A Quantum Algorithm for Solving Eigenproblem of the Laplacian Matrix of a Fully Connected Weighted Graph

Hai-Ling Liu, Lin-Chun Wan, Chao-Hua Yu, Shi-Jie Pan, Su-Juan Qin,* Fei Gao,* and Qiao-Yan Wen

Solving eigenproblem of the Laplacian matrix of a fully connected weighted graph has wide applications in data science, machine learning, and image processing, etc. However, this is very challenging because it involves expensive matrix operations. Here, an efficient quantum algorithm is proposed to solve it. Specifically, the optimal Hamiltonian simulation technique based on the block-encoding framework is adopted to implement the quantum simulation of the Laplacian matrix. Then, the eigenvalues and eigenvectors of the Laplacian matrix are extracted by the quantum phase estimation algorithm. The core of this entire algorithm is to construct a block-encoding of the Laplacian matrix. To achieve this, how to construct block-encoding of operators containing the information of the weight matrix and the degree matrix, respectively are shown in detail, and the block-encoding of the Laplacian matrix is further obtained. Compared with its classical counterpart, this algorithm has a polynomial speedup on the number of vertices and an exponential speedup on the dimension of each vertex. It is also shown that this algorithm can be extended to solve the eigenproblem of symmetric (non-symmetric) normalized Laplacian matrix.

1. Introduction

Quantum computing has exhibited potential acceleration advantages over classical computing by exploiting the unique properties of superposition and entanglement in quantum mechanics in solving certain problems, such as factoring integers,\textsuperscript{[1]} unstructured database searching,\textsuperscript{[2]} solving equations,\textsuperscript{[3–5]} regression,\textsuperscript{[6,7]} dimensionality reduction,\textsuperscript{[8,9]} mean centering.\textsuperscript{[10]} Overviews on quantum algorithms can be seen in refs. [11, 12].

In the era of big data, graph learning\textsuperscript{[13]} has attracted considerable attention owing to its wide applications in data science, machine learning, and image processing, etc. When dealing with problems related to graph learning, such as graph network,\textsuperscript{[14,15]} image processing,\textsuperscript{[16,17]} reinforcement learning,\textsuperscript{[18]} solving eigenproblem of the Laplacian matrix of a fully connected weighted graph (called ELFWG problem) is often encountered. In general, the Laplacian matrix $L$ is given by the difference between the degree matrix $D$ and the weight matrix $W$ ($L = D - W$). For a fully connected weighted graph, $W$ is a dense matrix, thus $L$ is also a dense matrix. Solving the eigenproblem of a dense matrix is very challenging because it involves expensive matrix operations. Therefore, it is imperative for us to design an efficient algorithm to solve this problem.

Fuelled by the success of quantum algorithms, Kerenidis et al.\textsuperscript{[19]} proposed a quantum algorithm for solving the eigenproblem of the symmetric normalized Laplacian matrix, and successfully applied it to the spectral clustering algorithm. However, as the authors pointed out, their algorithm cannot efficiently access $D$ and the norm of the row vectors of $W$. This makes their algorithm cannot be directly used to solve ELFWG problem. Subsequently, Li et al.\textsuperscript{[20]} designed a quantum algorithm to solve the eigenproblem of $L$ and applied it to accelerate the spectral clustering algorithm. However, their quantum algorithm can only handle the case that $L$ is a sparse matrix. Huang et al. proposed a quantum Laplacian eigenmap algorithm, which has a significant speedup compared with its classical counterparts.\textsuperscript{[21]} However, this algorithm relies on a strong assumption that the classically stored information of $W$ and $D$ can be accessed in parallel by a quantum random access memory.\textsuperscript{[25]} In fact, in most scenarios,\textsuperscript{[14–17,23,24]} we only have information about the vertices of the graph, $W$ and $D$ can be obtained by complex computations. Ref. [21] hides the complexity of this part with the above strong assumption. If the complexity of this part is also taken into account, the advantages of quantum algorithm will be greatly weakened or even disappeared. In short, the existing quantum algorithm cannot efficiently solve ELFWG problem.

In this paper, we design an efficient quantum algorithm to solve ELFWG problem without the above strong assumptions. And our algorithm starts from the vertex set of the graph, which
makes our proposed algorithm more suitable for practical sce-
narios. Specifically, the optimal Hamiltonian simulation tech-
nique based on the block-encoding framework \cite{25-27} is used to
implement the quantum simulation of \( L \), which reduces the
algorithm’s dependence on simulation error. Then we employ the
quantum phase estimation algorithm \cite{28} to extract the eigenval-
ues and eigenvectors of \( L \). The key of the whole algorithm is to
construct block-encodings of operators containing the informa-
tion of \( W \) and \( D \) to further obtain the block-encoding of \( L \). It is
shown that our algorithm can achieve polynomial speedup on the
number of vertices and exponential speedup on the dimension
of each vertex. In particular, our algorithm can also be used to
solve eigenproblem of \( W \), which is also of great significance. \cite{29-32}
We also show that our algorithm can be extended to solve the
eigenproblem of symmetric (non-symmetric) normalized Lapla-
cian matrix.

The remainder of the paper is organized as follows. In Sec-
tion 2, we give a brief overview of the Laplacian matrix. In Sec-
tion 3, we propose a quantum algorithm to solve ELFWG prob-
lem. In Section 4, we generalized the algorithm to solve eigen-
problem of symmetric (non-symmetric) normalized Laplacian
matrix. In Section 5, we give some discussions about our algo-

2. Review of the Graph Laplacian Matrix

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

\[
w_{ij} = w_{ji} = \begin{cases} 
0, & i \neq j, \\
1, & i = j
\end{cases}
\]

(1)

Specifically, the element \( w_{ij} > 0 \) represents that vertex \( x_i \) is con-
nected to vertex \( x_j \), otherwise \( w_{ij} = 0 \). Here, we take the Gaussian
similarity function \( w_{ij} = \exp(-\lambda \|x_i - x_j\|^2), i \neq j \), which is widely
used as an example, where \( \lambda > 0 \) is any given real number. \cite{33, 34}
It is worth noting that we can also choose other forms of similar-
functions, \cite{22, 24} such as sigmoid similar function.

Next the degree matrix \( D \in \mathbb{R}^{n \times n} \) of the graph \( G \) is defined as
a diagonal matrix \( D = \text{diag}(d_i), \) where \( d_i = \sum_{j=1}^{n} w_{ij} \).

Given \( W \) and \( D \), the graph Laplacian matrix \( L \) is defined as

\[
L = D - W \in \mathbb{R}^{n \times n},
\]

(2)

where \( L \) is a symmetric positive semi-definite matrix. Addition-
ally, there are two types of normalized Laplacian matrices as fol-
lows:

\[
L_{n} = D^{-1} - 2 \quad L_{s} = D^{-1} - 2 \quad W D^{-1},
\]

(3)

where \( L_{n} \) (\( L_{s} \)) is a symmetric (non-symmetric) matrix. For more
information about the Laplacian matrix \( L \), see refs. \cite{23, 24}.

Solving ELFWG problem has wide applications in graph
networks, \cite{14, 15} image processing, \cite{16, 17} and reinforcement
learning, \cite{18} etc. For example, in ref. \cite{15}, the eigenvalues of \( L \)
are used to construct the spectral filter to solve semi-supervised
classification problem. In ref. \cite{18}, the eigenvectors of \( L \) of
the state-transition graph are used to represent the state embed-
ing, which captures the geometry of the underlying state space, and
is beneficial for reinforcement learning tasks such as option
discovery and reward shaping. For a fully connected weighted
graph, \( W \) is a dense matrix, thus \( L \) is also a dense matrix. The
rank of \( L \) is \( n - 1 \) and the eigenvector corresponding to the zero
eigenvalue of \( L \) is \( 1 = (1, \ldots, 1)^T \in \mathbb{R}^n \). \cite{24} It is worth noting that
for any \( L \), the vector \( 1 \) is the eigenvector corresponding to the
eigenvalue of \( 0 \), thus it does not provide any extra information.
Here, we consider extracting \( 1 \leq d \leq (n - 1) \) non-zero eigen-
values and the corresponding eigenvectors of \( L \), and its classical
complexity is \( O(mn^2 + dn^3) \). \cite{35} 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.

3. A Quantum Algorithm for Solving ELFWG Problem

To design an efficient quantum algorithm to solve ELFWG prob-
lem, we adopt the optimal Hamiltonian simulation technology
based on the block-encoding framework \cite{25-27} to realize the quan-
tum simulation of \( L \) and then perform quantum phase estimation
algorithm \cite{28} to extract the eigeninformation of \( L \). The core
of our entire algorithm is to construct a block-encoding of \( L \). To
achieve it, we first need to construct block-encodings of operators
containing the information of \( W \) and \( D \), respectively.

The entire section consists of fourth subroutines: In Sec-
tion 3.1, we review the optimal Hamiltonian simulation tech-
nology based on the block-encoding framework. In Section 3.2, we
design quantum algorithm to construct a block-encoding of \( L \). In
Section 3.3, we extract the eigeninformation of \( L \). In Section 3.4,
we analyze the complexity of our algorithm. For convenience, we
abbreviate the function \( \log_2 x \) as \( \log x \), that is, \( \log_x = \log x \) and
use \( 0^{(m)} \) to represent that the register has \( m \) qubits.

Assume that the vertex set \( V = \{x_i | x_i \in \mathbb{R}^n\} \) is stored in a
quantum random access memory data structure \cite{36, 37} that is,
the element \( x_i \) 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. There exists a quantum
algorithm that can perform the following map with \( \varepsilon \)-precision
in \( O(\log n/\varepsilon) \) time:

\[
U(i) |0 \rangle \rightarrow |i \rangle |x_i \rangle = |i \rangle \frac{1}{\|x_i\|} \sum_{j=1}^{m} x_i[j]
\]

(4)

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

\[
O(i) |0 \rangle \rightarrow |i \rangle \|x_i\|
\]

(5)

where the binary representation of \( \|x_i\| \) is \( \|x_1\| \cdots \|x_n\| \), \( h \) is
the number of qubits. Then \( \|x_i\| = \|x_i[1]\| \otimes \|x_i[2]\| \cdots \otimes
\|x_i[h]\|, i = 1, \ldots, n \). Without loss of generality, assume that the
value of \( \|x_i\| \) between 0 and 1 in the following sections. If this
is not the case, we can do preprocessing by dividing \( \|x_i\| \) by
\( \max_i \|x_i\| \) so that it lies between 0 and 1. This data structure has
also been successfully applied to quantum data compression, quantum linear systems with displacement structures, and so on.

