Quantum Algorithm for Vector Set Orthogonal Normalization and Matrix QR Decomposition with Polynomial Speedup

Zi-Ming Li[1] and Yu-xi Liu[1,2,*]

[1] *School of Integrated Circuits, Tsinghua University, Beijing 100084, China*
[2] *Frontier Science Center for Quantum Information, Beijing, China*

(Dated: August 15, 2025)

Vector set orthogonal normalization and matrix QR decomposition are fundamental problems in matrix analysis with important applications in many fields. We know that Gram-Schmidt process is a widely used method to solve these two problems. However, the existing methods, including Gram-Schmidt process have problems of high complexity, scaling $O(N^3)$ in the system dimension N, which leads to difficulties when calculating large-scale or ill-conditioned problems. With the development of quantum information processing, a series of quantum algorithms have been proposed, providing advantages and speedups over classical algorithms in many fields. In this paper, we propose quantum algorithms to solve these two problems based on the idea of Gram-Schmidt process and quantum phase estimation. The complexity of proposed quantum algorithms is theoretically and numerically analyzed. We find that our algorithms provide polynomial acceleration over the best-known quantum algorithms and a potential polynomial acceleration over the best-known classical algorithms on these two problems, scaling $O(N^2\mathrm{poly}(\log N))$ in the dimension N of the system, ignoring the complexity of state readout, or $O(N^2\mathrm{poly}(\log N))$ when considering the complexity of state readout.

I. INTRODUCTION

Vector set orthogonal normalization and matrix QR decomposition stand as fundamental problems in linear algebra and matrix analysis [1, 2]. They are also crucial for a myriad of applications in various scenarios, including scientific computing [3–5], machine learning and artificial intelligence [6–8], and many other engineering fields [9–11]. The goal of vector set orthogonal normalization is to transform sets of vectors into orthogonal normalized ones. Matrix QR decomposition aims at decomposing full-rank matrices into the product of orthogonal matrix Q and upper triangular matrix R. In the last few decades, many classical algorithms have been proposed and improved for solving these two problems [12–15] by using Gram-Schmidt orthogonalization process. However, these algorithms have problems of relatively high complexity, which scales $O(N^3)$ with the dimension N of the system. Such high computational complexity often brings challenges to managing large-scale or ill-conditioned matrices efficiently and accurately.

In the 1980s, Feynman proposed quantum computers, which use quantum physical systems to achieve information-storage, transmission, and processing [16]. With recent advancements in quantum computation, there has been a surge of theoretical and experimental studies in utilizing quantum algorithms to enhance the efficiency of classical computational tasks [17–22]. In particular, many quantum machine learning-based algorithms have emerged as powerful tools to solve complex linear algebra problems, offering exponential and polynomial speedups to traditional computing methods [23]. These quantum algorithms can be applied to solve linear equations [24, 25], support vector machine [26], principal component analysis [27], Bayesian network [28], quantum Boltzmann machine [29], and many other problems [23].

Recently, a quantum Gram-Schmidt algorithm was proposed for vector set orthogonal normalization [30] using QRAM model [31] with a query complexity of $O(r^{27}\kappa^{14r})$. The column vectors to be orthogonalized were lined into a matrix with the rank r and conditional number κ. The proposed algorithm achieves quantum speedup with a low-rank matrix with efficient state readout but reaches high complexity when r and κ are larger. Also, a quantum algorithm was proposed for QR decomposition of square matrices with $O(N^{2.5}\mathrm{poly}\log_2(N)/\epsilon^2)$ computational complexity [32], where N is the size of the matrix and ϵ is the desired precision. The scaling on the size of the matrix N successfully achieves polynomial speedup over classical algorithms but the speedup is limited, also the complexity of state readout is not considered.

Stimulated by previous studies [30–32], we here revisit and propose new quantum algorithms for vector set orthogonal normalization and matrix QR decomposition. We explain the details of the proposed algorithms and prove their correctness by theoretical derivations and numerical simulations. We use the QRAM model for efficient quantum initial state preparation [31, 33], which is a reasonable quantum oracle model widely used. We also analyze the complexity of our algorithm, including the number of quantum gates and the number of oracles used in the algorithms. The query complexity of our vector set orthogonal normalization algorithm scales $O(N^3\mathrm{poly}\log N)$, considering the state readout complexity, thus the proposed algorithm provides polynomial speedup over the previous result [30], which scales at least $O(N^{27})$ for the full rank matrix. The

* yuxiliu@mail.tsinghua.edu.cn

query complexity of our matrix QR decomposition algorithm $O(N^3 \mathrm{poly}\log N)$ in the system dimension N, which also provides polynomial speedup over previous result [32] whose scaling in system dimension N is $O(N^{3.5}\mathrm{poly}\log N)$ when the state readout complexity is taken into account. Thus, the complexity of the proposed algorithms is optimal to date.

The paper is organized as follows. In Sec. II, we provide a formal definition of the problems to be solved and briefly review the classical algorithms for vector set orthogonal normalization and QR decomposition problems. We also explain how Gram-Schmidt process can be used to solve these problems. For the completeness of the paper, in Sec. III, we summarize the main result of the quantum phase estimation algorithm, explain the oracle used in our paper, and give the details on the controlled Hamiltonian simulation step. In Sec. IV, we introduce our quantum algorithm for vector set orthogonal normalization problem and evaluate the performance of the proposed algorithm. In Sec. V, we introduce our quantum algorithm for QR decomposition and also evaluate the performance of the proposed algorithm. In Sec. VI, we apply our algorithms to several important problems, including linear regression, solving linear equations, and finding eigenvalues. The potential applications of our algorithms are also discussed. In Sec. VII, we summarize our results.

