that $Y = O(n)$.

Finally, we analyze the error that produces $|\Psi\rangle$ as follows:

$$||\Psi\rangle - |\Psi\rangle||_2 = \left|\frac{1}{\sqrt{T}} \sum_{i=1}^{n} |i\rangle \langle \hat{\Psi}(x_i)| - \frac{1}{\sqrt{T}} \sum_{i=1}^{n} |i\rangle \langle \Psi(x_i)|\right||_2$$

$$\leq \frac{1}{\sqrt{T}} \sum_{i=1}^{n} \left|\langle \hat{\Psi}(x_i)| - |\Psi(x_i)|\right||_2 \quad (34)$$

$$\leq \frac{1}{\sqrt{T}} \sum_{i=1}^{n} \sqrt{\|x_i\|^2 \varepsilon_a + (1+2+\cdots+p)\varepsilon_x},$$

where $\|x_i\| > 1$ and a detailed analysis of the last inequality is given in Appendix A. In particular, for $0 < \|x_i\| < 1$, we also give a detailed analysis in Appendix A.

To ensure that the error of producing $|\Psi\rangle$ is $\varepsilon_f$, let $\varepsilon_a = \varepsilon_f$, we get

$$||\Psi\rangle - |\Psi\rangle||_2 \leq \sqrt{\max_i \|x_i\|^2} \varepsilon_f := \varepsilon_f. \quad (35)$$

Thus, we can obtain $1/\varepsilon_f = \sqrt{\max_i \|x_i\|^2} \varepsilon_f / \varepsilon_f$. Putting this all together, the complexity is

$$O\left(\sqrt{nC}poly \log(\sqrt{\max_i \|x_i\|^2} \max_i \|x_i\|^2) / \varepsilon_f\right).$$

\section{C. Prepare the quantum state to construct the block-encoding of an operator containing the information of $D$}

The elements of $D$ are $d_{ij} = \sum_{i=1}^{n} w_{ij}, i = 1, \cdots, n$, that is, the sum of each row of $W$. Thus, $d_{ij}$ can be regarded as the inner product of the row vector of $W$ and the vector $v = (1, \cdots, 1)^T \in R^n$. To achieve this, we first give the lemma required.

\textbf{Lemma 4} [44] (Distance / Inner Products Estimation of two vectors) Assume that the unitary operators $U|i\rangle\langle 0| = |i\rangle|x_j\rangle$ and $V|j\rangle\langle 0| = |j\rangle|x_j\rangle$ can be performed in time $T$, and the norms $\|x_i\|$ and $\|x_j\|$ are known. Then there is a quantum algorithm that can operate

$$|i\rangle|j\rangle|0\rangle \rightarrow |i\rangle|j\rangle\|x_i - x_j\|^2$$

or

$$|i\rangle|j\rangle\frac{1}{\sqrt{2}}(|0\rangle|x_i\rangle + |1\rangle|x_j\rangle)|0\rangle \rightarrow$$

$$|i\rangle|j\rangle\frac{1}{\sqrt{2}}(|0\rangle|x_i\rangle + |1\rangle|x_j\rangle)|x_i - x_j\rangle$$

with probability at least $1-2\delta$ for any $\delta$ with complexity $O\left(\|x_i\| \|x_j\| T \log(1/\delta) / \varepsilon\right)$, where $\varepsilon$ is the error of $\|x_i - x_j\|^2$ or $x_i \cdot x_j$.

\textbf{(C.1) The process of the quantum algorithm}

(1). Prepare the quantum state

$$|0\rangle_1 \frac{1}{n} \sum_{i,j=1}^{n} |i\rangle_2 |j\rangle_3 |0\rangle_4 |0\rangle_5 |0\rangle_6. \quad (39)$$

(2). The Distance Estimation algorithm of Lemma 4 is applied to the second, third and fourth registers, we get

$$|0\rangle_1 \frac{1}{n} \sum_{i,j=1}^{n} |i\rangle_2 |j\rangle_3 \|x_i - x_j\|^2 |4\rangle_4 |0\rangle_5 |0\rangle_6. \quad (40)$$

(3). Perform $\exp(-\lambda x)$ gate (See the detailed analysis in Appendix B) for the fourth register to produce

$$|0\rangle_1 \frac{1}{n} \sum_{i,j=1}^{n} |i\rangle_2 |j\rangle_3 \exp(-\lambda \|x_i - x_j\|^2) |4\rangle_4 |0\rangle_5 |0\rangle_6. \quad (41)$$
In fact, according to Eq. (1), we have $w_{ij} = 0$ when $i = j$. However, for Eq. (41), we have $w_{ij} = 1, i = j$. Therefore, we need to perform quantum amplitude amplification algorithm to discard $w_{ij} = 1, i = j$.

For convenience, we rewrite Eq. (41) as

$$
|0\rangle_1 \sqrt{n^2 - n} \sum_{i,j=1}^{n} |i\rangle |j\rangle |w_{ij}\rangle
+ \sqrt{n} \sum_{i,j=1}^{n} |i\rangle |j\rangle |w_{ij}\rangle |2,3,4\rangle |5\rangle |0\rangle_6.
$$

(4). Run the quantum amplitude amplification algorithm to generate

$$
|0\rangle_1 \sqrt{n^2 - n} \sum_{i,j=1}^{n} |i\rangle |j\rangle |w_{ij}\rangle |4\rangle |5\rangle |0\rangle_6.
$$

(5). Perform Hadamard gate $H$ on the first register

$$
\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \sqrt{n^2 - n} \sum_{i,j=1}^{n} |i\rangle |j\rangle |w_{ij}\rangle |0\rangle_6.
$$