3.1. Review of the Block-Encoding Framework

In this section, we review the optimal Hamiltonian simulation technology based on the block-encoding framework. We first give the definition of the block-encoding framework.

Definition 1 (Block-encoding). Assume that $A$ is an $s$-qubits operator, $\alpha, \epsilon, j \in \mathbb{R}^s$, and $a \in \mathbb{N}$, then we say that the $(s + a)$-qubits unitary $U$ is a $(\alpha, \epsilon, j)$ block-encoding of $A$ if it satisfies

$$\|A - \alpha(|0^a \otimes I|U(|0^a \otimes I))\| \leq \epsilon_A$$

(6)

Meanwhile, Low and Chuang proposed a block-encoding framework of density operator:

Lemma 2 (Block-encoding of density operator). Suppose that $\rho$ is an $s$-qubits density operator and $G$ is an $(a + s)$-qubits unitary operator that acts on the state $|0^a \otimes |0^s\rangle$ prepares a purification $|0^a \otimes |0^s\rangle \mapsto |\rho\rangle$, such that $\text{Tr}_a(|\rho\rangle\langle\rho|) = \rho$, $a = a_1 + a_2$. Then $(G^\dagger \otimes I_j)(|I_\alpha \otimes \text{SWAP}_{a_2})(G \otimes I_j)$ is an $(1, a + s, 0)$ block-encoding of $\rho$, where $\text{SWAP}_j$ is a unitary that swaps the register $a_2$ with the system register of size $s$.

Subsequently, András Gilyén et al. proposed to implement a block-encoding of a linear combination of block-encoded operators. It is shown as follows:

Definition 3 (State preparation pair). Let $\rho \in \mathbb{C}^m$ and $||\rho||_1 \leq \beta$. The pair of unitaries $(P_{b, c}, P_{\epsilon, j})$ is called an $(b, c)$-state-preparation-pair if $P_{b, c}(|0^a \otimes |0^s\rangle = \sum_{j=1}^{a+|s|} |c_j\rangle_\rho \otimes |j\rangle_\beta$ and $P_{\epsilon, j}(|0^a \otimes |0^s\rangle = \sum_{j=1}^{a+|s|} |\epsilon_j\rangle_\rho \otimes |j\rangle_\beta$ such that $\sum_{j=1}^{a+|s|} |\epsilon_j\rangle_\rho \otimes |j\rangle_\beta$ and for all $j \in \{m + 1, \ldots, 2^s\}$, we have $\epsilon_j \neq 0$.

Lemma 4 (Linear combination of block-encoding matrices).

Let $A = \sum_{j=1}^{a+|s|} |y_j\rangle \langle A_j|$ be an $s$-qubits operator and $\epsilon_j \in \mathbb{R}^s$. Assume that $(P_{b, c}, P_{\epsilon, j})$ is an $(b, c, \epsilon)$-state-preparation-pair for $\rho \in \mathbb{C}^m$, $W = \sum_{j=1}^{a+|s|} |y_j\rangle \langle U_j \otimes I_j^{2^s} |(I - \sum_{j=1}^{a+|s|} |y_j\rangle \langle y_j|) \otimes I_j^{2^s}$ is an $(a + s + \epsilon)$-qubits unitary operator such that for all $j = 1, \ldots, m$, we have $\rho_j = |y_j\rangle \langle y_j| \otimes I_j^{2^s}$. Then $U_j$ and $U_j = (a, \epsilon, j)$-block-encoding of $A_j$. Then we can implement an $(a\beta, a + b, \epsilon \alpha + \beta \epsilon, j)$-block-encoding of $A$, with a single use of $W$, $P_b$ and $P_{\epsilon, j}$.

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

Theorem 5 (Optimal block-Hamiltonian simulation). Assume that $U$ is an $(a, \epsilon, j)$-block-encoding of Hamiltonian $H$. Then we can implement an $\epsilon$-precise Hamiltonian simulation $V$ which is an $(1, a + 2, \epsilon)$ block-encoding of $\exp(iHt)$, with $O(|a|t) + \log(1/\epsilon)/\log(1/|\epsilon|)$ uses of controlled-$U$ or its inverse and with $O(|a|t) + \log(1/\epsilon)/\log(1/|\epsilon|)$ two-qubit gates.

The core of this technique is to construct a block-encoding of an operator. To realize the quantum simulation of $L$, next we will design quantum algorithms to construct a block-encodings of $L$.

3.2. Block-Encoding Construction of $L$

To construct a block-encoding of $L = D - W$, we first construct block encodings of operators containing the information of $W$ and $D$, respectively.

3.2.1. Constructing a Block-Encoding of an Operator Containing the Information of $W$

We know that the elements of $W$ are

$$w_j = \exp(-\lambda ||x_i - x_j||^2)$$

(7)

The term $\exp(-\lambda ||x_i - x_j||^2)$ is a simple scalar that depends on the size of $||x_i||$, $i = 1, \ldots, n$. Next we take the Taylor expansion for $\exp(2\lambda x_i \cdot x_j)$ to get

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

(8)

By keeping only the low-order terms of the Taylor expansion, we can get a finite-dimensional approximated of $\exp(2\lambda x_i \cdot x_j)$, that is,

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

(9)

Combine Equation (7) with Equation (9), we can obtain

$$w_j \approx \exp(-\lambda (||x_i||^2 + ||x_j||^2)) \sum_{k=0}^{p} a_k (x_i \cdot x_j)^k$$

(10)

where $a_k = (2\lambda)^k/k!$, $k = 0, 1, \ldots, p$. To facilitate the introduction of the algorithm, we first list the lemma required.

Lemma 6 (Quantum multiplier (QM)). Let integers $a, b$ be $h$-bit string and the binary representations of $a$ and $b$ are $a = a_1 a_2 \cdots a_h$ and $b = b_1 b_2 \cdots b_h$, respectively. Then there is a quantum algorithm with $O(poly(h))$ single- and double-qubit gates can realize

$$|a\rangle|b\rangle \mapsto |a\rangle|ab\rangle$$

(11)

where $|a\rangle = |a_1\rangle \otimes |a_2\rangle \otimes \cdots \otimes |a_h\rangle$ and $|b\rangle = |b_1\rangle \otimes |b_2\rangle \otimes \cdots \otimes |b_h\rangle$. When the accuracy of the algorithm is defined as $\epsilon_a = 2^{-h}$, the complexity of QM is given by $O(polylog(1/\epsilon_a))$.

The specific algorithm proceeds as following steps:

(1.1) Prepare the quantum state

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle, \frac{1}{\sqrt{a}} \sum_{k=0}^{a} \sum_{i=0}^{b} \sqrt{a_k} |k\rangle |0^{b}\rangle |0^{a}\rangle |0^{p}\rangle \sum_{i=0}^{\epsilon_a} |\epsilon_i\rangle$$

(12)

where $a = \sum i=0^{b} a_i$ and $1 = \sum_{i=0}^{\epsilon_a} \sqrt{a_i} |k\rangle$ can be prepared with the gate complexity of $O(polylog(p + 1)/\epsilon_a)$, where $\epsilon_a$ is the precision of the algorithm.
Perform the controlled unitary operator $R_O := \sum_{k=0}^{p-1}\ket{k}_2\bra{k}_2 \otimes (O^{\otimes 4} f^{(p-k)}_{(0,0)})_{1,3} \otimes O^{\otimes 2}_5 \otimes I_5$ for the system state, where the unitary operator $O$ can be seen in Equation (5).

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i \rangle \frac{1}{\sqrt{a}} \sum_{k=0}^{p} \sqrt{a_k} |k \rangle_2 |\|x_i\|_1\|^2 |\|x_i\|_2\|^{\otimes k} \|x_i\|_3^{\otimes 2} \otimes |y_i\rangle_5^{\otimes 2}$$

The quantum circuit of implementing $R_O$ is sketched in Figure 1.

Apply Lemma 6 to the second register and the fourth register, respectively, and undo the redundant registers, we have

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i \rangle \frac{1}{\sqrt{a}} \sum_{k=0}^{p} \sqrt{a_k} |k \rangle_2 |\|x_i\|_1\|^2 |\|x_i\|_2\|^{\otimes k} \otimes |y_i\rangle_5^{\otimes 2}$$

Perform the gate $\exp(-\lambda x)$ on the fourth register to generate

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i \rangle \frac{1}{\sqrt{a}} \sum_{k=0}^{p} \sqrt{a_k} |k \rangle_2 |\|x_i\|_1\|^2 |\|x_i\|_2\|^{\otimes k} \otimes \exp(-\lambda |x_i|_2)$$

Apply Lemma 6 for the third and fourth registers and uncompute redundant registers

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i \rangle \frac{1}{\sqrt{a}} \sum_{k=0}^{p} \sqrt{a_k} |k \rangle_2 |\|x_i\|_1\|^2 |\|x_i\|_2\|^{\otimes k} \otimes |y_i\rangle_5^{\otimes 2}$$

Add one qubit and rotate it from $|0\rangle$ to $[\exp(-\lambda |x_i|_2^2) |x_i|_2^2] |0\rangle + \sqrt{1 - \exp(-\lambda |x_i|_2^2) |x_i|_2^2} |1\rangle$ controlled on the third register. Then we uncompute the third register to obtain

$$\frac{1}{\sqrt{Y}} \sum_{i=1}^{n} |i \rangle \frac{1}{\sqrt{a}} \sum_{k=0}^{p} \sqrt{a_k} |k \rangle_2 |\|x_i\|_1\|^2 |\|x_i\|_2\|^{\otimes k} \otimes |y_i\rangle_5^{\otimes 2}$$

Apply the quantum amplitude amplification algorithm\cite{44} to produce

$$\frac{1}{\sqrt{Y}} \sum_{i=1}^{n} |i \rangle \frac{1}{\sqrt{a}} \sum_{k=0}^{p} \sqrt{a_k} |k \rangle_2 |\|x_i\|_1\|^2 |\|x_i\|_2\|^{\otimes k}$$

Perform the controlled unitary operator $R_U := \sum_{k=0}^{p} |k \rangle_2 \bra{k}_2 \otimes (I^{\otimes 4} f^{(p-k)}_{(0,0)})_{1,3} \otimes I_5 \otimes I_5$, where the unitary operator $U$ can be seen in Equation (4), the system state becomes

$$\frac{1}{\sqrt{Y}} \sum_{i=1}^{n} |i \rangle \frac{1}{\sqrt{a}} \sum_{k=0}^{p} \sqrt{a_k} \exp(-\lambda |x_i|_2^2) |x_i|_2^2 |\|x_i\|_2\|^{\otimes k} |y_i\rangle_5^{\otimes 2}$$

where

$$|\Psi(x_i)\rangle = \exp(-\lambda |x_i|_2^2) (\sqrt{a_0} |0\rangle^{\otimes p} + \ldots + \sqrt{a_p} |p\rangle^{\otimes p} |x_i\rangle^{\otimes 2})$$

Figure 1. The quantum circuit of the controlled unitary operator $R_O$. The subscripts 1, …, 5 represent the index of the registers. Each sub-register in third and fourth registers has $h$ qubits and each sub-register in the fifth register has $\log m$ qubits, and $H$ denotes Hadamard gate.
Figure 2. The quantum circuit of the controlled unitary operator $R_U$. The subscripts 1, 2, 5 represent the index of the registers. The fifth register has $p \log m$ qubits which comes from $p$ sub-registers with $\log m$ qubits and $H$ denotes Hadamard gate.