II. PROBLEM DEFINITION AND CLASSICAL ALGORITHMS

Formally, the problems of vector set orthogonal normalization and matrix QR decomposition of matrix are defined respectively as follows [1, 2].

Problem 1. Vector Set Orthogonal Normalization
Let S be a set containing M elements, each of the element is an N-dimensional vector, i.e. $S = \{a_1, a_2, \cdots, a_M\}$, $a_m \in \mathbb{C}^N, \forall m \in \{1, 2, \cdots, M\}$. Find a set of vectors $S' = \{u_1, u_2, \cdots, u_T\}$ satisfying:

1. $u_{t_1}^\dagger u_{t_2} = \delta_{t_1 t_2}, \forall t_1, t_2 = 1, 2, \cdots, T$
2. $span\{a_1, a_2, \cdots, a_M\} = span\{u_1, u_2, \cdots, u_T\}$

Problem 2. Matrix QR Decomposition
Let $A \in \mathbb{C}^{N\times M}$ be an arbitrary matrix with full rank satisfying $N \geq M$. Find an orthogonal matrix $Q \in \mathbb{C}^{N\times M}$ and an upper triangular matrix $R \in \mathbb{C}^{M\times M}$ satisfying:

1. $Q^\dagger Q = I_{M\times M}$
2. $A = QR$

There are many classical numerical methods for Problems 1 and 2. Problem 2 can be regarded as a generalization of Problem 1. The transformation matrix R is obtained at the same time when the column vectors of given matrix A is orthogonal normalized. One of the most common ways to solve Problems 1 and 2 is the Gram-Schmidt orthogonalization process. Gram-Schmidt process based algorithms for these two problems are summarized in following Algorithm 1 and Algorithm 2, respectively.

Algorithm 1: Gram-Schmidt Process for Vector Set Orthogonal Normalization [1, 2]

Input: $S = \{a_1, a_2, \cdots, a_M\}$, $a_m \in \mathbb{C}^N, \forall m = 1, 2, \cdots, M$
Output: $S' = \{u_1, u_2, \cdots, u_T\}$, satisfying $u_{t_1}^\dagger u_{t_2} = \delta_{t_1 t_2}, \forall t_1, t_2 = 1, 2, \cdots, T$, $span\{a_1, a_2, \cdots, a_M\} = span\{u_1, u_2, \cdots, u_T\}$

$v \leftarrow a_1 / \| a_1 \|$;
$S' \leftarrow \{v\}$;
for $m \leftarrow 2$ ***to*** M **do**
  $v \leftarrow a_m$;
  for u_t ***in*** S' **do**
    $v \leftarrow v - u_t^\dagger a_m u_t$;
  end
  if $\| v \| > 0$ **then**
    $S' \leftarrow S' \cup \{v / \| v \|\}$;
  end
end

Algorithm 2: Gram-Schmidt Process for Matrix QR Decomposition [1, 2]

Input: $A = (a_1, a_2, \cdots, a_M)$, $a_i \in \mathbb{C}^N$, $N \geq M$
Output: $Q = (q_1, q_2, \cdots, q_M)$, $R = [R_{m_1 m_2}]$ satisfying $QR = A$, $Q^\dagger Q = I_{N\times N}$, $R_{m_1 m_2} = 0, \forall m_1 > m_2$

$q_1 \leftarrow a_1 / \| a_1 \|$;
$R_{11} \leftarrow \| a_1 \|$;
for $m_1 = 2$ ***to*** M **do**
  $q_{m_1} \leftarrow a_{m_1}$;
  for $m_2 = 1$ ***to*** $m_1 - 1$ **do**
    $R_{m_2 m_1} \leftarrow q_{m_2}^\dagger a_{m_1}$;
    $q_{m_1} \leftarrow q_{m_1} - R_{m_2 m_1} q_{m_2}$;
  end
  $R_{m_1 m_1} \leftarrow \| q_{m_1} \|$;
  $q_{m_1} \leftarrow q_{m_1} / R_{m_1 m_1}$;
end

Hereafter $\| \cdot \|$ denotes 2-norm for matrices and vectors unless specified otherwise. In addition to the Gram-Schmidt orthogonalization process, many other classical numerical methods have also been proposed to solve these two problems, among which Householder transformation and Givens transformation are most commonly used. The complexity and numerical stability of these methods has been analyzed in detail over the last few decades [12–15].

It is noted that classical algorithms have problems of relatively high complexity, always scaling $O(N^3)$ in the system dimension N. For instance, for the QR decomposition of $N \times N$ matrices, the classical Gram-Schmidt process needs $2N^3$ floating point operations, while Householder transformation and Givens transformation need $8N^3/3$ floating point operations. Also, there

is loss of orthogonality in the final result due to the rounding errors in each computational step [13, 34, 35]. To reduce the error, one may need to perform the Gram-Schimdt procedure for many times, which all contributes to the computational complexity. To address the problem of high complexity of the classical algorithms, we develop quantum algorithms for Problems 1 and 2 and describe our algorithms in the following sections.