(6). Apply the controlled unitary operator $|0\rangle |1\rangle \otimes I_{2,3} \otimes R_w (|4\rangle |5\rangle |6\rangle + |1\rangle |1\rangle |2\rangle - |6\rangle |2\rangle - |5\rangle |2\rangle)$, where $R_w$ is a controlled rotation operator, and uncompute the fourth register, we get

$$
\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle |2\rangle + \frac{1}{\sqrt{n-1}} \sum_{j \neq i, j=1}^{n} |j\rangle |3\rangle |w_{ij}\rangle |0\rangle_6
+ \sqrt{1 - (w_{ij})^2} |1\rangle_5 + |1\rangle_1 \frac{1}{\sqrt{n-1}} \sum_{j \neq i, j=1}^{n} |j\rangle |3\rangle |0\rangle_5 |5\rangle_6
:= \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle |2\rangle (|0\rangle_1 |\varphi\rangle_3 + |1\rangle_1 |\psi\rangle_3_5) |0\rangle_6.
$$

where

$$
|\varphi\rangle_3 = \frac{1}{\sqrt{n-1}} \sum_{j \neq i, j=1}^{n} |j\rangle |3\rangle |w_{ij}\rangle |0\rangle_5 + \sqrt{1 - (w_{ij})^2} |1\rangle_5,
$$

$$
|\psi\rangle_3_5 = \frac{1}{\sqrt{n-1}} \sum_{j \neq i, j=1}^{n} |j\rangle |3\rangle |0\rangle_5.
$$

Note that the inner products of $|\varphi\rangle$ and $|\psi\rangle$ is

$$
\langle \varphi |\psi \rangle = \sum_{i \neq j, i,j=1}^{n} w_{ij} = \frac{d_{ii}}{n-1}.
$$

(7). Apply the Inner Products Estimation algorithm of Lemma 4 and undo the redundant registers, we have

$$
\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle |2\rangle \left( |0\rangle_1 |\varphi\rangle_3 + |1\rangle_1 |\psi\rangle_3_5 \right) |\langle \varphi |\psi \rangle_6.
$$

(8). Attach a register, then perform a controlled rotation operator $R_p$ and uncompute the first, third, fifth and sixth registers to produce

$$
\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle |2\rangle \left( |\varphi\rangle_3 + |\psi\rangle_3_5 \right) |0\rangle_6.
$$

(9). Run the quantum amplitude amplification algorithm to get

$$
\frac{1}{\sqrt{\tau}} \sum_{i=1}^{n} |i\rangle |2\rangle |0\rangle_0, \quad (50)
$$

where $\tau = \sum_{j \neq i, j=1}^{n} \exp(-\lambda |x_i - x_j|^2)$.

(10). Apply CNOT gate to the zeroth register

$$
\frac{1}{\sqrt{\tau}} \sum_{i=1}^{n} \sqrt{d_{ii}} |i\rangle |2\rangle |0\rangle := |\phi\rangle. \quad (51)
$$

(11). Take partial trace for the zeroth register to get

$$
\rho_2 := \text{Tr}_0 (|\phi\rangle \langle \phi |) = \frac{1}{\tau} \sum_{i=1}^{n} d_{ii} \frac{|2\rangle |i\rangle}{\text{Tr}(D)}. \quad (52)
$$

According to Lemma 1, the process of producing $|\phi\rangle$ also can be regarded as an $(2 \log n + 5)$-qubits unitary operator $G_2$ which realize $G_2 |\phi\rangle \otimes \log n \otimes \log \rightarrow |\phi\rangle$, s.t., $\text{Tr}_0 (|\phi\rangle \langle \phi |) = \rho_2$. Thus $V_2 := (G_2 \otimes I_{1-6})(I_0 \otimes \text{SWAP}_{1-6})(G_2 \otimes I_{1-6})$ is an $(1,2 \log n + 5, 2\epsilon)$-block-encoding of $p_2$, where $\epsilon$ is the error that generate $|\phi\rangle$ with a detailed analysis for $\epsilon_2$ in (C.2). The whole quantum circuit is shown in Fig. 4.

(C.2) The complexity of the algorithm
In step(1), we perform $2\log n$ Hadamard gates on the second and third registers of the initial state $|0\rangle_1|0\rangle_2^\otimes \log n |0\rangle_3^\otimes \log n |0\rangle_4^\otimes |0\rangle_5^\otimes |0\rangle_6$ to produce the quantum states of Eq.(39). Thus, the complexity of step(1) is $2\log n$.

In step(2), according to Lemma4, the complexity is

$$O\{[(\max_i \|x_i\|)^2 \text{poly log}(mn/\varepsilon_x) \log(1/\delta_1)]/\varepsilon_d\},$$  \hspace{1cm} (53)

where $1 - 2\delta_1$ is the probability of success of the algorithm with any $\delta_1$ and $\varepsilon_d$ is the error of $\|x_i - x_j\|^2$.

In step(3), the complexity of the exp($-\lambda x$) gate is $O(\text{poly log}(1/\varepsilon))$ (with details given in Appendix B), which is smaller than $O(1/\varepsilon_d)$ caused by the step(2).

Therefore we can ignore the complexity of these gates.

In step(4), the probability amplitude of the target states of Eq.(42) is $O(\text{poly log}(1/\varepsilon))$ (Lemma4), which has complexity $O(1)$.

In step(5)-step(6), it contains one Hadamard gate and a controlled rotation operator $R_{\omega}$ which has complexity $O(1)$.

Finally, we analyze the error that produces $|\phi\rangle$ is

$$||\phi\rangle - |\phi\rangle ||^2 = || \frac{1}{\sqrt{\tau}} \sum_{i=1}^{n} \{ [d_{ii} - d_{ii}] |i\rangle_0 |i\rangle_0 - \langle i | i \rangle \} ||^2$$

$$= \frac{1}{\tau} \sum_{i=1}^{n} (n - 1) |\langle \sqrt{\phi} | \psi \rangle - \langle \sqrt{\phi} | \psi \rangle^2$$

$$= \frac{1}{\tau} \sum_{i=1}^{n} (n - 1) \frac{1}{2\sqrt{\xi_0}} (|\langle \sqrt{\phi} | \psi \rangle - \langle \sqrt{\phi} | \psi \rangle^2)$$

$$\leq \frac{1}{\tau} \sum_{i=1}^{n} (n - 1) \frac{1}{4\xi_0} (\lambda^2 \varepsilon_d^2 \leq \lambda^2 e_d^2/4r),$$  \hspace{1cm} (57)

where the second equation comes from Eq.(47), the third equation follows from the Lagrange’s mean value theorem (45), $\xi_0$ takes value from $|\sqrt{\phi} | \psi \rangle$ to $|\sqrt{\phi} | \psi \rangle$, and the last inequality holds by $r \geq n(n-1)r$.