Figure 3. The quantum circuit that generates the quantum state $|\Psi\rangle$. Here the subscripts 0, 1, ..., 5 represent the index of the registers. $RO$ and $RU$ denote the controlled unitary operator $RO$ and $RU$, respectively. $QM$ denotes the quantum multiplier in Lemma 6. $R$ 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 (1.2)–(1.8) represent the sequential steps of algorithm.

The quantum circuit of implementing $R_U$ is shown in Figure 2. Note that

\[
\langle \Psi(x_i)|\Psi(x_j) \rangle = \exp(-\lambda(\|x_i\|^2 + \|x_j\|^2))|a_0 + a_1\|x_i\|\|x_j\|\langle x_i|x_j \rangle + \cdots + a_{p}(\|x_i\|\|x_j\|\langle x_i|x_j \rangle)^p \approx w_{ij}
\]

(21)

(1.9) Take partial trace for the second and fifth registers, we get

\[
\rho_1 := \text{Tr}(|\Psi\rangle\langle \Psi|)_{2,5} = \frac{1}{2^n} \sum_{i=1}^{2^n} \langle \Psi(x_i)|\Psi(x_i) \rangle |i_1\rangle_1 \langle j_1|
\]

\[
\approx (W + I)/\text{Tr}(W + I)
\]

(22)

According to Lemma 2, the process of generating the state $|\Psi\rangle$ can be regarded as an $(a_1 + s_1)$-qubits unitary operator $G_1$ which can implement $G_1|0\rangle^{\otimes a_1}|0\rangle^{\otimes s_1} \mapsto |\Psi\rangle$, such that, $\text{Tr}_{1,2,3}(\rho_1)_{i} = \rho_1$, where $a_1 = \log(p + 1) + (p + 1)\lambda + p\log m + 1$, $s_1 = \log n$. Then $V_1 := (G_1 \otimes I_{1,2,3,4,5})(G_1 \otimes I_{1,2,3,4,5})$ is an $(1, a_1 + s_1, \epsilon_1)$-block-encoding of $\rho_1$, where $\epsilon_1$ is the error that produces the operator $\rho_1$ and the value of $\epsilon_1$ can be seen in the complexity analysis of Section 3.4. The whole quantum circuit is shown in Figure 3. For ease of understanding, we present a quantum algorithm for the simple case $\|x\| = 1, i = 1, \ldots, n$ in Appendix C. And our quantum algorithm can be extended to other forms of similarity functions which can be found in Appendix E.
3.2.2. Constructing a Block-Encoding of an Operator Containing the Information of D

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

**Lemma 7** ([24]) Distance/inner products estimation of two vectors. Assume that the unitary operators $U|i\rangle \langle 0| = |i\rangle \langle x_i|$ and $V|j\rangle \langle 0| = |j\rangle \langle x_j|$ can be performed in time $T$, and the norms $\|x_i\|$ and $\|x_j\|$ are known. Then there is a quantum algorithm that compute

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

(23)

or

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

(24)

with probability at least $1 - 2\delta$ for any $\delta \in (0, 1/2]$ with complexity $O(\|x_i\| \|x_j\| T \log(1/\delta)/\epsilon)$, where $\epsilon$ is the error of $\|x_i - x_j\|^2$ or $\|x_i\|$.

The process of the quantum algorithm as following steps:

(2.1) Prepare the quantum state

$$|0\rangle_1 \frac{1}{n} \sum_{i=1}^{n} \left|i\right\rangle \langle 0| \langle 0| \cdot 0 \odot \log(n) \rangle_2 0 \odot \log(n) \rangle_6$$

(25)

(2.2) The distance estimation algorithm of Lemma 7 is applied to the second, third, and fourth registers, we get

$$|0\rangle_1 \frac{1}{n} \sum_{i=1}^{n} \left|i\right\rangle \langle 0| \langle 0| \cdot 0 \odot \log(n) \rangle_6$$

(26)

(2.3) Perform $\exp(-i\lambda x)$ gate (see the detailed analysis in Appendix B) for the fourth register to produce

$$|0\rangle_1 \frac{1}{n} \sum_{i=1}^{n} \left|i\right\rangle \langle 0| \langle 0| \cdot 0 \odot \log(n) \rangle_6$$

(27)

In fact, according to Equation (1), we have $w_{ij} = 0$ when $i = j$. However, for Equation (27), 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 Equation (27) as

$$\left|0\right\rangle_1 \left(\sqrt{\frac{n^2 - n}{n^2}} \cdot \frac{1}{\sqrt{n^2 - n}} \sum_{i,j=1, i \neq j}^{n} \left|i\right\rangle \langle j\left| w_{ij} \right| \right) + \sqrt{\frac{n}{n^2}} \cdot \frac{1}{\sqrt{n}} \sum_{i,j=1}^{n} \left|i\right\rangle \langle j\left| w_{ij} \right| \right)_2 \odot \log(n) \rangle_6$$

(28)

(2.4) Run the quantum amplitude amplification algorithm to generate

$$|0\rangle_1 \frac{1}{\sqrt{n^2 - n}} \sum_{i,j=1}^{n} \left|i\right\rangle \langle j\left| w_{ij} \right| \cdot 0 \odot \log(n) \rangle_6$$

(29)

(2.5) Perform Hadamard gate $H$ for the first register

$$\frac{1}{\sqrt{2}} (|0\rangle_1 + |1\rangle_1) \frac{1}{\sqrt{n^2 - n}} \sum_{i,j=1}^{n} \left|i\right\rangle \langle j\left| w_{ij} \right| \cdot 0 \odot \log(n) \rangle_6$$

(30)

(2.6) Apply the controlled unitary operator $|0\rangle_1 \odot I_1 \odot R_{\psi(\phi)} \odot I_1 \odot I_1 \odot I_1$ where $R_{\psi(\phi)}$ is a controlled rotation operator, and uncompute the fourth register, we get

$$\frac{1}{\sqrt{n}} \sum_{i,j=1}^{n} \left|i\right\rangle \langle j\left| \left(\sqrt{1 - w_{ij}^2} |1\rangle_1 + w_{ij} |0\rangle_1\right) + |1\rangle_1 \frac{1}{\sqrt{n^2 - 1}} \sum_{j=1}^{n} \left|j\right\rangle_1 |0\rangle_1 \cdot 0 \odot \log(n) \rangle_6$$

(31)

$$:= \frac{1}{\sqrt{n}} \sum_{i,j=1}^{n} \left|i\right\rangle \langle j\left| \left(\sqrt{1 - w_{ij}^2} |1\rangle_1 + w_{ij} |0\rangle_1\right) + |1\rangle_1 \frac{1}{\sqrt{n^2 - 1}} \sum_{j=1}^{n} \left|j\right\rangle_1 |0\rangle_1 \cdot 0 \odot \log(n) \rangle_6$$

(32)

Note that the inner products $\langle \phi, \psi \rangle$ is

$$\langle \phi, \psi \rangle = \frac{\sum_{i,j=1}^{n} w_{ij}}{n - 1} = \frac{d_{ij}}{n - 1}$$

(33)

(2.7) Apply the inner products estimation algorithm of Lemma 7, we have

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} \left|i\right\rangle \langle 0\left| \langle 0| \cdot 0 \odot \log(n) \rangle_6$$

(34)

(2.8) Attach a register with $\log(n)$ qubits, then perform a controlled rotation operator $R_{\psi(\phi)}$ qubits, and uncompute the first, third, fifth, and sixth registers to produce

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} \left|i\right\rangle \langle 0\left| \langle 0| \cdot 0 \odot \log(n) \rangle_6$$

(35)
Run the quantum amplitude amplification algorithm to get
\[ \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \sqrt{\delta} |i \rangle_2 |0^{\log mn} \rangle_0 \] (36)
where \( \delta = \sum_{i=1}^{n} \exp(-\lambda \| x_i - x_0 \|^2) \).

Apply CNOT gate to the zeroth register
\[ \frac{1}{\sqrt{2}} \sum_{i=1}^{n} \sqrt{\delta} |i \rangle_2 |i \rangle_0 := |\phi \rangle \] (37)

Take partial trace for the zeroth register to get
\[ \rho_2 := \text{Tr}_0(|\phi \rangle \langle \phi |) = \frac{1}{n} \sum_{i=1}^{n} d_{0i} |i \rangle_2 |i \rangle_2 = \frac{D}{\text{Tr}(D)} \] (38)

According to Lemma 2, the process of producing the state \(|\phi \rangle\) also can be regarded as an \((a_2 + s_2\)-qubits unitary operator \(G_2\) which can realize \(G_2|0^{\log n} \rangle |0^{\log n} \rangle \rightarrow |\phi \rangle\), such that,
\[ \text{Tr}_0(|\phi \rangle \langle \phi |) = \rho_2, \] where \(a_2 = 2(1 + \log n + \log (mn))\), \(s_2 = \log n\).
Thus \(V_2 := (G_2 \otimes I_{1-2})(I_0 \otimes \text{SWAP}_{1-2})(G_2 \otimes I_{1-2})\) is an \((1, a_2 + s_2, 2e_2)\)-block-encoding of \(\rho_2\), where \(e_2\) is the error which generates the state \(|\phi\rangle\), and \(e_2\) is analyzed in detail in the complexity analysis of the algorithm. The whole quantum circuit is shown in Figure 4.