III. QUANTUM PHASE ESTIMATION

Our algorithm is based on quantum phase estimation (QPE), which mainly includes initial state preparation, controlled-U^{2^j} operations, and inverse quantum Fourier transform. In our paper, the initial state preparation is realized via the model of the quantum random-access memory (QRAM) [24, 26, 27]. The controlled-U^{2^j} operations are implemented via the Hamiltonian simulation with qubitization [36]. For the completeness of the paper, we here briefly summarize the main results of QPE, the QRAM model, and the qubitization, which are used in our algorithms.

QPE is a widely-used quantum algorithm based on quantum Fourier transformation and is performed as a subroutine in many other quantum algorithms [17, 22, 24]. Several quantum algorithms with QPE as a subroutine have been proved to have exponential acceleration over classical algorithms for many important problems such as discrete logarithm problem, hidden subgroup problem, large integer prime factorization problem, etc [22, 37, 38].

We assume that an N-dimensional unitary operator U has an eigenvalue $e^{2\pi i\phi_n}$ with unknown ϕ_n corresponding to eigenstate $|u_n\rangle$ for $n = 1, \cdots, N$. The goal of the QPE algorithm is to estimate each ϕ_n with the assistance of black box oracle for preparing an initial state $|u\rangle_s$ and performing the controlled-U^{2^j} operator [17]. Thus, the QPE procedure uses two registers. The first register containing j qubits is initialized in the ground state $|0\rangle_f^{\otimes j}$, and the second one is initialized in a state $|u\rangle_s$ in the N-dimensional space. The corresponding quantum circuit for QPE is given in Fig. 1.

The unitary operator U is usually realized via the time evolution operator of a given Hamiltonian H, which has spectral decomposition $H = \sum_{n=1}^{N} \lambda_n |u_n\rangle\langle u_n|$. That is, the unitary operator U can be expressed as

$$U = \sum_{n=1}^{N} \exp(-i\lambda_n t)\, |u_n\rangle\langle u_n|, \tag{1}$$

in which the phase factor $-\lambda_n t$ corresponds to $2\pi\phi_n$, i.e., $-\lambda_n t = 2\pi\phi_n$. Hereafter, we take $\hbar = 1$. The second register is initialized in an arbitrary quantum state $|u\rangle_s$ with

$$|u\rangle_s = \sum_{n=1}^{N} \langle u_n|u\rangle_s |u_n\rangle, \tag{2}$$

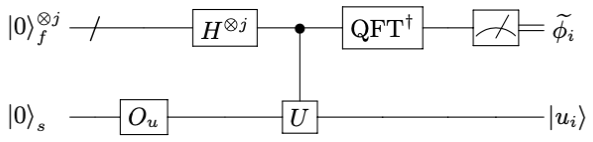

FIG. 1. Quantum circuit for QPE. O_u represents the preparation of an initial state $|u\rangle$. In our paper, it is implemented by the QRAM model. The controlled-U operator denotes a series of controlled-U^{2^j} operators $\{C - U^{2^{j-1}}, C - U^{2^{j-2}}, \cdots, C - U^{2^1}, C - U^{2^0}\}$. QFT† denotes inverse quantum Fourier transform. The control qubits are in the first register, labelled by the subscript f. The target qubits are in the second register, labelled by the subscript s. To avoid confusion, H represents Hadamard gate here in the circuit.

which is a linear combination of the eigenstates of U. The final state of the quantum circuit before measurements is

$$\sum_{n=1}^{N} \langle u_n|u\rangle_s \left|\frac{-\lambda_n t}{2\pi} 2^j\right\rangle |u_n\rangle, \tag{3}$$

i.e.,

$$\sum_{n=1}^{N} \langle u_n|u\rangle_s \left|\phi_n 2^j\right\rangle |u_n\rangle. \tag{4}$$

Hereafter $|\beta\rangle|\gamma\rangle$ denotes that the first (second) register is in the state $|\beta\rangle$ ($|\gamma\rangle$). It is noted that λ_n can be 0. Derivation of QPE is given in Appendix A when the qubit number of the first register is 1, which is always the case in our proposed algorithms.

A key component in QPE algorithm is to implement an oracle, which is used for preparing the initial state $|u\rangle_s$ and performing the controlled-U^{2^j} operator. Here, we assume that the state $|u\rangle$ preparation is realized via the QRAM model, which is used in many quantum algorithms [24, 26, 27]. The QRAM model for realizing O_u in Fig. 1 can prepare initial state $|u\rangle_s$ in $O(\log_2 N)$ time, i.e.,

$$|0\rangle_s \xrightarrow{\text{QRAM}} O_u|0\rangle_s = |u\rangle_s, \tag{5}$$

with the ground state $|0\rangle_s$ of the second register. The implementation of the controlled-U^{2^j} operators requires the Hamiltonian simulation, which is the original intention of Feynman's quantum computer [16]. In our algorithm, the Hamiltonian simulation is realized via the qubitization [36], which is considered as the best Hamiltonian simulation method [39] up to now. In qubitization, additional auxiliary qubits are introduced to the second register. The Hamiltonian H is accessed through two operators V and G, in which the operator G only acts on the states of auxiliary qubits in the second register, and the operator V acts on the states of both

auxiliary qubits and original qubits of the second register. G and V are chosen to satisfy the following condition,

$$\left(\langle 0|_a G^\dagger \otimes I_s\right) V \left(G|0\rangle_a \otimes I_s\right) \propto H, \tag{6}$$

with the ground state $|0\rangle_a$ of the auxiliary qubits. Here, the subscript a is for ancillary qubits introduced in the second register and s is for original qubits of the second register.