The complexity of each step of the algorithm is shown in Table I.

| TABLE I. The complexity of each step of the algorithm |
|-----------------|-----------------|
| steps           | complexity       |
| (1)             | $2\log n$       |
| (2)             | $O\{[(\max_i \|x_i\|)^2 \text{poly log}(mn/\varepsilon_x) \log(1/\delta_1)]/\varepsilon_d\}$ |
| (3)             | $O(\text{poly log}(1/\varepsilon))$ |
| (4) - (6)       | $O(1)$ |
| (7)             | $O\{[(\max_i \|x_i\|)^2 \text{poly log}(mn/\varepsilon_x) \nu]/(\varepsilon_d^2)\}$ |
| (8) - (11)      | $O(1/\sqrt{\tau})$ |

To make the error of $|\phi\rangle$ equal to $\varepsilon_2$, we have

$$||\phi\rangle - |\phi\rangle ||^2 \leq \varepsilon_d/2\sqrt{\tau} := \varepsilon_2.$$  \hspace{1cm} (58)

Thus, we can obtain $1/\varepsilon_d = \lambda/2\sqrt{ve_d}$.

Putting all complexity together and letting $\varepsilon_x = \varepsilon_d$, $\delta_1 = \delta_2 = O(\text{poly log} n)$, we get

$$O\{\lambda^{\max_i \|x_i\|} \text{poly log}(\lambda mn)/(\sqrt{ve_d})\} := O(c_2).$$  \hspace{1cm} (59)

D. Implement the quantum simulation of $L$

Here, we first implement the quantum simulation of $L$ in the case of $\|x_i\| \neq 1, i = 1, \ldots, n$. Similarly, we provide a detailed analysis for the case of $\|x_i\| = 1$ in Appendix C.

According to $L = D - W$, we have

$$\mathcal{L} = \frac{L}{\text{Tr}(L)} = \frac{D}{\text{Tr}(D)} - \frac{W}{\text{Tr}(D)},$$  \hspace{1cm} (60)

where the last equation comes from $\text{Tr}(L) = \text{Tr}(D)$.

Due to $\rho_1 = (W + I)/\text{Tr}(W + I)$ and $\rho_2 = D/\text{Tr}(D)$, we can obtain

$$\mathcal{L} = \rho_2 - \frac{\text{Tr}(W + I)}{\text{Tr}(D)} \rho_1 + \frac{I}{\text{Tr}(I)} \frac{\text{Tr}(I)}{\text{Tr}(D)}.$$  \hspace{1cm} (61)
For $I/\text{Tr}(I)$, we can prepare the quantum state $|\tau\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle_1 |i\rangle_2$ can be prepared with $O(\log n)$ Hadamard and CNOT gates. Thus, we have

$$\frac{I}{\text{Tr}(I)} = \text{Tr}_2(|\tau\rangle \langle \tau|) = \frac{1}{n} \sum_{i=1}^{n} |i\rangle_1 |i\rangle_1 \equiv \rho_3.$$  

According to Lemma 1, we also get an $(1, \log n + 1, 0)$-block-encoding of $\rho_3$, that is, $V_3 := (G_3^l \otimes I_l)(I_2 \otimes \text{SWAP}_3)(G_3 \otimes I_l)$ where $G_3(0)^{\otimes \log n}(0) \mapsto |\tau\rangle$, s.t. $\text{Tr}_2(|\tau\rangle \langle \tau|) = \rho_3$.

Thus, we obtain

$$\mathcal{L} = -c\rho_1 + \rho_2 + c\rho_3,$$

where $c = \text{Tr}(W + I)/\text{Tr}(D) = \text{Tr}(I)/\text{Tr}(D)$, $0 < c < 1$, and $\text{Tr}(D)$ can be effectively evaluated in Sec. III C. This can be viewed as a linear combination of block-encoded operators.

By Definition 2, let $y = (-c, 1, c)$ and $||y||_1 \leq \beta = 3$. Let $b = 2, c_j = d_j = \sqrt{y_j}$, $j = 1, 2, 3$. We can effectively construct an $(3, 2, \epsilon_y)$-state-preparation-pair $(P_L, P_R)$ of $y$ that satisfies the requirements of Definition 2.

In addition, we construct a $(\log n + l + 2)$-qubits unitary $Q = \sum_{j=1}^{L} |j\rangle \langle j| \otimes V_j + (I - \sum_{j=1}^{L} |j\rangle \langle j|) \otimes I_l \otimes I_{\log n}$ such that for $j = 1, 2, 3$, where $l = \max\{2p + 3 + \log[n(p + 1)], 2\log n + 5\}$, we have that $V_j$ is an $(1, l, \epsilon_l)$-block-encoding of $\rho_j$, where $\epsilon_l$ depends on the value of $l$.

According to Lemma 2, we can implement unitary $G$ which is an $(3, l + 2, \epsilon_y + 3\epsilon_l)$-block-encoding of $\mathcal{L}$, with a single use of $Q, P_L$, and $P_R$.

Combining Theorem 1, we can implement an $\epsilon$-precise the Hamiltonian simulation unitary operator which is an $(1, l + 4, \epsilon)$-block-encoding of $\exp(-i\mathcal{L}t)$ with

$$O[3t + \log(1/\epsilon)/\log(1/\epsilon)]$$  

uses of controlled-$G$ or its inverse and with $O[3(l + 2)t + (l + 2)\log(1/\epsilon)/\log(1/\epsilon)]$ two-qubit gates, where $\epsilon = 2t(\epsilon_y + 3\epsilon_l)$.

### E. Extract the eigeninformation of $L$

In this section, we use quantum phase estimation algorithm to extract the $1 \leq d \leq (n - 1)$ non-zero eigenvalues and eigenvectors of $\mathcal{L}$.

Suppose that the eigendecomposition form of $\mathcal{L}$ is

$$\mathcal{L} = \sum_{j=1}^{n} \gamma_j |u_j\rangle \langle u_j|,$$

where $\{\gamma_j\}_{j=1}^{n}$ and $\{|u_j\rangle\}_{j=1}^{n}$ are the eigenvalues and the corresponding eigenvectors of $\mathcal{L}$, respectively.

Our algorithm works as the following steps.