Next, we construct the block-encoding of \(L\). According to \(L = D - W\), we have
\[ \mathcal{L} = \frac{L}{\text{Tr}(L)} = \frac{D}{\text{Tr}(D)} - \frac{W}{\text{Tr}(D)} \] (39)
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
\[ \mathcal{L} = \rho_2 - \frac{\text{Tr}(W + I)}{\text{Tr}(D)} \rho_1 + \frac{I}{\text{Tr}(I)} \frac{\text{Tr}(I)}{\text{Tr}(D)} \] (40)

For \(I/\text{Tr}(I)\), we can prepare the quantum state \(|\tau\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i \rangle_1 |i \rangle_2\) with \(O(\log n)\) Hardmark and CNOT gates. Thus, we have
\[ \frac{I}{\text{Tr}(I)} = \text{Tr}_2(|\tau \rangle \langle \tau |) = \frac{1}{n} \sum_{i=1}^{n} |i \rangle_1 |i \rangle_1 = \rho_3 \] (41)

According to Lemma 2, we also get an \((1, 2 \log n, 0)\)-block-encoding of \(\rho_3\), that is, \(V_3 := (G_2 \otimes I_{1-2})(I_0 \otimes \text{SWAP}_{1-2})(G_2 \otimes I_{1-2})\) where \(G_2|0^{\log n} \rangle |0^{\log n} \rangle \rightarrow |\tau \rangle\), such that, \(\text{Tr}_2(|\tau \rangle \langle \tau |) = \rho_3\).
Thus, we obtain
\[ \mathcal{L} = -c\rho_1 + \rho_2 + c\rho_3 \] (42)
where \(c = \text{Tr}(W + I)/\text{Tr}(D) = \text{Tr}(I)/\text{Tr}(D), 0 < c < 1\). This can be viewed as a linear combination of block-encoded operators.

By Definition 3, let \(y = (-c, 1, c)\) and \(\|y\|_1 \leq \beta = 3\). Let \(b = 2, e_j = d_j = \sqrt{\beta}, j = 1, 2, 3\). We can effectively construct an \((3, 2, e_2)\)-state-preparation-pair \((P_2, P_3)\) of \(y\) that satisfies the requirements of Definition 3. In addition, we construct an \((\log n + 1 + 2)\)-qubits unitary \(Q = \sum_{i=1}^{l} |j \rangle \langle j| \otimes V_j + \sum_{j=1}^{l} |j \rangle \langle j| \otimes I_{1-2} \otimes I_{1-2} \otimes \text{SWAP}_{1-2}\) such that for \(j = 1, 2, 3\), where \(l = \max(a_1 + s_1, a_2 + s_2, a_3)\), we have that \(V_j\) is an \((1, l, e_2)\)-block-encoding of \(\rho_j\) where \(e_2 = \min(e_1, e_2)\).

According to Lemma 4, we can implement unitary \(G\) which is an \((3, 1 + 2, e_2 + 3e_2)\)-block-encoding of \(\mathcal{L}\), with a single use of \(Q, P_1, P_3\). Then combined with Theorem 5, we can implement the quantum simulation of \(L\). Specifically, we can implement a
$\varepsilon$-precise the Hamiltonian simulation unitary operator which is an $(1, l + 4, \varepsilon)$-block-encoding of $\exp(-i\mathcal{L}t)$ with

$$O(3t + \log(1/\varepsilon)/\log\log(1/\varepsilon))$$

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

### 3.3. Extracting the Eigeninformation of $\mathcal{L}$

Next we use quantum phase estimation algorithm\textsuperscript{[28]} 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:

1. Perform Hadamard and CNOT gates on the initial states $|0^{\log n}_1\rangle|0^{\log n}_2\rangle|0^{\log n}_3\rangle$ to produce the quantum state

$$|\alpha\rangle = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} |\tilde{u}_j\rangle |\tilde{u}_j\rangle |\tilde{u}_j\rangle$$

According to refs. [9, 19], we can obtain

$$\frac{1}{\sqrt{n}} \sum_{j=1}^{n} |\tilde{u}_j\rangle |\tilde{u}_j\rangle = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} |u_j\rangle |u_j\rangle$$

2. Run the quantum phase estimation algorithm\textsuperscript{[28]} by simulating $\exp(-i\mathcal{L}t)$ to reveal the eigenvalues and eigenvectors of $\mathcal{L}$

$$\frac{1}{\sqrt{n}} \sum_{j=1}^{n} |u_j\rangle |u_j\rangle \langle u_j|$$

3. Use quantum algorithm for finding the minimum\textsuperscript{[48]} to reveal the $d$ minimized nonzero eigenvalues $\gamma_j$, and we also can obtain the sample of the corresponding eigenvectors in the form quantum states $|u_j\rangle \propto u_j$ (the exact eigenvector), $j = 1, \ldots, d$.

Therefore, we can exact the eigeninformation of $\mathcal{L}$.

### 3.4. The Complexity of the Algorithm

The complexity of our algorithm mainly comes from constructing the block-encoding of $\mathcal{L}$ and extracting the eigeninformation of $\mathcal{L}$. Next, we will analyze their complexity in detail. Due to the complexity of constructing the block-encoding of $\mathcal{L}$ mainly comes from the algorithms for constructing block-encodings of $\rho_2$, we first analyze their complexity. For convenience, we use $\varepsilon$ to represent the actual state produced by our quantum algorithm containing errors in the following sections.

#### 3.4.1. The Complexity of Constructing the Block-Encoding of $\rho_2$

We first define three types of $w_j$: a) $w_j$: the exact value of $w_j$; b) $\tilde{w}_j$: the low-order approximate value of $w_j$ obtained by the Taylor expansion; and c) $\hat{w}_j$: the value of $w_j$ obtained by our quantum algorithm. According to Taylor’s theorem with Lagrange remainder\textsuperscript{[39]} we can obtain

$$w_j = \exp(-\lambda |\mathbf{x}_j - \mathbf{x}_j|^{2}) = \exp(-\lambda (|\mathbf{x}_j|^{2} + |\mathbf{x}_j|^{2})) \cdot \exp(2\lambda \mathbf{x}_j \cdot \mathbf{x}_j)$$

$$= \exp(-\lambda (|\mathbf{x}_j|^{2} + |\mathbf{x}_j|^{2})) \cdot \left[ \sum_{k=0}^{p} \left( \frac{z_j}{k!} \right)^{k} + R_{p+1}(x) \right]$$

where $z_j = 2\lambda \mathbf{x}_j \cdot \mathbf{x}_j$, $R_{p+1}(x) = \frac{\exp(\frac{\varepsilon x}{p})}{(p+1)!}$ is the reminder, $\xi \in (0, z_j)$. The error $\varepsilon_T$ between $\hat{w}_j$ and $\tilde{w}_j$ can be bounded with

$$\varepsilon_T = |\hat{w}_j - \tilde{w}_j| = |\exp(-\lambda (|\mathbf{x}_j|^{2} + |\mathbf{x}_j|^{2})) \cdot R_{p+1}(x)|$$

$$\leq |\exp(-\lambda (|\mathbf{x}_j|^{2} + |\mathbf{x}_j|^{2})) \cdot |R_{p+1}(x)|$$

$$\leq \frac{e^{\varepsilon_1(z_j)}}{(p+1)!} \leq \frac{e^{\varepsilon_1(z_j)}}{(p+1)/(\varepsilon^{[p+1]})}$$

where the third inequality comes from $(p+1)! > (p+1)/(\varepsilon^{[p+1]})$. Then we can get

$$\frac{1}{\varepsilon_T} \geq \frac{(p+1)/(\varepsilon^{[p+1]})}{e^{\varepsilon_1(z_j)}}$$

Taking the logarithmic function with the base of 2 on both sides of the above inequality and let $(p+1)/(\varepsilon^{[p+1]}) = 2$, we have $\varepsilon_T = O(2^{[p+1]})$. Next we consider the error $\varepsilon_Q$ between $\tilde{w}_j$ and $\hat{w}_j$ caused by our quantum algorithm:

In step (1.1), the complexity is $O(\log \log((p+1)/\varepsilon_3))$, which is derived from $\log n$ Hadamard gates generating the first register and $O(\log \log((p+1)/\varepsilon_3))$ with gate complexity generating the second register. The error of step (1.1) is $\varepsilon_3$.

In step (1.2), the circuit of the controlled unitary operator $R_{\mathcal{Q}}$ is uniformly controlled by $|\log(p+1)\rangle$ quibits in the second register. Ref. [60] shows that an implementation of any $n$-qubit gate of uniformly controlled one-qubit gates using at most $2^{n-1} - 1$ controlled-NOT gates, $2^{n-1}$ one-qubit gates, and a single diagonal $n$-qubit gate. Ref. [61] shows that an implementation of an arbitrary diagonal $n$-qubit unitary gate involving $2^n - 2$ controlled-NOT gates and $2^n$ elementary one qubit gates. Thus, any $n$-qubit gate of uniformly controlled one-qubit gates using $2^{n-1} - 1 + 2^n - 2 + 2^{n-1} + 2^n = O(2^n)$ single- and double-qubit quantum gate. Here, $n = \log(p+1)$, according to refs. [63, 64], the complexity of decomposing $R_{\mathcal{Q}}$ into a single- and double-qubit quantum gate is $O(p)$. And $\tilde{R}_{\mathcal{Q}}$ need to call $O(p^2)$ times of the unitary operator $\tilde{Q}$ with gate complexity $O(\log(mn))$. Thus, the complexity of $\tilde{R}_{\mathcal{Q}}$ is $O(p^2\log(mn))$.