In the most common case and here, the Hamiltonian H is assumed to have a form of linear combination of unitaries $H = \sum_{l=1}^{d} \alpha_l V_l$. Then operators V and G can be chosen as

$$V = \sum_{l=1}^{d} |l\rangle_a\langle l|_a \otimes V_l, \tag{7}$$
$$G = \sum_{l=1}^{d} \sqrt{\frac{\alpha_l}{\sum_{l=1}^{d}|\alpha_l|}}|l\rangle_a\langle 0|_a + \cdots.$$

The greatest advantage of the qubitization is that the qubitization has low computational complexity. From Corollary 16 in [36], the query complexity for simulating the Hamiltonian H with form of linear combination of unitaries can be given in the following lemma.

Lemma 1 (the Linear-Combination-of-Unitaries (LCU) algorithm for Qubitization [36])**.** Given H is accessed via operators V and G as in Eq. (7), which specifies a Hamiltonian $H = \sum_{l=1}^{d} \alpha_l V_l$, the time evolution by H can be simulated for time t and error ϵ_0 with $O(\alpha t + \log_2(1/\epsilon_0))$ queries to V and G, where $\alpha = \sum_{l=1}^{d} |\alpha_l|$. The desired qubit number of the second register, including both original and auxiliary qubits, is $\lceil \log_2 N \rceil + \lceil \log_2 d \rceil + 2$ for simulating the Hamiltonian H, where N is the dimension of the Hamiltonian H. The number of additional two-qubit quantum gates needed is $O\left(\log_2 d\left(\alpha t + \log_2(1/\epsilon_0)\right)\right)$.

Thus, all parts of the QPE circuit used for our algorithms are explained. Below, we give our main algorithms based on the QPE circuit.

IV. QUANTUM GRAM-SCHMIDT ORTHOGONALIZATION ALGORITHM

In this section, based on QPE and the classical Gram-Schmidt orthogonalization algorithm, we propose quantum Gram-Schmidt orthogonalization algorithm to solve vector set orthogonal normalization problem as described in Problem 1.

A. Algorithm Description

Our algorithm comes from QPE algorithm. We assume that the unitary U as in Fig. 1 is the evolution operator of given Hamiltonian H, which has spectral decomposition

$$H = \sum_{n=1}^{k} \lambda_n |u_n\rangle\langle u_n| \tag{8}$$

with $k < N$ and $\lambda_n > 0$ for $\forall n = \{1, 2, \cdots, k\}$, i.e., the Hamiltonian has $N - k$ zero eigenvalues. Let us show how $\{|u_1\rangle, |u_2\rangle, \cdots, |u_k\rangle\}$ can be expanded to a set of orthogonal complete vectors $\{|u_1\rangle, |u_2\rangle, \cdots, |u_N\rangle\}$ via the QPE algorithm. We can write the input state $|u\rangle_s$ prepared by the QRAM model in the second register as the linear combination of these orthogonal complete vectors, i.e.,

$$|u\rangle_s = \sum_{n=1}^{N} b_n |u_n\rangle = \sum_{n=1}^{k} b_n |u_n\rangle + \sum_{n=k+1}^{N} b_n |u_n\rangle. \tag{9}$$

Then, the output state of the QPE algorithm is

$$\sum_{n=1}^{k} b_n |\tilde{\lambda_n}\rangle |u_n\rangle + \sum_{n=k+1}^{N} b_n |0\rangle |u_n\rangle, \tag{10}$$

which is after the inverse quantum Fourier transform and before the measurement. Here, $|\tilde{\lambda_n}\rangle$ and $|0\rangle$ denote the state of the first register, where $\tilde{\lambda_n}$ denotes an estimate value of the eigenvalues λ_n as in Eq. (8). $|u_n\rangle$ is the state of the second register.

We then make measurement in the first register for the state in Eq. (10) in the computational basis. If the outcome is 0, then the state of the second register will collapse into

$$|\psi\rangle = \frac{\sum_{n=k+1}^{N} b_n |u_n\rangle}{\| \sum_{n=k+1}^{N} b_n |u_n\rangle \|}. \tag{11}$$

It is clear that $|\psi\rangle$ is strictly orthogonal with $|u_n\rangle$ for $\forall n = \{1, 2, \cdots, k\}$ as

$$\langle u_n | u_{n'} \rangle = 0, \ \ \forall n' > k. \tag{12}$$

Thus, a quantum state $|\psi\rangle$, which is orthogonal with arbitrary state $|u_n\rangle$ with $n \leq k$, is constructed. After we obtain $|\psi\rangle$, the input state $|u\rangle$ in Eq. (9) can be represented by linear combinations of $|\psi\rangle$ and $\{|u_1\rangle, |u_2\rangle, \cdots, |u_k\rangle\}$. Then we can run the QPE algorithm again to find another state which is orthogonal to the states $|\psi\rangle$ and $\{|u_1\rangle, |u_2\rangle, \cdots, |u_k\rangle\}$. This process can be used to construct the bases in Gram-Schmidt orthogonalization procedure. When the first k bases $\{|u_1\rangle, |u_2\rangle, \cdots, |u_k\rangle\}$ are known, the $(k+1)$th base $|u_{k+1}\rangle$ can be defined as $|\psi\rangle$ as in Eq. (11). Thus, we construct the $(k+1)$th base by using k former constructed bases and the input state $|u\rangle$.