#### (E.1) The steps of the quantum algorithm

1. Perform Hadamard and CNOT gates on the initial states $|0\rangle_{1}^{\otimes \log n} |0\rangle_{2} |0\rangle_{3}$ to produce the quantum state

$$|\omega\rangle = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} |j\rangle_{1} |j\rangle_{2} \otimes |0\rangle_{3}.$$  

2. Take partial trace for the second register to get

$$\text{Tr}_2(|\omega\rangle \langle \omega|) = \frac{1}{n} \sum_{j=1}^{n} |j\rangle_{1} |j\rangle_{1} \otimes |0\rangle_{3} |0\rangle_{3}. $$

Although we don’t know the information of the state $|u_j\rangle$, $j = 1, \cdots, n$ in advance, we have

$$\frac{1}{n} \sum_{j=1}^{n} |j\rangle_{1} \langle j| = \frac{I}{\text{Tr}(I)} = \frac{1}{n} \sum_{j=1}^{n} |u_j\rangle \langle u_j|.$$  

Therefore, we can obtain

$$\frac{1}{n} \sum_{j=1}^{n} |j\rangle_{1} \langle j| \otimes |0\rangle_{0} = \frac{1}{n} \sum_{j=1}^{n} |u_j\rangle \langle u_j| \otimes |0\rangle_{0}.$$  

3. Run the quantum phase estimation algorithm by simulating $\exp(-i\mathcal{L}t)$ to reveal the eigenvalues and eigenvectors of $\mathcal{L}$

$$\frac{1}{n} \sum_{j=1}^{n} |u_j\rangle \langle u_j| \otimes |\gamma_j\rangle \langle \gamma_j|.$$  

4. Use the quantum algorithm for finding the minimum to reveal the $d$ minimized nonzero eigenvalues $\gamma_j$, and the corresponding eigenvectors $|u_j\rangle$, $j = 1, \cdots, d$.

#### (E.2) The complexity of the algorithm

In step 1, the complexity is $O(\log \log n)$, which comes from Hadamard and CNOT gates.

In step 3, according to Ref. [9], it takes $t = O(1/\epsilon)$ times to yields the eigeninformation of $\mathcal{L}$ with accuracy $\epsilon$. Therefore, the complexity of the algorithm is

$$O(\max\{c_1, c_2\}/3\epsilon),$$

where $c_1$ and $c_2$ are shown in Eq. (36) and Eq. (59), which represent the complexity of generating the block-encodings of $W$ and $D$, respectively.

In step 4, to reveal the $1 \leq d \leq (n - 1)$ minimized nonzero eigenvalues and the corresponding eigenvectors of $\mathcal{L}$, we need to run $O(d)$ times of the algorithm with the query complexity $O(\sqrt{n})$ for find the minimum values. Thus the total complexity is $O(d\sqrt{n})$.

Putting all the complexity together, we can obtain the complexity of the whole quantum algorithm is

$$O(d\sqrt{n} \max\{c_1, c_2\}/3\epsilon).$$

When $a, p, C, \lambda, \sqrt{\gamma}$ and $\max_i \|x_i\|$ are all $O(1)$, and letting $1/\epsilon_1 = 1/\epsilon_2 = 1/\epsilon = O(\log \log n)$, our quantum algorithm takes time

$$O(d\sqrt{n} \log(\log n)).$$
It is shown that our algorithm achieve polynomial speedup in $n$ and exponential speedup in $m$ compared with the classical algorithm whose complexity is $O(mn^2 + dn^3)$.

In addition, our algorithm can also extract the eigeninformation of $W$, which is of great significant [31] [34]. See the detailed analysis in Appendix D.

IV. GENERALIZATION: SOLVE THE EIGENPROBLEM OF $L_s$ AND $L_r$

In this section, we extend our quantum algorithm to solve the eigenproblem of $L_s$ and $L_r$ in Eq. (3). We assume that the eigendecomposition form of $L_s$ is

$$L_s = \sum_{i=1}^{n} \mu_i |v_i\rangle \langle v_i|,$$

(74)

where $\{\mu_i\}_{i=1}^{n}$ and $\{|v_i\rangle\}_{i=1}^{n}$ are the eigenvalues and the corresponding eigenvectors of $L_s$, respectively.

According to the properties of $L_s$ and $L_r$ [29], we know that the eigenvalues of $L_s$ are also $\{\mu_i\}_{i=1}^{n}$ and the corresponding eigenvectors of $L_r$ are $\{D^{-\frac{1}{2}} |v_i\rangle\}_{i=1}^{n}$. In particular, $\mu_1 = 0$ is the unique zero eigenvalue of $L_s$, and the corresponding eigenvector is $|v_1\rangle = D^{-\frac{1}{2}} 1$. Therefore, we only need to extract the eigeninformation of $L_r$.

To achieve this, our core task is to first utilize the quantum simulation of $L_s$. According to Eq. (3), we have

$$L_s = \frac{D}{\text{Tr}(D)^{\frac{1}{2}}} L \frac{D}{\text{Tr}(L)^{\frac{1}{2}}} \frac{D}{\text{Tr}(D)^{\frac{1}{2}}} = (p_2)^{-\frac{1}{2}} L (p_2)^{-\frac{1}{2}},$$

(75)

where the first equation comes from $\text{Tr}(L) = \text{Tr}(D)$.

We have constructed the block-encodings of $p_2$ and $L$, respectively. To design the block-encoding of $L_s$, we propose the following lemma based on Ref. [29].

Lemma5 (Block-encoding of $A^{-\frac{1}{2}}BA^{-c}$)

Let $c \in (0, \infty)$, $s_1 \in (0, 1/2]$, and let $A$ and $B$ are Hermitian matrices, and $A$ satisfy $I/\sqrt{\kappa} \preceq A \preceq I$ where $\kappa \geq 2$. Let $\zeta_i = O(\kappa^{\frac{1}{\kappa}}, \max(1, c) \log(\kappa^{1/c} s_1) \log^2(\kappa(c+1) \log(1/s_1)))$. And $U$ with $T_U$ elementary gates and $V$ with $T_V$ elementary gates are the $(a_1, a_1, \zeta_1)$-block-encoding of $A$ and the $(a_2, a_2, \zeta_2)$-block-encoding of $B$, respectively. Then we can implement a unitary $F$ that is an $(4\kappa^2 a_2 \sigma_{02}, 2a_1 + a_2 + O(\log(\kappa^2 \max(1, c) \log(\kappa^{1/c} s_1)))$, $4\kappa^2 a_2 \sigma_{02} + 4\kappa^2 \zeta_2)$-block-encoding of $A^{-\frac{1}{2}}BA^{-c}$ in cost

$$O(\max(1, c) [2a_1 \kappa \log^{\frac{\kappa}{s_1}}(a_1 + T_U) + \kappa n] + T_V).$$

(76)

where $\eta = \log^2([\max(1, c)\kappa \max(1, c)]/s_1]$.