In steps (1.3–1.5), the complexity mainly comes from $Q\mathcal{M}$ and $\exp(-\lambda x)$ quantum gates. Here, assume that $O(\log h)$ qubits is sufficient to accurately encode $|x|^{k}$, $k = 1, \ldots, p$, then the complexity of each execution of $Q\mathcal{M}$ is $O(\log(h))$ according to Lemma 6. When the accuracy of the algorithm is defined as $\varepsilon_h = 2^{-h}$, the complexity of $Q\mathcal{M}$ is given by $O(\log(1/\varepsilon_h))$. We
need to perform $O(p^3)$ times of QM to obtain Equation (14), so the complexity is $O(p^3 \text{poly} (1/e_h))$. The error caused by QM is

$$
\frac{1}{\sqrt{d}} \sum_{k=0}^{p} \sqrt{a_k} |k\rangle_2 \langle |k\rangle_2^i| = \frac{1}{\sqrt{d}} \sum_{k=0}^{p} \sqrt{a_k} |k\rangle_2 \langle |k\rangle_2^i| \\
\leq \frac{1}{\sqrt{d}} \sum_{k=0}^{p} \sqrt{a_k} |k\rangle_2 \langle |k\rangle_2^i| - |||\xi_i\rangle_2^i|| \\
\leq p \sum_{k=0}^{p} |||\xi_i\rangle_2^i|| - |||\xi_i\rangle_2^i|| \\
\leq p \sum_{k=0}^{p} \varepsilon_i = p^2 \varepsilon_i
$$

(51)

When using $h$ qubits to encode $||\xi_i||^2$ and $||\xi_i||^2$ and the first inequality follows from Lagrangian median theorem of $\exp(-\lambda x)$ which can be seen in Appendix B. And the error is

$$
\exp(-\lambda ||\xi_i||^2) - \exp(-\lambda ||\xi_i||^2) \leq \lambda \exp(-\lambda \xi_j) (||\xi_i||^2 - ||\xi_i||^2) \\
\leq \lambda ||\xi_i||^2 - ||\xi_i||^2 = \lambda \varepsilon_i
$$

(52)

where $\xi$ between $||\xi_i||^2$ and $||\xi_i||^2$ and the first inequality follows from Lagrangian median theorem of $\exp(-\lambda x)$. So the complexity is $O(p^3 \text{poly} (1/e_h))$. The total error of steps (1.3)–(1.5) is

$$
\frac{1}{\sqrt{d}} \sum_{k=0}^{p} \sqrt{a_k} |k\rangle_2 \langle |k\rangle_2^i| + \frac{1}{\sqrt{d}} \sum_{k=0}^{p} \sqrt{a_k} |k\rangle_2 \langle |k\rangle_2^i| \\
\leq \frac{1}{\sqrt{d}} \sum_{k=0}^{p} \sqrt{a_k} |k\rangle_2 \langle |k\rangle_2^i| - |||\xi_i\rangle_2^i|| \\
\leq p^2 \varepsilon_i + 2 \varepsilon_i
$$

(53)

Table 1. The complexity and the error of each step of the quantum algorithm.

| Steps | Complexity | Error |
|-------|------------|-------|
| (1.1) | $O[\text{poly} \log (\rho + 1/e_h)]$ | $\varepsilon_{a}$ |
| (1.2) | $O[p^3 \text{poly} \log (mn)]$ | $-$ |
| (1.3-1.5) | $O[p^3 \text{poly} \log (1/e_h)]$ | $(\lambda + p^2) e_h$ |
| (1.6) | $O(h)$ | $-$ |
| (1.7) | $O(\sqrt{a})$ | $-$ |
| (1.8) | $O[p^3 \text{poly} \log (mn/e_h)]$ | $2a(\sqrt{a}(\lambda + p^2) e_h + e_a + p^2 \varepsilon_i)$ |

Thus, we need perform $O(\sqrt{a})$ repetitions of quantum amplitude amplification algorithm to obtain the quantum state of Equation (18). Besides, we can obtain

$$
\sum_{i=0}^{n} a_i \exp(-\lambda ||\xi_i||^2) ||\xi_i||^{2k}/n = O(1)
$$

(56)

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

In step (1.8), the complexity of the uniformly controlled unitary operator $R_{ij}$ is $O(p^3 \text{poly} \log (mn/e_h))$, and its analysis is similar to $R_{ij}$. Next, we analyze the error $\varepsilon_{ij}$ is shown as follows:

$$
\exp(-\varepsilon i) = ||\hat{u}_{ij} - \hat{u}_{ij}|| = ||(\hat{\Psi}(\xi_i)|\hat{\Psi}(\xi_i)) - (\hat{\Psi}(\xi_i)|\hat{\Psi}(\xi_i))|| \\
\leq 2a(\sqrt{a}(\lambda + p^2) e_h + e_a + p^2 \varepsilon_i)
$$

(57)

where a detailed analysis of the above inequality can be seen in Appendix A. The complexity and error of each step of the quantum algorithm is shown in Table 1. The equation above is established by the Taylor expansion of order $p$ of $[\exp(\lambda ||\xi_i||^2)]^2$, that is,

$$
[\exp(\lambda ||\xi_i||^2)]^2 = 2 \exp(2\lambda ||\xi_i||^2) \sum_{k=0}^{p} a_k ||\xi_i||^{2k}
$$

(55)

$\frac{\sum_{i=0}^{n} a_i \exp(-\lambda ||\xi_i||^2) ||\xi_i||^{2k}}{na} = \frac{1}{a}

(54)

Putting this all together, the complexity of constructing the block-encoding of $\rho_i$ is

$$
O(\sqrt{ap^3 \text{poly} \log mn^2/(\varepsilon_a e_h e_f)}).
$$

(59)
To make the error that generates the operator $ρ_j$ is $ε_j$, and let $ε_x = ε_a = ε_y = ε_z$, we can obtain $1/ε_a = O(a^{1/2}(l + p^2)/ε_1)$. Thus, the complexity is

$$O(\sqrt{a} \, \text{poly} \, \log(mn^p/\varepsilon_1)) := O(\varepsilon_1)$$  (60)

3.4.2. The Complexity of Constructing the Block-Encoding of $ρ_2$

In step (2.1), we perform $2 \log n$ Hadamard gates for the second and third registers of the initial quantum state $|0\rangle_0 |0\rangle_1 |0\rangle_2 |0\rangle_3$ to produce the quantum state of Equation (25). Thus, the complexity of step (2.1) is $2 \log n$.

In step (2.2), according to Lemma 7, the complexity is

$$O((\max(\|x_i\|)) \, \text{poly} \, \log(mn/\varepsilon_d) \, \log(1/\delta_i))/\varepsilon_d)$$  (61)

where $1 - 2\delta_i$ is the probability of success of the algorithm with any $\delta_i \in (0, 1/2]$ and $\varepsilon_d$ is the error of $|x_i - x_i\rangle$.

In step (2.3), the complexity of the exp($-λ\delta_j$) gate is $O(\text{poly} \, \log(1/\delta_j))$ [with details given in Appendix B] which is smaller than the error $O(1/\varepsilon_d)$ caused by the step (2.2). Therefore we can ignore the complexity of these gates. The error of step (2.3) is

$$|\exp(-λ\delta_j) - \exp(-λ\delta_j)| \leq |λ \exp(-λ\delta_j)\delta_j - \delta_j| \leq λ\delta_j \leq λ\varepsilon_d$$  (62)

where the second inequality comes from Lagrange’s mean value theorem of $\exp(λ\delta_j)$,[14] $ξ_j$ takes value from $δ_j$ to $δ_j$.

In step (2.4), the probability amplitude of the states of Equation (28) is $p = (p^2 - n)/n^2 = O(1)$. Thus, we perform $O(1)$ times of quantum amplitude amplification algorithm to produce the state in Equation (29). In steps (2.5–2.6), it contains one Hadamard gate and a controlled rotation operator $R_n$. Due to the fourth register has $\text{log}(mn)$ qubits, thus the complexity of $R_n$ is $O(\text{log}(mn))$.[16]

In step (2.7), we analyze the error in calculating $\langle φ|\psi\rangle$ is

$$|(\langle φ|\psi\rangle - \langle φ|\psi\rangle)| \leq \frac{1}{n} \sum_{j=1}^{n} |\exp(-λ\delta_j) - \exp(-λ\delta_j)|$$  (63)

According to Lemma 7, we can get that the complexity is

$$O(\text{poly} \, \log(mn/\varepsilon_d))/\varepsilon_d)$$  (64)

where $χ = (\max(\|x_i\|))$, $v = \text{log}(1/\delta_i) \, \text{log}(1/\delta_j)$, $1 - 2\delta_j$ is the probability of success of the algorithm with any $\delta_j \in (0, 1/2]$ and $λ\varepsilon_d$ is the error of $\langle φ|\psi\rangle$. Here, let $δ_j = δ_j = O(\text{poly} \, \text{log} \, n)$.

In steps (2.8–2.10), the complexity comes mainly from quantum amplitude amplification algorithm. The probability amplitude of the target state in Equation (35) is

$$p_0 = \frac{\sum_{j=1}^{n} \exp(-λ\|x_i - x_j\|)}{(n(n - 1))}$$

$$\geq \min_j |\exp(-λ\|x_i - x_j\|)| := r$$  (65)

Table 2. The complexity and the error of each step of the algorithm.

| Steps | Complexity | Error |
|-------|------------|-------|
| (2.1) | $2 \log n$ | $–$ |
| (2.2) | $O(\text{poly} \, \log(mn/\varepsilon_d))$ | $\varepsilon_d$ |
| (2.3) | $O(\text{poly} \, \log(1/\varepsilon_d))$ | $\varepsilon_d$ |
| (2.4) | $O(\text{log}(mn))$ | $–$ |
| (2.7) | $O(\text{poly} \, \log(mn/\varepsilon_d))/\varepsilon_d$ | $\varepsilon_d$ |
| (2.8–2.10) | $O(1/\sqrt{T})$ | $λ^2 \varepsilon_2^2/4r$ |

Thus, we need $O(1/\sqrt{T})$ applications of quantum amplitude amplification algorithm to generate the state in Equation (36). In addition, we can get $p_0 = Tr(D)|n(n - 1)|$, that is, $Tr(D) = n(n - 1)p_0$.

Finally, we analyze the error that produces the quantum state $|φ\rangle$ is

$$‖|φ\rangle - |φ\rangle‖_2^2 = \frac{1}{\sqrt{T}} \sum_{i=1}^{n} |\exp(-λ\delta_i) - \exp(-λ\delta_i)|^2$$

$$= \frac{1}{\sqrt{T}} \sum_{i=1}^{n} |1 - \exp(-λ\delta_i)|^2$$

$$= \frac{1}{\sqrt{T}} \sum_{i=1}^{n} (\exp(-λ\delta_i))^2$$

$$\leq \frac{1}{\sqrt{T}} \sum_{i=1}^{n} (1 - \exp(-λ\delta_i))^2$$

where the second equation comes from Equation (33), the third equation follows from the Lagrange’s mean value theorem of $\sqrt{x}$.[16] $ξ_5$ takes value from $\langle φ|\psi\rangle$ to $\langle φ|\psi\rangle$, and the last inequality holds by $r \geq n(n - 1)/r$. The complexity and error of each step of the algorithm are shown in Table 2.