Based on the principle above, let us now study how to construct orthogonal normalized bases by using a set of N-dimensional vectors $S = \{a_1, a_2, \cdots, a_M\}$ as in Problem 1. We find that one qubit in the first

register is enough to realize our algorithm, thus the qubit number in the first register is always taken as one in the following description. In our algorithm, N components a_{nm} with $n = 1, \cdots, N$ of each N-dimensional vector a_m is encoded in a quantum state $|a_m\rangle$ via the computational basis $|n'\rangle \equiv \{0,1\}^{\otimes \log_2 N}$ of $\log_2 N$ qubits in the second register as

$$|a_m\rangle = \frac{1}{\| a_m \|} \sum_{n'=0}^{N-1} a_{n'm}|n'\rangle \equiv \frac{1}{\| a_m \|} \sum_{n=1}^{N} a_{nm}|n-1\rangle. \tag{13}$$

It is clear that $|n-1\rangle \equiv |n'\rangle$ for $n = 1, \cdots, N$. For the cases that N is not a power of 2, zeros are padded to the end of each a_m to enlarge the length from N to $2^{\lceil \log_2 N \rceil}$, so that we can use $\lceil \log_2 N \rceil$ qubits to encode each a_m, $\forall m = 1, 2, \cdots, M$. It is noted that the number of the padded zeros is no more than N, thus this padding won't affect the complexity of the algorithm.

For the case that $\{a_1, a_2, \cdots, a_M\}$ is a set of linearly independent vectors, orthogonal normalized bases $\{u_1, \cdots, u_M\}$ can be constructed as follows. We first set $|u_1\rangle \equiv |a_1\rangle$. Then a_1 is encoded by $|u_1\rangle$, which corresponds to a normalized base u_1. The nth component u_{n1} of the vector base u_1 is $u_{n1} \equiv \langle n-1|u_1\rangle$ with $n = 1, \cdots, N$. Based on $|u_1\rangle$, we can successively construct $|u_2\rangle, \cdots, |u_M\rangle$ with $\langle u_i|u_j\rangle = \delta_{ij}$ by using $|a_1\rangle, \cdots, |a_M\rangle$ as follows. Suppose $\{|u_1\rangle, |u_2\rangle, \cdots, |u_k\rangle\}$ has been constructed and encode the orthogonal normalized bases $\{u_1, \cdots, u_k\}$, with $u_{n_1}^\dagger u_{n_2} = \delta_{n_1 n_2}$, $\forall n_1, n_2 \leq k$ and $span\{u_1, \cdots, u_k\} \equiv span\{a_1, a_2, \cdots, a_k\}$. That is, k orthogonal normalized bases $\{u_1, \cdots, u_k\}$ have been constructed. Let us now construct the $(k+1)$th base u_{k+1}. Following the circuit of QPE, we assume that the input state $|u\rangle_s$ of the second register is the state $|a_{k+1}\rangle$, i.e., $|u\rangle_s = |a_{k+1}\rangle$, which encodes the N-dimensional vector a_{k+1} as shown in Eq. (13). The Hamiltonian H to obtain u_{k+1} can be chosen as $H = \sum_{n=1}^{k} |u_n\rangle\langle u_n|$, which can be simulated as shown in Eq. (6) and Eq. (7) by the qubitization. We assume that the system evolves with a time $t = \pi$, then the output state of the QPE circuit is

$$|1\rangle \left(\sum_{n=1}^{k} \langle u_n|u\rangle_s |u_n\rangle \right) + |0\rangle \left(|u\rangle_s - \sum_{n=1}^{k} \langle u_n|u\rangle_s |u_n\rangle \right), \tag{14}$$

with $|u\rangle_s \equiv |a_{k+1}\rangle$. We measure the first register. If the outcome is 0, we denote the state in the second register as

$$|u_{k+1}\rangle = \frac{|a_{k+1}\rangle - \sum_{n=1}^{k} \langle u_n|a_{k+1}\rangle |u_n\rangle}{\| |a_{k+1}\rangle - \sum_{n=1}^{k} \langle u_n|a_{k+1}\rangle |u_n\rangle \|}. \tag{15}$$

Thus, a state $|u_{k+1}\rangle$ is constructed and encodes an orthogonal normalized vector $u_{k+1} \in \mathbb{C}^N$ such that $a_{k+1} \in span\{u_1, \cdots, u_k, u_{k+1}\}$ and $u_{k+1}^\dagger u_n = 0, \forall n \leq k$. Combining $span\{a_1, \cdots, a_k\} = span\{u_1, \cdots, u_k\}$ and $u_{n_1}^\dagger u_{n_2} = \delta_{n_1 n_2}$, $\forall n_1, n_2 \leq k$, we have

$$span\{a_1, \cdots, a_{k+1}\} = span\{u_1, \cdots, u_{k+1}\} \tag{16}$$

and

$$u_{n_1}^\dagger u_{n_2} = \delta_{n_1 n_2}, \forall n_1, n_2 \leq k+1. \tag{17}$$

The quantum circuit of the $(k+1)$th step of Gram-Schmidt process is given in Fig. 2.