The proof of Lemma5 can be obtained by combining Lemma4 and Lemma8 in Ref. [29].

According to Lemma5, for $L_s$, we know that $c = 1/2$, $A = p_2$, $B = L$. And we assume that $p_2$ and $L$ satisfy the conditions of Lemma5. We have obtained that the unitary operator $V_2$ which is an $(1, 2 \log n + 5, 2\epsilon_2)$-block-encoding of $p_2$ and the unitary operator $G$ which is an $(3, l + 2, \epsilon_2 + 3\epsilon_1)$-block-encoding of $L$, respectively.

Therefore, we can implement a unitary $F$ that is an $(12\kappa, 4 \log n + 12 + l + O(\log(\kappa^2 \log(\frac{s_1}{\kappa}))), \epsilon_f)$-block-encoding of $L_s$ in cost

$$O[2\kappa \log(\frac{s_1}{\kappa}) (2 \log n + 5 + c_2) + \kappa \log^2(\frac{s_1}{\kappa}) + T_L],$$

(77)

where $\epsilon_f = 12\kappa^2 \zeta_1 + 4\kappa(\epsilon_2 + 3\epsilon_1)$ and $T_L = \max\{c_1, c_2\}$ is the complexity of produces $G$.

Finally, combining Theorem1 and Sec. III E, we can solve the eigenproblem of $L_s$.

V. DISCUSSION

We apply our algorithm to solve the eigenproblem of the weighted matrix $W$ in Appendix D. Owing to the Gaussian kernel matrix $K$ satisfies $K = W + I$, our algorithm can also be used to solve the eigenproblem of $K$.

Clearly, our algorithm is an universal quantum algorithm for solving the eigenproblem of $K$ under the circuit model, while the quantum algorithms proposed in Refs. [45] [49] are formulated with the generalized coherent states that can construct $K$ in the language of quantum optics. This makes them likely not universal quantum algorithms. Besides, compared to Ref. [50], our algorithm can not only process the scenario where the modulus length of each sample data point $x_i$ is not equal to 1, i.e., $||x_i|| \neq 1$, $i = 1, 2, \ldots, n$, but also provide the optimal Hamiltonian simulation algorithm for $K$ to reduce the algorithm’s dependence on simulation error.

Similarly, our algorithm can also be extended to solve arbitrary nonlinear kernel matrix, which has a wide range of applications in classification, dimensionality reduction, regression and so on.

Besides, the advantages of our algorithms usually rely on a fault-tolerant quantum computer, which may take a long time horizon to implement. Recently, a few scholars have employed variational quantum algorithms [51] [52] which can be implemented on the Noisy Intermediate-Scale Quantum (NISQ) devices [53] to solve problems related to the Laplacian matrix $L$. In 2020, Shimane et al. proposed a variational Laplacian eigenmap algorithm [54], and demonstrate that it is possible to use the embedding for graph machine learning tasks through implementing a quantum classifier on the top of it. However, their algorithm cannot be used directly to deal with the case where the element of $W$ is a Gaussian similarity function. In 2022, Li et al. designed a Laplacian eigenmap algorithm based on variational quantum generalized eigensolver [55] and their simulation results demonstrate that the proposed algorithm has good convergence. However, their algorithm employs the controlled SWAP test and maximum searching algorithm to construct the
weight matrix, which cannot be implemented on NISQ devices. In addition, how to design a good strategy to suppress the barren plateau phenomenon in variational quantum algorithms [25] is also a thorny problem. Therefore, designing a variational quantum algorithm that can solve the above problems to solve eigenproblem of the Laplacian matrix \( L \) may be an important direction of future work.

VI. CONCLUSION

In summary, we designed an efficient quantum algorithm to solve the eigenproblem of the Laplacian matrix \( L \) of a fully connected graph. Specifically, we designed special controlled unitary operators to construct the block-encodings of operators containing the information of \( W \) and \( D \) respectively, and further obtain the block-encoding of \( L \). Then we employed the optimal Hamiltonian simulation technology based on the block-encoding framework to realize the quantum simulation of \( L \). Finally, we adopted the quantum phase estimation algorithm to extract the eigenvalues and eigenvectors of \( L \). It is shown that compared with the classical algorithm, our quantum algorithm achieve polynomial speedup in the number of vertices and exponential speedup in the dimension of each vertex. Additionally, we also extended our algorithm to solve the eigenproblem of \( W, L_s \), and \( L_r \).

We expect that our quantum algorithm and the techniques mentioned, such as the quantum simulation technique to implement the matrix \( A^{-1}BA^{-c} \) and the analysis of the error propagation of the quantum state, can provide new ideas for quantum algorithms to solve other problems. Furthermore, exploring the application of our quantum algorithms to real data is a goal worth considering in the future.

ACKNOWLEDGMENTS

We thank Shanwei Ma for useful discussions on the subject. This work is supported by the Beijing Natural Science Foundation (Grant No. 4222031), the National Natural Science Foundation of China (Grant Nos. 61976024, 61972048, 62006105), BUPT innovation and entrepreneurship support program (Grant No. 2021-YC-A206).

Appendix A: A detailed analysis of the last inequality of Eq.(17) and Eq.(34)

Before we analyze the last inequality of Eq.(17) and Eq.(34) in detail, we first give the lemmas as follows:

LemmaA1 (Error propagation of quantum states)

If \( ||\langle x|| - |y\rangle||_2 \leq \varepsilon \), then \( ||a|x\rangle - b|y\rangle||_2 \leq (a - b)||x||_2 + b||x\rangle - |y\rangle||_2 \leq a - b + b\varepsilon \), where \( a, b \) are any positive real numbers.