To make the error of $|φ\rangle$ equal to $ε_2$, and letting $ε_x = ε_2$, we have

$$‖|φ\rangle - |φ\rangle‖_2 \leq λ \varepsilon_2/2\sqrt{T} := ε_2$$  (66)

Thus we can obtain $1/ε_2 = λ/2\sqrt{T} ε_2$. Putting all complexity together, we get

$$O\left(\frac{λ(\max(\|x_i\|))^2 \text{poly} \, \log(4mn)|/\sqrt{T} ε_2|)}{4r^2/2^2} \right) := O(ε_2)$$  (68)

Therefore, the complexity of the block-encoding of $L$ is $\max(c_1, c_2)$.

Second, we analyze the complexity of extracting the eigeninformation of $L$ in Section 3.3.

In step (3.1), the complexity is $O(\text{poly} \, \log \, n)$, which comes from Hadamard and CNOT gates. In step (3.2), according to ref. [28], it takes $t = O(1/ε)$ times of controlled-$\exp(-iLt)$ to perform phase
estimation ensures that the eigenvalues \( \gamma_j \) of \( \mathcal{L} \) being estimated within accuracy \( \epsilon \). Therefore, the complexity of the algorithm is

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

(69)

where \( c_1 \) and \( c_2 \) are shown in Equations (58) and (65), respectively. In step (3.3), 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 for find the minimum that output the minimum values with probability larger than 1/2 with the query complexity \( O(\sqrt{n}) \). 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

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

(70)

When \( a, b, c, L, \lambda, \sqrt{r} \) and max \( \|x\| \) are all \( O(1) \), and letting \( 1/e_1 = 1/e_2 = 1/\epsilon = O(\text{poly} \log n) \), \( h = O(\text{log}(mn)) \), our quantum algorithm takes time

\[
O(d \sqrt{n} \text{poly} \log (mn))
\]

(71)

It is worth noting that the output of our algorithm is the \( d \) minimized nonzero eigenvalues \( \gamma_j \), and the corresponding eigenvectors in the form quantum states \( |u_i\rangle \) or \( u_i \) (the exact eigenvector), \( j = 1, \ldots, d \). This is the same output as some quantum algorithms, such as quantum principal component analysis,\(^{[8]} \) quantum linear discriminant analysis,\(^{[44]} \) and our algorithm achieve a polynomial speedup on \( n \) and an exponential speedup on \( m \) compared with the classical algorithm whose complexity is \( O(mn^2 + dn^3) \).

At present, using classical shadow technique,\(^{[63]} \) the information of \( |u_i\rangle \) can be efficiently represented and stored by a classical computer. The classical shadow information has the ability in providing many significant properties of \( |u_i\rangle \), such as amplitude distribution, linear observable information and entropy computation. And this technique only takes order \( O(\log K) \) measurements suffice to accurately predict \( K \) different functions of the state with high success probability. Therefore, the proposed quantum algorithm can provide similar outputs to classical algorithms.

In addition, our algorithm can also extract the eigeninformation of \( W \), which is of great significance.\(^{[29–32]} \) See the detailed analysis in Appendix D.

### 4. 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 Equation (3). We assume that the eigendecomposition form of \( L_s \) is

\[
L_s = \sum_{i=1}^{n} \mu_i |v_i\rangle \langle v_i|
\]

(72)

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 \( L_r^{[23]} \) the eigenvalues of \( L_s \) are also \( \{\mu_i\}_{i=1}^{n} \) and the corresponding eigenvectors are \( \{D^{-\frac{1}{2}}|v_i\rangle\}_{i=1}^{n} \). In particular, \( \mu_i = 0 \) is the unique zero eigenvalue of \( L_r \), and the corresponding eigenvector is \( |v_0\rangle = D^{\frac{1}{2}}1 \). Therefore, when we obtain the eigeninformation of \( L_s \), we can also get the eigeninformation of \( L_r \). Next we show that how to extract the eigeninformation of \( L_r \).

To achieve this, our core task is to first realize the quantum simulation of \( L_r \). According to Equation (3), we have

\[
L_r = \left( \frac{D}{\text{Tr}(D)} \right)^{-\frac{1}{2}} \left[ \frac{L}{\text{Tr}(L)} \right] \left( \frac{D}{\text{Tr}(D)} \right)^{-\frac{1}{2}} = (\rho_2)^{-\frac{1}{2}} L (\rho_2)^{-\frac{1}{2}}
\]

(73)

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

We have constructed the block-encodings of \( \rho_2 \) and \( L \), respectively. Based on Lemma 4 and Lemma 8 of ref. [27], we can design the block-encoding of \( L_r \):

**Lemma 8** (Block-encoding of \( A^{-\frac{1}{2}}BA^{-\frac{1}{2}} \)). Let \( c \in (0, \infty) \), \( \zeta_1 \in (0, 1/2) \), and let \( A \) and \( B \) are Hermitian matrices, and \( A \) satisfy \( 1/k \leq A \leq I \) where \( \kappa \geq 2 \). Let \( \zeta = \sqrt{c^{-1} \max(1, c) \log(\log(\rho_2) + 1)} \).

\( U \) is a \( (a_1, b_1, \zeta_1) \)-block-encoding of \( A \) that can be implemented using \( T_{u_1} \) elementary gates and \( V \) is an \( (a_2, b_2, \zeta_2) \)-block-encoding of \( B \) that can be implemented using \( T_{u_2} \) elementary gates. Then we can implement a unitary operator \( F \) that is an \( (4x^2a_1, 2b_1 + b_2 + O(\log(\kappa^2 \max(1, c) \log(\log(\rho_2) + 1)) \zeta)), 4x^2a_2, 4x^2a_2 + 4x^2b_2 \)-block-encoding of \( A^{-\frac{1}{2}}BA^{-\frac{1}{2}} \) in cost

\[
O\left\{ \begin{array}{l}
\max\{1, c\} \left[ 2a_1 \kappa \log \left( \frac{\kappa^2}{\zeta_1} \right) \right] (a_1 + T_{u_1}) \\
+ \kappa \log^2 \left( \max\{1, c\} \kappa \max(1, c) \zeta_1 / \zeta_2 \right) + T_{v_2} \end{array} \right\}
\]

(74)

According to Equation (70), we get \( c = 1/2, A = \rho_2, L = \mathcal{L} \). And we assume that \( \rho_2 \) and \( L \) satisfy the conditions of Lemma 8. We can get the unitary operator \( V \) which is an \( (1, a_2, s_2, 2e_2) \)-block-encoding of \( \rho_2 \) and the unitary operator \( G \) which is an \( (3, l + 2, e_2 + 3e_3 \) \)-block-encoding of \( L \), respectively.

Therefore, we can implement a unitary \( F \) that is an \( (12x^2, 2(a_2 + s_2 + 1) + l + O(\log(\kappa^2 \log(\zeta_2^2 \zeta_1^2))) ) \)-block-encoding of \( L_r \) in cost

\[
O\left[ 2x \log \left( \frac{\kappa^2}{\zeta_1} \right) (a_2 + s_2 + c_2) + \kappa \log^2 \left( \frac{\kappa^2}{\zeta_1} \right) + T_L \right]
\]

(75)

where \( e_2 = 12x^2 \zeta_1 + 4x(1 - 12e_3) \) and \( T_L = \max\{c_1, c_2\} \) is the complexity of produces the operator \( G \).

Finally, combining Theorem 5 and Section 3.3, we can obtain the the \( d \) minimized nonzero eigenvalues of \( L_s \) and the corresponding eigenvectors \( \{|v_i\rangle\}_{i=1}^{n} \). To obtain the eigeninformation of \( L_r \), we perform the quantum technique of Lemma 30 in ref. [27] to get the state \( \frac{1}{\|L^{-\frac{1}{2}}|v_i\rangle\|} \) for each \( |v_i\rangle, i = 1, \ldots, d \), that is, the corresponding eigenvectors \( \{D^{-\frac{1}{2}}|v_i\rangle\}_{i=1}^{n} \) of \( L_r \).
5. Discussion

In refs. [19, 21], scholars use the definition of $L = BB^T$ to design quantum algorithms, where $B$ is an incidence matrix to store the relationship between each node and its connected edges. As the introduction pointed out, their algorithm cannot efficiently solve ELFWG problem. A straightforward idea is to design quantum algorithm to efficiently realize the strong assumption of ref. [21], then obtain the eigeninformation of $L$ by using the existing technique of ref. [21]. However, the above strong assumption also require access to the norms of the column vector of $B$ and $D$, and an efficient quantum algorithm has not yet been found to implement it. In addition, ref. [21] uses the Hermitian chain product technique [44] to realize the quantum simulation of $L$. This makes the complexity of the algorithm have a cubic dependent on the inverse of the simulation error.

In our algorithm, we adopt the definition of $L = D - W$ to design the quantum algorithm, which is mainly caused by the following three reasons: 1) the strong assumption of ref. [21] is avoided; 2) the eigeninformation of $W$ can be extracted; 3) the eigenproblem of $L$ and $L_0$ can be solved. In addition, we adopt the optimal Hamiltonian simulation technique based on the block-encoding framework that can reduce the algorithm’s dependence on simulation error.

In addition, we find that the Gaussian kernel matrix $K$ satisfies $K = W + I$, thus our algorithm can also be used to solve the eigenproblem of $K$. Clearly, our algorithm is a quantum algorithm under the circuit model, while the quantum algorithms proposed in refs. [49–51] are formulated with the generalized coherent states, which is the specialized language of quantum optics. This makes them likely not universal quantum computing paradigm. [52] Compared to ref. [52], our algorithm can not only process the scenario where the modulo length of each sample data point $x$ is not equal to 1, that is, $||x|| \neq 1$, $i = 1, 2, 3, ..., n$, but also provide the optimal Hamiltonian simulation algorithm based on the block-encoding framework for $K$. Similarly, our algorithm can also be extended to solve other kernel matrix, which has a wide range of applications in classification, regression and so on.

6. Conclusion

In summary, we have designed an efficient quantum algorithm to solve ELFWG problem. Specifically, we have designed special controlled unitary operators to construct block-encodings of operators containing the information of $W$ and $D$ respectively to obtained 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 $W$, $L_0$, and $L$.

We expect that our quantum algorithm and the techniques mentioned, such as the quantum technique for constructing the block-encoding of an operator 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.