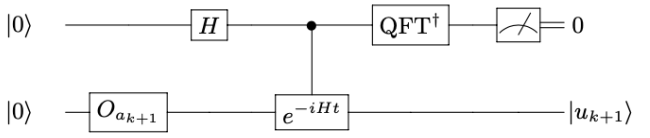

FIG. 2. Circuit constructing the $(k+1)$th state $|u_{k+1}\rangle$ based on $\{|u_1\rangle, |u_2\rangle, \cdots, |u_k\rangle\}$ and $|a_{k+1}\rangle$. $O_{a_{k+1}}$ is the oracle preparing the quantum state $|a_{k+1}\rangle$, which encodes vector a_{k+1} with $\lceil \log_2 N \rceil$ qubits in the second register. There is only one qubit in the first register. To avoid confusion, the top H is Hadamard gate and the bottom H is Hamiltonian. We post select the first register to be 0 and readout $|u_{k+1}\rangle$ from the second register. In our algorithm $\text{QFT}^\dagger$ is Hadamard gate because the size of the first register is 1. The Hamiltonian simulation is realized with qubitization with $\lceil \log_2 k \rceil + 2$ ancillary qubits for the kth step.

In the linearly dependent case for a set of the N-dimensional vectors $S = \{a_1, a_2, \cdots, a_M\}$, we also first set $|u_1\rangle = |a_1\rangle$. We assume that $a_1, \cdots, a_k$ with $k < M$ are linearly independent, then we can apply the algorithm of the linearly independent to these k vectors and obtain k orthogonal normalized vectors $\{u_1, \cdots, u_k\}$ such that $span\{a_1, \cdots, a_k\} = span\{u_1, \cdots, u_k\}$. Let us assume that the $(k+1)$th vector a_{k+1} is not linearly independent and can be expressed by the former k vectors, i.e., $a_{k+1} \in span\{a_1, \cdots, a_k\} = span\{u_1, \cdots, u_k\}$. This can be verified by following the same circuit as in the linearly independent case. That is, we encode the vector a_{k+1} as the quantum state $|a_{k+1}\rangle$ via Eq. (13) and prepare the input state $|u\rangle_s$ as $|u\rangle_s \equiv |a_{k+1}\rangle$ in the second register. Thus, the output corresponding to the input state $|u\rangle_s = |a_{k+1}\rangle$ is

$$|1\rangle \left(\sum_{n=1}^{k} \langle u_n|u\rangle_s |u_n\rangle \right) \equiv |1\rangle \left(\sum_{n=1}^{k} \langle u_n|a_{k+1}\rangle |u_n\rangle \right), \tag{18}$$

in which the outcome of the measurement on the first register is 1. It is impossible that the outcome of the measurement on the first register is 0. Thus, if we run the circuit for a certain time and the outcome for each of the measurements is not 0, we can infer $a_{k+1} \in span\{a_1, \cdots, a_k\}$. As $span\{a_1, \cdots, a_{k+1}\} = span\{u_1, \cdots, u_k\}$, we can just move on to the next step of Gram-Schmidt process to find a new $|u_{k+1}\rangle$ by considering the vector a_{k+2}. We prove in Appendix B 2 that if the circuit is run for $\kappa^2 \ln(M/\epsilon)$ times and each of the measurement outcomes is not 0, then with probability larger than $1 - \epsilon/M$, a_{k+1} is linearly dependent of $\{a_1, a_2, \cdots, a_k\}$, where κ is defined as the conditional number of the matrix $A = (a_1, a_2, \cdots, a_M)$, i.e., the ratio

of the maximum sigular value $\sigma_{\max}$ and the minimum non-zero singular value $\sigma_{\min}$.

$$\kappa = \frac{\sigma_{\max}}{\sigma_{\min}}. \tag{19}$$

Also, we need to readout the vector form of the state $|u_{k+1}\rangle$ in the computational basis to get the classical vector u_{k+1}. The readout process requires the tools of quantum state tomography [40, 41]. Suppose the state that is readout is $|u'_{k+1}\rangle$, it is required that we have multiple copies of $|u_{k+1}\rangle$ so that $|\langle u_{k+1}|u'_{k+1}\rangle| > 1-\epsilon^2$.

We now summarize our quantum Gram-Schmidt process in Algorithm 3. By running algorithm 3, a series of orthogonal normalized bases $\{u_1, u_2, \cdots, u_T\}$ are constructed, satisfying $span\{a_1, a_2, \cdots, a_M\} = span\{u_1, u_2, \cdots, u_T\}$ with $T \leq M$, thus the task of orthogonal normalization of vector set is completed. But there are still some questions left. Why do we decide the linearly dependent case with $\kappa^2 \ln(M/\epsilon)$ runs? Will errant Hamiltonian simulation lead to loss of orthogonality? How many copies of the states are needed for high-fidelity state tomography? How many quantum oracles and gates are needed? We answer these questions in the following section.

Algorithm 3: Quantum Gram-Schmidt Process for Vector Set Orthogonal Normalization

Input: $S = \{a_1, a_2, \cdots, a_M\}$, $a_i \in \mathbb{C}^N$, error ϵ
Output: $S' = \{u_1, u_2, \cdots, u_T\}$ satisfying $u_{t_1}^\dagger u_{t_2} = O(\epsilon)$, $\forall t_1 \neq t_2$, $span\{a_1, \cdots, a_M\} = span\{u_1, \cdots, u_T\}$ with succeeding probability larger than $1-\epsilon$

$|u_1\rangle \leftarrow |a_1\rangle$;
$S' \leftarrow \{u_1 \equiv a_1 / \parallel a_1 \parallel\}$;
$H \leftarrow |u_1\rangle\langle u_1|$, $t \leftarrow \pi$, $\epsilon_0 \leftarrow \epsilon/2\kappa^3$;
for $k = 1$ ***to*** $M-1$ **do**
  for $count = 0$ ***to*** $\kappa^2 \ln(M/\epsilon)$ **do**
    `// count is a counting variable with no other meaning.`
    construct e^{-iHt} to ϵ_0 with qubitization;
    run circuit with oracle $O_{a_{k+1}}$;
    measure 1st register to get result x;
    if $x = 0$ **then**
      readout $|\psi\rangle$ from the 2nd register;
      denote the vector form of $|\psi\rangle$ as ψ;
      $S' \leftarrow S' \cup \{\psi\}$;
      $H \leftarrow H + |\psi\rangle\langle\psi|$;
      break;
    end
  end
end
return S';

B. Complexity Analysis

We first provide some lemmas, then we give our main theorem and prove it based on these lemmas.