Proof:

\[
||a|x\rangle - b|y\rangle||_2 = ||a|x\rangle - b|x\rangle + b|x\rangle - b|y\rangle||_2 \leq ((a - b)||x||_2 + b||x\rangle - |y\rangle||_2 \leq (a - b)||x||_2 + b||x\rangle - |y\rangle||_2 \leq a - b + b\varepsilon.
\]

(A1)

LemmaA2 (Error propagation of tensor products of quantum states)

If \( ||x|| - |y||_2 \leq \varepsilon \), then \( ||x^{\otimes p} - |y\rangle^{\otimes p}||_2 \leq p\varepsilon \), where \( p \) is any positive real numbers.

Proof: we prove it by mathematical induction.

When \( p = 2 \), we have

\[
||x^{\otimes 2} - |y\rangle^{\otimes 2}||_2 = ||x\rangle \otimes ((x\rangle - |y\rangle) + (|x\rangle - |y\rangle) \otimes |y\rangle\rangle_2 \leq ||x\rangle||_2 ||x\rangle - |y\rangle||_2 + ||x\rangle - |y\rangle||_2 ||y\rangle||_2 \leq 2\varepsilon.
\]

(A2)

Assume that when \( k = p - 1 \), we get \( ||x^{\otimes (p-1)} - |y\rangle^{\otimes (p-1)}||_2 \leq (p - 1)\varepsilon \).

When \( k = p \), we obtain

\[
||x^{\otimes p} - |y\rangle^{\otimes p}||_2 = ||x^{\otimes (p-1)} \otimes (x\rangle - |y\rangle) + (|x\rangle^{\otimes (p-1)} - |y\rangle^{\otimes (p-1)}) \otimes |y\rangle\rangle_2
\]

\[
= ||x^{\otimes (p-1)}||_2 ||x\rangle - |y\rangle||_2 + ||(|x\rangle^{\otimes (p-1)} - |y\rangle^{\otimes (p-1)})\otimes |y\rangle\rangle_2 \leq \varepsilon + (p - 1)\varepsilon = p\varepsilon.
\]

(A3)

Then we use the lemmas above to analyze the last inequality of Eq.(17) and Eq.(34). At the same time, we expect that the lemmas above can be used to deal with the error analysis of other quantum algorithms in the future.
(i) The last inequality of Eq.(17):

\[ ||\hat{\Phi}(x_i) - \Phi(x_i)||_2 \]

\[ = ||\frac{1}{\sqrt{a}} \sum_{k=0}^{p} \sqrt{a_k} |k \rangle |0\rangle^{\otimes p-k} |\hat{x}_i \rangle^{\otimes k} - \frac{1}{\sqrt{a}} \sum_{k=0}^{p} \sqrt{a_k} |k \rangle |0\rangle^{\otimes p-k} |\hat{x}_i \rangle^{\otimes k}||_2 \]

\[ \leq (\frac{1}{\sqrt{a}} - \frac{1}{\sqrt{a}}) ||\sum_{k=0}^{p} \sqrt{a_k} |k \rangle |0\rangle^{\otimes p-k} |\hat{x}_i \rangle^{\otimes k}||_2 + \frac{1}{\sqrt{a}} ||\sum_{k=0}^{p} \sqrt{a_k} |k \rangle |0\rangle^{\otimes p-k} |\hat{x}_i \rangle^{\otimes k} - \sum_{k=0}^{p} \sqrt{a_k} |k \rangle |0\rangle^{\otimes p-k} |\hat{x}_i \rangle^{\otimes k}||_2 \]

\[ = (\frac{1}{\sqrt{a}} - \frac{1}{\sqrt{a}}) \sum_{k=0}^{p} \sqrt{a_k} |k \rangle |0\rangle^{\otimes p-k} |\hat{x}_i \rangle^{\otimes k}||_2 + \frac{1}{\sqrt{a}} ||\sum_{k=0}^{p} \sqrt{a_k} |k \rangle |0\rangle^{\otimes p-k} (\sqrt{a_k} |\hat{x}_i \rangle^{\otimes k} - \sqrt{a_k} |\hat{x}_i \rangle^{\otimes k})||_2 \]

\[ = (1 - \frac{\sqrt{a}}{\sqrt{a}}) \sum_{k=0}^{p} (\sqrt{a_k} - \sqrt{a_k})^2 + \frac{1}{\sqrt{a}} \sum_{k=1}^{p} \sqrt{a_k} ||(\sqrt{a_k} |\hat{x}_i \rangle^{\otimes k} - \sqrt{a_k} |\hat{x}_i \rangle^{\otimes k})||_2 \]

\[ \leq \epsilon_a + \frac{1}{\sqrt{a}} \sum_{k=0}^{p} (\sqrt{a_k} - \sqrt{a_k})^2 \leq \epsilon_a \]

where the first inequality comes from applying the LemmaA1 and the penultimate inequality comes from

\[ ||\frac{1}{\sqrt{a}} \sum_{k=0}^{p} \sqrt{a_k} |k \rangle - \frac{1}{\sqrt{a}} \sum_{k=0}^{p} \sqrt{a_k} |k \rangle||_2 \leq (\frac{1}{\sqrt{a}} - \frac{1}{\sqrt{a}}) ||\sum_{k=0}^{p} \sqrt{a_k} |k \rangle||_2 + \frac{1}{\sqrt{a}} ||\sum_{k=0}^{p} (\sqrt{a_k} - \sqrt{a_k}) |k \rangle||_2 \]

\[ \leq (1 - \frac{\sqrt{a}}{\sqrt{a}}) \sum_{k=0}^{p} (\sqrt{a_k} - \sqrt{a_k})^2 \leq \epsilon_a \]

and the LemmaA2.