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 [53, 54] which can be implemented on noisy intermediate-scale quantum (NISQ) devices [55] to solve several problems related to $L$. In 2020, Slimane et al. proposed a variational Laplacian eigenmap algorithm [56], 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 [57] and their simulation results demonstrate that the proposed algorithm has good convergence. However, their algorithm employs the controlled SWAP test and maximum searching algorithm [58] to construct $W$, which cannot be implemented on NISQ devices. In addition, how to design a good strategy to suppress the barren plateau phenomenon [59] in variational quantum algorithms is also a thorny problem. Therefore, designing a variational quantum algorithm that can solve the above problems to solve eigenproblem of $L$ may be an important direction in the future work.

Appendix A: A Detailed Analysis of Equation (57)

Before we analyze Equation (57), we first give the lemmas as follows:

**Lemma 9** (Error propagation of quantum states). If $||x_i - y_i||_2 \leq \epsilon$, then $||a(x_i - b(y_i))||_2 \leq ||a - b|| ||x_i||_2 + ||b(x_i - y_i)||_2 \leq a - b + bx$, where $a, b$ are any positive real numbers.

**Proof:**

$$
||a(x_i - b(y_i))||_2 = ||a(x_i - b(x_i) + b(x_i - b(y_i))||_2 \\
\leq ||a - b|| ||x_i||_2 + ||b(x_i - y_i)||_2 \\
= (a - b) ||x_i||_2 + ||b|| ||x_i - y_i||_2 \leq a - b + bx
$$

**Lemma 10** (Error propagation of tensor products of quantum states). If $||x - y||_2 \leq \epsilon$, then $||x^{(p)} - y^{(p)}||_2 \leq p\epsilon$, where $p$ is any positive real numbers.

**Proof:** We prove it by mathematical induction. When $p = 2$, we have

$$
||x^{(p=2)} - y^{(p=2)}||_2 = ||x\otimes(x - y) + (x - y)\otimes y||_2 \\
\leq ||x||_2 ||x - y||_2 + ||x - y||_2 ||y||_2 \\
= ||x - y||_2 (||x||_2 + ||y||_2) \leq 2\epsilon
$$

Assume that when $k = p - 1$, we get $||x^{(p-1)} - y^{(p-1)}||_2 \leq (p - 1)\epsilon$. When $k = p$, we obtain

$$
||x^{(p)} - y^{(p)}||_2 \\
= ||x^{(p-1)}\otimes(x - y) + (x^{(p-1)} - y^{(p-1)})\otimes y||_2 \\
= ||x^{(p-1)}||_2 ||(x - y)||_2 + ||(x^{(p-1)} - y^{(p-1)})\otimes y||_2 \\
\leq \epsilon + (p - 1)\epsilon = p\epsilon
$$
Then we use the above lemmas to analyze Equation (57)

\[ |\hat{\omega}_j - \hat{\omega}_j| = |(\hat{\Psi}(x_i)|\hat{\Psi}(y_j)\rangle - (\hat{\Psi}(x_i)|\hat{\Psi}(y_j)\rangle)\]

\[ \leq ||(\hat{\Psi}(x_i)|\hat{\Psi}(y_j)\rangle + (\hat{\Psi}(x_i)|\hat{\Psi}(y_j)\rangle) - (\hat{\Psi}(x_i)|\hat{\Psi}(y_j)\rangle)\|
\]

\[ \leq ||\hat{\Psi}(x_i)\rangle - |\hat{\Psi}(x_i)\rangle\rangle_2
\]

where the second inequality comes from the Cauchy inequality of the vectors inner product.\(^{[24]}\) Then we analyze the size of \( ||\hat{\Psi}(x_i)\rangle - |\hat{\Psi}(x_i)\rangle\rangle_2 \) which is shown as follows

\[ ||\hat{\Psi}(x_i)\rangle - |\hat{\Psi}(x_i)\rangle\rangle_2 \]

\[ = \sum_{k=0}^{p} \sqrt{a_k} \exp(-\lambda ||\Psi(x_i)||^2) |k\rangle |0\rangle^{(|\Psi(x_i)||\Psi(x_i)||k\rangle}^{\otimes k}
\]

\[ \leq a^2 + \sqrt{\sum_{k=0}^{p} \sqrt{a_k} \exp(-\lambda ||\Psi(x_i)||^2) |k\rangle |0\rangle^{(|\Psi(x_i)||\Psi(x_i)||k\rangle}^{\otimes k}}
\]

\[ \leq a^2 + \sqrt{\sum_{k=0}^{p} \sqrt{a_k} \exp(-\lambda ||\Psi(x_i)||^2) |k\rangle |0\rangle^{(|\Psi(x_i)||\Psi(x_i)||k\rangle}^{\otimes k}}
\]

Similarly, we can also get \( ||\hat{\Psi}(x_i)\rangle - |\hat{\Psi}(x_i)\rangle\rangle_2 \leq a^2 + \sqrt{\sum_{k=0}^{p} \sqrt{a_k} \exp(-\lambda ||\Psi(x_i)||^2) |k\rangle |0\rangle^{(|\Psi(x_i)||\Psi(x_i)||k\rangle}^{\otimes k}} \)

\[ \times ||\hat{\Psi}(x_i)\rangle - |\hat{\Psi}(x_i)\rangle\rangle_2 \]

where the fourth inequality comes from Equation (21) and the fifth inequality comes from \( \lambda \leq 1 \) and \( a = a_1 + a_2 + \cdots + a_p \).

**Appendix B: The Circuit of \( f(x) = \exp(-\lambda x) \)**

Let \( x \in [0, 1) \) and the binary representations of \( x = x_1x_2 \cdots x_n \). The goal is to build an exponential function quantum gate computing the value of \( \exp(-\lambda x) \), that is, \( |x\rangle|0^\otimes\rangle|0^\otimes\rangle \rightarrow |x\rangle\exp(-\lambda x)|0^\otimes\rangle \).

The Taylor expansion of \( f(x) \) is shown as:

\[ \exp(\lambda x) = 1 - \lambda x + \frac{(\lambda x)^2}{2!} + \cdots \]

\[ \frac{(-1)^k(\lambda x)^k}{k!} + \frac{(-1)^{k+1}(\lambda x)^k}{(k+1)!}, \ z \in (0, x) \]

According to Taylor’s theorem,\(^{[19]}\) the \( (k + 1) \)th term in the expansion is \( \frac{(-1)^{k+1}(\lambda x)^{k+1}/(k+1)!} \) and the derivative of \( f(x) \) is bounded.

We can design the quantum circuit of \( f(x) \) by the quantum multiply-adder (QMA),\(^{[42]}\) which is shown in Figure B1. The complexity of \( f(x) = \exp(-\lambda x) \) gate is \( \text{poly}(n) \) single- and double-qubit gates. To put in another way, \( O(\text{polylog}(1/\epsilon)) \) one- or two-qubit gates can construct an exponential function quantum gate with accuracy \( \epsilon = O(2^{-n}) \). The specific analysis can be found in ref. [42].

**Appendix C: Constructing the Block-Encoding for \( W \) in the Simple Case**

For ease of understanding, we propose a quantum algorithm for constructing the block-encoding of an operator containing the information of \( W \) under the simple case \( ||x|| = 1, i = 1, \ldots, n \). According to Equation (10), when \( ||x|| = 1 \), we have \( \exp(-\lambda ||x||^2) = \exp(-\lambda x) = \exp(-\lambda x) \).

\[ \lambda \] is a constant that can be absorbed into \( a_n \), namely \( a_n = [\exp(-\lambda x)]^{(1/k)} = 1, k = 0, 1, \ldots, p \). Then we get \( a_n = \sum_{k=0}^{p} a_k x_k \) can be regarded as a polynomial function and its form is simpler. The specific steps of our quantum algorithm are as follows.

\[ (C.1) \text{ Prepare the quantum state} \]

\[ \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle \frac{1}{\sqrt{\beta}} \sum_{k=0}^{p} \sqrt{a_k} (|0^\otimes\rangle^k|0^\otimes\rangle^1|0^\otimes\rangle^1) \]

Here, assume that the vector \( \hat{a} = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} x_i \) is stored in a QRAM data structure.\(^{[36]}\) Then there is a quantum
Next, we can implement the block-encoding of $L$ in the case of $|x_i| = 1, i = 1, \ldots, n$. According to Equation (C2), we obtain $W = \tilde{a}(n\rho_0 - I)$. Due to $\text{Tr}(L) = \text{Tr}(D)$, we have

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle \frac{1}{\sqrt{2}} \sum_{x=0}^{2^n-1} \sqrt{a_i} |k_i\rangle (0^\otimes m_3 \otimes (0^\otimes p)_{-1}|x_i\rangle)^{\otimes k}$$

$$= \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle |\Phi(x_i)\rangle_{2,3} := |\Phi\rangle$$

(C2)

where

$$|\Phi(x_i)\rangle_{2,3} = \frac{1}{\sqrt{\tilde{a}}} \left( \sqrt{\tilde{a}_0} |0\rangle |0^\otimes m_3 \otimes (0^\otimes p)_{-1}|x_i\rangle + \cdots + \sqrt{\tilde{a}_n} |n\rangle |0^\otimes m_3 \otimes (0^\otimes p)_{-1}|x_i\rangle \right)$$

(C3)

Note that

$$\langle \Phi(x_i)| \Phi(x_j) \rangle = \left[ \tilde{a}_0 + \tilde{a}_1 (x_i|x_j) + \cdots + \tilde{a}_n (x_i|x_j)^n \right] / \tilde{a} \approx w_{ij} / \tilde{a}$$

(C4)

(C3) Take partial trace for the second and third registers, we have

$$\rho_0 := \text{Tr}(\Phi)(\Phi)_{2,3}$$

$$= \frac{1}{n} \sum_{i=1}^{n} (\Phi(x_i) | \Phi(x_i) \rangle | i \rangle \langle i |) = (W + \tilde{a} I)/n$$

(C5)

According to Lemma 2, we can design an $(\rho_0 + n\rho_0)$-qubits unitary operator $C_0$ which can implement $C_0 |0\rangle^{\otimes n} |0\rangle^{\otimes \rho_0} \mapsto |\Phi\rangle$, such that, $\text{Tr}_{2,3}(\Phi)(\Phi) = \rho_0$, where $\rho_0 = n \rho_0 + m \rho_0 + (p + 1) \rho_0 = n \rho_0$. Then $V_0 := (C_0^{\otimes \rho_0} \otimes I_1) (I_{2,3} \otimes \text{SWAP}_{1,2}) (C_0 ^{\otimes \rho_0} \otimes I_1)$ is an $(1, a_0 + a_0, c_0)$-block-encoding of $\rho_0$, where $c_0$ is the error that produces the operator $\rho_0$. The complexity analysis of this algorithm is similar to Section 3.4, thus it will not be analyzed in detail here.