Lemma 2 (Sample complexity for pure state tomography [40, 41])**.** Given $O(N/\epsilon \ln(1/\epsilon))$ copies for an N-dimensional quantum state $|\psi\rangle$, one can apply the tools of quantum state tomography to readout a quantum state $|\phi\rangle$, such that

$$|\langle\phi|\psi\rangle| > 1-\epsilon. \tag{20}$$

Lemma 3 (Error of quantum circuit on Errant Hamiltonian Simulation)**.** Suppose A and B are square matrices. Then $\parallel A \otimes B \parallel = \parallel A \parallel \cdot \parallel B \parallel$. Suppose we use the Hamiltonian simulation tools to simulate Hamiltonian $H = \sum_{n=1}^{k} |u_n\rangle\langle u_n|$ for arbitrary time t to error ϵ_0 for each $k = 1, 2, \cdots, M$

$$\parallel e^{-iHt} - U \parallel < \epsilon_0, \tag{21}$$

then the error of unitary of the whole quantum circuit for constructing $|u_{k+1}\rangle$ as in Fig. 2 can be bounded with

$$\parallel U_{\text{real}} - U_{\text{exact}} \parallel < \epsilon_0. \tag{22}$$

Lemma 4 (Algorithm 3 Generate Complete Vector Set)**.** Suppose we use Algorithm 3 for Problem 1 to generate a series of states successively. Then $span\{a_1, a_2, \cdots, a_M\} = span\{u_1, u_2, \cdots, u_T\}$ is satisfied with the probability larger than $1-\epsilon$.

Lemma 5 (Loss of Orthogonality Based on Algorithm 3)**.** Suppose we use Algorithm 3 for Problem 1 to generate a series of states successively, then the generated states satisfy

$$\langle u_{t_1}|u_{t_2}\rangle = O(\epsilon) \;\; \forall t_1 \neq t_2. \tag{23}$$

with probability $\Omega(1)$. Therefore,

$$u_{t_1}^\dagger u_{t_2} = O(\epsilon) \;\; \forall t_1 \neq t_2 \tag{24}$$

satisfies with probability $\Omega(1)$

Detailed proofs of the lemmas above are given in Appendix B. Now we give our main result.

Theorem 1 (Quantum Gram-Schmidt Orthogonalization)**.** If we consider a vector set orthogonal normalization problem as defined in Problem 1, then there exists a quantum algorithm to generate vector set $S' = \{u_1, u_2, \cdots, u_T\}$ which satisfies $u_{t_1}^\dagger u_{t_2} = O(\epsilon)$, $\forall t_1 \neq t_2$ and $span\{a_1, a_2, \cdots, a_M\} = span\{u_1, u_2, \cdots, u_T\}$, succeeding with the probability larger than $\Omega(1)$. The query complexity of the algorithm is

$$O\left(\frac{M^2 N \kappa^2}{\epsilon} \text{poly}\log\left(\frac{M\kappa}{\epsilon}\right)\right) \tag{25}$$

and $\lceil \log_2 M \rceil + \lceil \log_2 N \rceil + 3$ qubits are needed. The total number of additional two-qubit quantum gates is larger than the query complexity by a factor $\log_2 M$.

Proof. The correctness of Algorithm 3 is proved through Lemma 4 and 5. Now we calculate the complexity of the algorithm.

In the $(k+1)$th step of the quantum Gram-Schmidt process, to simulate Hamiltonian $H = \sum_{n=1}^{k} |u_n\rangle\langle u_n|$ for a time $t = \pi$ to accuracy $\epsilon_0 = \epsilon/2\kappa^3$ as in Algorithm 3, we first rewrite the Hamiltonian H as

$$H = \frac{1}{2}\sum_{n=1}^{k}\left(2|u_n\rangle\langle u_n| - I\right) + \frac{kI}{2} = \frac{H'}{2} + \frac{kI}{2}. \tag{26}$$

where I denotes the identity matrix. Thus, the evolution matrix of the Hmailtonian H through the time t is

$$\begin{aligned}\exp\left(i\pi H\right) &= \exp\left(\frac{i\pi H' + i\pi kI}{2}\right)\\ &= \exp\left(i\frac{\pi}{2}H'\right)\exp\left(i\frac{\pi k}{2}I\right),\end{aligned} \tag{27}$$

and therefore, to simulate the evolution of the Hamiltonian H with the time π is equivalent to simulate the evolution of H' with a time $\pi/2$ with an additional operator $\exp\left(ik\pi I/2\right)$. The additional operator $\exp\left(ik\pi I/2\right)$ can be realized by applying single qubit operation on the ancillary qubit for bringing the additional phase factor. When the remainder of k modulo 4 is 0, we do not apply any additional operation. When the remainder of k modulo 4 is 1, we apply the phase gate on the ancillary qubit. When the remainder of k modulo 4 is 2, we apply the Pauli Z gate on the ancillary qubit. When the remainder of k modulo 4 is 3, we apply the inverse phase gate on the ancillary qubit. The Hamiltonian H' has an LCU form with