(ii) The last inequality of Eq.(34):

\[ ||\hat{\Psi}(x_i) - |\Psi(x_i)||_2 \]

\[ = ||\exp(-\lambda ||x_i||^2) \cdot (\sum_{k=0}^{p} \sqrt{a_k} |k \rangle |0\rangle^{\otimes p-k} |\hat{x}_i \rangle^{\otimes k} - \sum_{k=0}^{p} \sqrt{a_k} |k \rangle |0\rangle^{\otimes p-k} |\hat{x}_i \rangle^{\otimes k})||_2 \]

\[ = \exp(-\lambda ||x_i||^2) \cdot ||(\sqrt{a_0} - \sqrt{a_0}) |0\rangle^{\otimes p} + \sum_{k=1}^{p} |k \rangle |0\rangle^{\otimes p-k} |x_i \rangle^{k} (\sqrt{a_k} - \sqrt{a_k}) |\hat{x}_i \rangle^{\otimes k} + \sqrt{a_k} (|\hat{x}_i \rangle^{\otimes k} - |x_i \rangle^{\otimes k})||_2 \]

\[ \leq \sum_{k=0}^{p} (\sqrt{a_k} - \sqrt{a_k})^2 ||x_i ||^{2k} \leq \sum_{k=1}^{p} \sqrt{a_k} ||x_i ||^{k} |||\hat{x}_i \rangle^{\otimes k} - |x_i \rangle^{\otimes k}||_2 \]

\[ \leq \sum_{k=0}^{p} (\sqrt{a_k} - \sqrt{a_k})^2 ||x_i ||^{2k} + \sum_{k=1}^{p} \sqrt{a_k} ||x_i ||^{k} k \epsilon_x, \]

where the last inequality comes from LemmaA2.
can implement the following transformation: $\Pi_k^+|\lambda\rangle|0\rangle = |\lambda\rangle|\lambda x\rangle$. It can be represented as $\Pi_k^+ = (I \otimes I \otimes \text{QFT}^\dagger) \cdot \pi^+ \cdot (I \otimes I \otimes \text{QFT})$, where $\pi^+|a\rangle|b\rangle|\phi(c)\rangle = |a\rangle|b\rangle|\phi(c \pm a \cdot b)\rangle$ with $|\phi(c)\rangle = \text{QFT}(c)$ and $a, b, c$ are the input qubits.

When $\|\mathbf{x}_i\| > 1$, we have

$$
\|\hat{\Psi}(\mathbf{x}_i) - |\Psi(\mathbf{x}_i)\| \leq \|\mathbf{x}_i\|_p \left[ \sum_{k=0}^{p} (\sqrt{a_k} - \sqrt{\bar{a}_k})^2 + \|\mathbf{x}_i\|_p^2 \cdot \left( \frac{1}{\sqrt{a}} \sum_{k=1}^{p} \sqrt{a_k} k \bar{e}_x \right) \right],
$$

where the second inequality comes from the Eq.(A5).

When $0 < \|\mathbf{x}_i\| < 1$, we get

$$
\|\hat{\Psi}(\mathbf{x}_i) - |\Psi(\mathbf{x}_i)\| \leq \left[ \sum_{k=0}^{p} (\sqrt{a_k} - \sqrt{\bar{a}_k})^2 + \sqrt{a} \cdot \left( \frac{1}{\sqrt{a}} \sum_{k=1}^{p} \sqrt{a_k} k \bar{e}_x \right) \right] \leq \sqrt{a} [\bar{e}_a + (1 + 2 + \cdots + p) \bar{e}_x],
$$

Appendix B: The quantum circuit of function $f(x) = \exp(-\lambda x)$

The Taylor expansion of $f(x) = \exp(-\lambda x)$ is shown as:

$$
\exp(-\lambda x) = 1 - \lambda x + \frac{(\lambda x)^2}{2!} + \cdots + \frac{(-1)^k(\lambda x)^k}{k!} + \frac{(-1)^{k+1}(\lambda^2 x)^{k+1}}{(k + 1)!}, \xi \in (0, x).
$$

According to Taylor’s theorem, we know that the $(k + 1)$th term in the expansion is $[(-1)^{k+1}(\lambda^2 x)^{k+1}] / [(k + 1)!]$ and the derivative of $f(x)$ are bounded. We can design the quantum circuit of $f(x)$ by the Quantum Multiply-Adder (QMA), which is shown in Fig. 5, with $O(\text{poly log}(1/\epsilon))$ one- or two-qubits gates, where $\epsilon$ is the accuracy of the algorithm.

Appendix C: Implement the quantum simulation of $L$ in the case of $\|\mathbf{x}_i\| = 1, i = 1, \cdots, n$

In this section, we analyze the quantum simulation of implementing $L$ in the case of $\|\mathbf{x}_i\| = 1, i = 1, \cdots, n$. According to Eq.(16) in Sec.III B, we can obtain $W = \hat{a}(n\rho_0 - I)$. Due to $\text{Tr}(L) = \text{Tr}(D)$, we have

$$
\mathcal{L} = \frac{L}{\text{Tr}(L)} = \frac{D - W}{\text{Tr}(D)} = \frac{D}{\text{Tr}(D)} \rho_0 + \frac{I}{\text{Tr}(I)} \hat{a}\text{Tr}(I) \text{Tr}(D) := \rho_1 - d\rho_0 + \epsilon \rho_3,
$$
where $d = \hat{n} \alpha / \text{Tr}(D)$, $e = \hat{a} \text{Tr}(I) / \text{Tr}(D)$.

Similarly to Sec. III C, we can implement the quantum simulation of $\mathcal{L}$.

**Appendix D: Reveal the eigeninformation of the weight matrix $W$**

In this section, we introduce the algorithm to reveal the eigeninformation of $W$ as follows:

For the case of $\| \mathbf{x}_i \| = 1$, we can obtain $W = \hat{a} (\rho_0 - I)$. Thus, we have

$$W / n = \hat{a} (\rho_0 - I / n) = \hat{a} \rho_0 - \hat{a} \rho_3.$$  \hspace{1cm} (D1)

For the case of $\| \mathbf{x}_i \| \neq 1$, we can obtain $W = \text{Tr}(W + I) \rho_1 - I$. Due to $\text{Tr}(W + I) = n$, Thus, we have

$$W / n = \rho_1 - \rho_3.$$ \hspace{1cm} (D2)

In short, Eq.(D1) and Eq.(D2) can be viewed as a linear combination of block-encoded operators, respectively. Similarly to Sec. III C, we can implement the quantum simulation of $W$. Then we can solve the eigenproblem of $W$ by using the algorithm in Sec. III E.