Next, we can implement the block-encoding of $L$ in the case of $|x_i| = 1, i = 1, \ldots, n$. According to Equation (C2), we obtain $W = \tilde{a}(n\rho_0 - I)$. Due to $\text{Tr}(L) = \text{Tr}(D)$, we have

$$L = \frac{L}{\text{Tr}(L)} = \frac{D - W}{\text{Tr}(D)} = \frac{D}{\text{Tr}(D)} - \frac{n\tilde{a}}{\text{Tr}(D)} \rho_0 + \frac{I}{\text{Tr}(D)} \tilde{a} \text{Tr}(I) / \text{Tr}(D) := \rho_1 - d\rho_0 + e\rho_3$$

(C6)

where $d = n\tilde{a} / \text{Tr}(D), e = \tilde{a} \text{Tr}(I) / \text{Tr}(D)$. Similarly to Section 3.2, we can block-encode the eigenvalues of $L$.

Appendix D: Reveal the Eigeninformation of $W$

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

For the case of $|x_i| = 1$, we can obtain $W = \tilde{a}(n\rho_0 - I)$. Thus, we have

$$W/n = \tilde{a}(n\rho_0 - I/n) = \tilde{a}\rho_0 - \tilde{a}(\rho_1)$$

(D1)

For the case of $|x_i| \in (0, 1)$, we can obtain $W = \text{Tr}(W + I)\rho_1 - I$. Due to $\text{Tr}(W + I) = n$, Thus, we have

$$W/n = \rho_1 - \rho_2$$

(D2)

In short, Equations (C4) and (C5) can be viewed as a linear combination of block-encoded operators, respectively. Similarly to Section 3.2, we can construct the block-encoding of $W$ by applying the algorithm in Section 3.3.

Appendix E: Our Algorithm is Extended to Other forms of Edge Weights

We will show that our algorithm can be extended to the edge weight function $w_{ij}$ that satisfy the following conditions: The Taylor expansion of $w_{ij}$ can be expressed as

$$w_{ij} = f(\|x_i\|) \times f(\|x_j\|) \times g(x_i^T x_j)$$

(E1)

where $f(x)$ is a function that can be efficiently computed in quantum computer, $g(x) = a_0 + a_1x + \cdots + a_{p}x^p + \cdots$ is a polynomial function. Here, we also adopt the $p$-order Taylor approximation of $w_{ij}$. A detailed proof of the above condition can be implemented below:

1) Prepare the quantum state

$$\frac{1}{\sqrt{2}} \sum_{i=1}^{n} |i\rangle \frac{1}{\sqrt{2}} \sum_{x=0}^{2^n-1} \sqrt{a_i} |k_i\rangle |0^\otimes m_3 \otimes (0^\otimes p)_{-1}|x_i\rangle$$

(E2)
where \( a_k, k = 1, \ldots, p \) is the polynomial coefficient of \( g(x) \), and the state \( \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |\Psi(k)\rangle \) can be prepared with precision \( \epsilon_a \) in the gate complexity of \( O(\log(p + 1)/\epsilon_a^2) \).

2) Perform the controlled unitary operator \( R_0 := \sum_{k=0}^{p} |k\rangle_2 \otimes (O^{\otimes k} \otimes (p-k))_{1,3} \otimes O_{1,4} \otimes I_5 \), where the unitary operator \( O \) can be seen in Equation (5), we can get

\[
\frac{1}{\sqrt{N}} \sum_{m=1}^{n} |\varphi_{m1}\rangle \frac{1}{\sqrt{\alpha}} \sum_{k=0}^{p} \sqrt{a_k} |\varphi_{k}\rangle |\varphi_{m2}\rangle |\varphi_{m3}\rangle |\varphi_{m4}\rangle \prod_{i=1}^{k} O_{i}^{(p-k)} |\varphi_{m5}\rangle
\]

(E3)

3) Apply Lemma 6 to the third register and perform \( f(x) \) gate on the fourth register, and undo the redundant registers, we have

\[
\frac{1}{\sqrt{N}} \sum_{m=1}^{n} |\varphi_{m1}\rangle \frac{1}{\sqrt{\alpha}} \sum_{k=0}^{p} \sqrt{a_k} |\varphi_{k}\rangle |\varphi_{m2}\rangle |\varphi_{m3}\rangle |\varphi_{m4}\rangle \prod_{i=1}^{k} O_{i}^{(p-k)} |\varphi_{m5}\rangle
\]

(E4)

4) Use Lemma 6 for the third and fourth registers and uncompute redundant registers, we obtain

\[
\frac{1}{\sqrt{N}} \sum_{m=1}^{n} |\varphi_{m1}\rangle \frac{1}{\sqrt{\alpha}} \sum_{k=0}^{p} \sqrt{a_k} |\varphi_{k}\rangle |\varphi_{m2}\rangle |\varphi_{m3}\rangle |\varphi_{m4}\rangle \prod_{i=1}^{k} O_{i}^{(p-k)} |\varphi_{m5}\rangle
\]

(E5)

5) Add one qubit and perform the controlled rotation operator \( R \). Then we uncompute the third register to obtain

\[
\frac{1}{\sqrt{N}} \sum_{m=1}^{n} |\varphi_{m1}\rangle \frac{1}{\sqrt{\alpha}} \sum_{k=0}^{p} \sqrt{a_k} |\varphi_{k}\rangle \left| f \left( \left\langle \varphi_{m2} \right| \left| \varphi_{m3} \right| \right) \right|^2 \prod_{i=1}^{k} O_{i}^{(p-k)} |\varphi_{m5}\rangle
\]

(E6)

where \( \max = \max \left| f \left( \left\langle \varphi_{m2} \right| \left| \varphi_{m3} \right| \right) \right| \).

6) Apply the quantum amplitude amplification algorithm[43] to produce

\[
\frac{1}{\sqrt{N}} \sum_{m=1}^{n} |\varphi_{m1}\rangle \frac{1}{\sqrt{\alpha}} \sum_{k=0}^{p} \sqrt{a_k} \left| f \left( \left\langle \varphi_{m2} \right| \left| \varphi_{m3} \right| \right) \right|^2 \prod_{i=1}^{k} O_{i}^{(p-k)} |\varphi_{m5}\rangle
\]

(E7)

where \( \gamma = \sum_{m=1}^{n} \sum_{k=0}^{p} a_k \left| f \left( \left\langle \varphi_{m2} \right| \left| \varphi_{m3} \right| \right) \right|^2 \).

7) Perform the controlled unitary operator \( R_U := \sum_{k=0}^{p} |k\rangle_2 \otimes (O^{\otimes k} \otimes (p-k))_{1,3} \otimes O_{1,4} \otimes I_5 \), where the unitary operator \( U \) can be seen in Equation (4), the system becomes

\[
\frac{1}{\sqrt{N}} \sum_{m=1}^{n} |\varphi_{m1}\rangle \frac{1}{\sqrt{\alpha}} \sum_{k=0}^{p} \sqrt{a_k} \left| f \left( \left\langle \varphi_{m2} \right| \left| \varphi_{m3} \right| \right) \right|^2 \prod_{i=1}^{k} O_{i}^{(p-k)} |\varphi_{m5}\rangle
\]

(E8)

where

\[
|\Psi(x)\rangle = f (-\lambda \left\| x \right\| \left( \sqrt{a_0} \otimes O^{\otimes m} \right)^{\otimes p} + \ldots + \sqrt{a_p} \otimes O_{p}^{(p-k)} |\varphi_{m5}\rangle)
\]

(E9)

Note that

\[
\langle \Psi(x) | \Psi(x) \rangle = f \left( \left\langle \varphi_{m2} \right| \left| \varphi_{m3} \right| \right) \left( a_0 + \alpha_1 \left\| x \right\| \left\| \varphi_{m4} \right\| \left( \varphi_{m4} \right) \right) + \ldots
\]

(E10)

8) Take partial trace for the second and fifth registers, we get

\[
\rho := \text{Tr}(|\Psi\rangle \langle \Psi |)_{2,5} = \frac{1}{N} \sum_{m=1}^{n} \langle \Psi (x) | \Psi (x) \rangle |\varphi_{m1}\rangle |\varphi_{m2}\rangle |\varphi_{m3}\rangle |\varphi_{m4}\rangle
\]

(E11)

According to Lemma 2, the process of generating the state \( |\Psi\rangle \) can be regarded as an \( (p + 1) \)-qubits unitary operator \( G \) which can implement \( G(0^{\otimes a})0^{\otimes b} \rightarrow |\Psi\rangle \), such that \( T_{\frac{\alpha}{\lambda}} (|\varphi\rangle \langle \varphi |) = \rho \)

Here \( V := (G \otimes f_{1,3,4,0})(f_{1,2,5} \otimes \text{SWAP}_{1,3,4,0})(G \otimes f_{1,3,4,0}) \) is a \( (1, a + s, c) \)-block-encoding of \( \rho \), where \( c \) is the error that produces the operator \( T \).

Therefore, our quantum algorithm can be extended to the edge weight function \( w_j \) that satisfy the above conditions.

For example, when \( w_j \) is sigmoid similarity function:

\[
w_j = \tanh (x^T x + c) = \tanh c + (-\tanh^2 c + 1) x^T x
\]

(E12)

This above equation can be regarded as \( w_j = g(x^T x) \) which is a polynomial function, thus our algorithm can be extended to sigmoid similarity function. However, when \( w_j \) is exponential similarity function:

\[
w_j = \exp(-\lambda \left\| x - y \right\|)
\]

(E13)

Thus our algorithm cannot be extended to the case where \( w_j \) is exponential similarity function.

**Acknowledgements**

The authors thank Shaowei Ma for useful discussions on the subject. This work was supported by the Beijing Natural Science Foundation (Grant No. 4220301), the National Natural Science Foundation of China (Grant Nos. 61976024, 61972048, and 62006105).

**Conflict of Interest**

The authors declare no conflict of interest.

**Data Availability Statement**

The data that support the findings of this study are available from the corresponding author upon reasonable request.

**Keywords**

block-encoding, fully connected weighted graph, Laplacian matrix, quantum algorithm, solving eigenproblem

Received: February 1, 2023
Revised: March 28, 2023
Published online: April 28, 2023