[1] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu, IEEE Transactions on Artificial Intelligence 2, 109 (2021).
[2] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, AI Open 1, 57 (2020).
[3] D. Alfke and M. Stoll, Data Mining and Knowledge Discovery 35, 1318 (2021).
[4] Y. Romano, M. Eldad, and P. Milanfar, SIAM Journal on Imaging Sciences 10, 1804 (2017).
[5] G. Gilboa and S. Osher, Multiscale Modeling & Simulation 7, 1005 (2009).
[6] K. Wang, K. Zhou, Q. Zhang, J. Shao, B. Hooi, and J. Feng, International Conference on Machine Learning, 11003 (2021).
[7] P. W. Shor, in *Proceedings 35th Annual Symposium on Foundations of Computer Science* (1994) pp. 124–134.
[8] L. K. Grover, Phys. Rev. Lett. 79, 325 (1997).
[9] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. 103, 150502 (2009).
[10] L.-C. Wan, C.-H. Yu, S.-J. Pan, F. Gao, Q.-Y. Wen, and S.-J. Qin, Phys. Rev. A 97, 062322 (2018).
[11] H.-L. Liu, Y.-S. Wu, L.-C. Wan, S.-J. Pan, S.-J. Qin, F. Gao, and Q.-Y. Wen, Phys. Rev. A 104, 022418 (2021).
[12] N. Wiebe, D. Braun, and S. Lloyd, Phys. Rev. Lett. 109, 050505 (2012).
[13] L. Cong and L. Duan, New Journal of Physics 18, 073011 (2016).
[14] C.-H. Yu, F. Gao, and Q. Wen, IEEE Transactions on Knowledge and Data Engineering , 1 (2019).
[15] S. Lloyd, M. Mohseni, and P. Rebentrost, Nature Physics 10, 631 (2014).
[16] S.-J. Pan, L.-C. Wan, H.-L. Liu, Q.-L. Wang, S.-J. Qin, Q.-Y. Wen, and F. Gao, Phys. Rev. A 102, 052402 (2020).
[17] S.-J. Pan, L.-C. Wan, H.-L. Liu, Y.-S. Wu, S.-J. Qin, Q.-Y. Wen, and F. Gao, Chinese Physics B 2 (2022).
[18] N. Liu and P. Rebentrost, Physical Review A 97, 042315 (2018).
[19] M.-C. Guo, H.-L. Liu, Y.-M. Li, W.-M. Li, S.-J. Qin, Q.-Y. Wen, and F. Gao, arXiv preprint arXiv:2109.13820 (2021).
[20] A. Abbas, D. Sutter, C. Zoufal, A.Lucchi, A. Figalli, and S. Woerner, Nature Computational Science 1, 403 (2021).
[21] Y. Huang and X. Li, arXiv preprint arXiv:1611.00760 (2016).
[22] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. 100, 160501 (2008).
[23] F. R. Chung and F. C. Graham, *Spectral graph theory* (American Mathematical Soc., 1997).
[24] U. Von Luxburg, Statistics and computing 17, 395 (2007).
[25] I. Kerenidis and J. Landman, Phys. Rev. A 103, 022418 (2021).
[26] L. Wossnig, Z. Zhao, and A. Prakash, Phys. Rev. Lett. 120, 050502 (2018).
[27] G. H. Low and I. L. Chuang, Quantum 3, 163 (2019).
[28] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (2019) pp. 193–204.
[29] S. Chakraborty, A. Gilyén, and S. Jeffery, arXiv preprint arXiv:1804.01973 (2018).
[30] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information* (American Association of Physics Teachers, 2002).
[31] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, and C. Faloutsos, ACM Transactions on Information and System Security (TISSEC) 10, 1 (2008).
[32] M. Ghorbani, X. Li, S. Zangi, and N. Amraei, Applied Mathematics and Computation 397, 29 (2021).
[33] G. Li, Y. Jiang, W. Jiao, W. Xu, S. Huang, Z. Gao, J. Zhang, and C. Wang, Brain Sciences 10, 92 (2020).
[34] M. Zhou, J. Tan, H. Liao, Z. Wang, and R. Mao, Chaos: An Interdisciplinary Journal of Nonlinear Science 30, 083118 (2020).
[35] M. Belkin and P. Niyogi, Neural computation 15, 1373 (2003).
[36] M. Karasuyama and H. Mamitsuka, Machine Learning 106, 307 (2017).
[37] H. Strange and R. Zwiggelaar, (2014).
[38] C.-H. Yu, F. Gao, S. Lin, and J. Wang, Quantum Information Processing 18, 1 (2019).
[39] L.-C. Wan, C.-H. Yu, S.-J. Pan, S.-J. Qin, F. Gao, and Q.-Y. Wen, Phys. Rev. A 104, 062414 (2021).
[40] M. Kline, Calculus: an intuitive and physical approach (Courier Corporation, 1998).
[41] K.-P. Lin and M.-S. Chen, in Proceedings of the 2011 SIAM International Conference on Data Mining (SIAM, 2011) pp. 211–222.
[42] S. Zhou, T. Loke, J. A. Izaac, and J. Wang, Quantum Information Processing 16, 1 (2017).
[43] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Contemporary Mathematics 305, 53 (2002).
[44] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, Advances in Neural Information Processing Systems 32 (2019).
[45] P. Sahoo and T. Riedel, Mean value theorems and functional equations (World Scientific, 1998).
[46] L. Grover and T. Rudolph, arXiv preprint quant-ph/0208112 (2002).
[47] C. Durr and P. Hoyer, arXiv preprint quant-ph/9607014 (1996).
[48] R. Chatterjee and T. Yu, Quantum Inf. Comput. 17, 1292 (2017).
[49] C. Shao, Quantum Inf. Comput. 19, 609 (2019).
[50] Y. Li, R.-G. Zhou, R. Xu, W. Hu, and P. Fan, Quantum Science and Technology 6, 014001 (2020).
[51] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, Nature Communications 5, 4213 (2014).
[52] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, et al., Nature Reviews Physics 3, 625 (2021).
[53] J. Preskill, Quantum 2, 79 (2018).
[54] S. Thabet and J.-F. Hullo, arXiv preprint arXiv:2011.05128 (2020).
[55] Z.-T. Li, F.-X. Meng, X.-T. Yu, and Z.-C. Zhang, Quantum Information Processing 21, 1 (2022).
[56] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Nature communications 9, 1 (2018).