$$H' = \sum_{n=1}^{k}\left(2|u_n\rangle\langle u_n| - I\right) \tag{28}$$

because each of the operator $2|u_n\rangle\langle u_n| - I$ is a unitary matrix, thus $\exp(i\pi H'/2)$ can be realized with qubitization as shown in Lemma. 1. From Lemma 1 we know,

$$O\left(\alpha t + \log_2\left(\frac{1}{\epsilon_0}\right)\right) = O\left(\frac{k\pi}{2} + \log_2\left(\frac{\kappa}{\epsilon}\right)\right) \tag{29}$$

queries to quantum oracles and $\lceil\log_2 k\rceil + \lceil\log_2 N\rceil + 2$ qubits are needed to simulate the Hamiltonian H' with a time $\pi/2$. It is noted that the time complexity and number of qubits needed are different for each step. And the maximum number of qubits required for the quantum circuit is $\lceil\log_2 M\rceil + \lceil\log_2 N\rceil + 3$, including $\lceil\log_2 N\rceil$ qubits for storing $|u_T\rangle$, $\lceil\log_2 M\rceil + 2$ ancillary qubits for realizing the qubitization Hamiltonian simulation, and one ancillary qubit as the first qubit register in the quantum phase estimation.

Also in the $(k+1)$th step, it is required that the overlap between the readout state and the true state is larger than $1-\epsilon^2$. Following Lemma. 2, a total number of $(N/\epsilon\ln(1/\epsilon))$ copies of state $|u_k\rangle$ are needed, $\forall k = 0, 1, \cdots, M-1$. Thus, the total complexity for readout a classical vector u_k contains the complexity of the quantum circuit and the complexity of state readout.

The number of accesses to quantum oracles required to run the quantum circuit for one time is $O\left(k\pi + 4\log_2\left(1/\epsilon\right)\right)$ and the quantum circuit needs to be repeatedly run for at most $\kappa^2\ln(M/\epsilon)$ times for $\forall k$. Therefore, there is an upper bound on the quantum query complexity, i.e., the total number N_O of calls to oracles.

$$\begin{aligned}N_O &\le \sum_{k=1}^{M}\kappa^2\ln\left(\frac{M}{\epsilon}\right) * O\left(\frac{N}{\epsilon}\ln\left(\frac{1}{\epsilon}\right)\right) * O\left(\frac{k\pi}{2} + \log_2\left(\frac{\kappa}{\epsilon}\right)\right)\\ &= O\left(\frac{M^2N\kappa^2\ln M}{\epsilon}\left(\ln\frac{1}{\epsilon}\right)^2\right) + O\left(\frac{MN\kappa^2\ln\kappa}{\epsilon}\left(\ln\frac{1}{\epsilon}\right)^3\right)\\ &= O\left(\frac{M^2N\kappa^2\ln M\ln\kappa}{\epsilon}\left(\ln\frac{1}{\epsilon}\right)^3\right)\\ &= O\left(\frac{M^2N\kappa^2}{\epsilon}\mathrm{poly}\log\left(\frac{M\kappa}{\epsilon}\right)\right).\end{aligned} \tag{30}$$

Moreover, there is an upper bound on the total number of additional two-qubit quantum gates N_G with

$$\begin{aligned}N_G &\le \sum_{k=1}^{M}\kappa^2\ln\left(\frac{M}{\epsilon}\right) * O\left(\frac{N}{\epsilon}\ln\left(\frac{1}{\epsilon}\right)\right)\\ &\quad * O\left(\log_2 k\left(\frac{k\pi}{2} + \log_2\left(\frac{\kappa}{\epsilon}\right)\right)\right)\\ &= O\left(\frac{M^2N\kappa^2\ln\kappa\left(\ln M\right)^2}{\epsilon}\left(\ln\frac{1}{\epsilon}\right)^3\right)\\ &= O\left(\frac{M^2N\kappa^2}{\epsilon}\mathrm{poly}\log\left(\frac{M\kappa}{\epsilon}\right)\right)\end{aligned} \tag{31}$$

Thus, the proof of theorem 1 is completed. □

C. Validations of algorithm

In this section, we further show the correctness and the performance of our proposed algorithms under different situations. We first calculate the orthogonality of generated vectors $S' = \{u_1, u_2, \cdots, u_T\}$ with different dimension of input vectors $S = \{a_1, a_2, \cdots, a_M\}$. Without loss of generality, we assume that the dimension N of the input vector is the same as the number M of input vectors. The loss of orthogonality of the generated $S' = \{u_1, u_2, \cdots, u_T\}$ is calculated by

$$\eta = \parallel S'^{\dagger}S' - I \parallel. \tag{32}$$

We fix the conditional number κ of the generated matrix S to be 100. To randomly generate matrices with fixed conditional number κ and size N, we first define a diagonal matrix D whose conditioanl number is κ and size is N with

$$D = diag\left\{1, \kappa^{-\frac{1}{N-1}}, \kappa^{-\frac{2}{N-1}}, \cdots, \kappa^{-\frac{N-2}{N-1}}, \kappa^{-1}\right\}. \tag{33}$$

Then two unitary matrices U and V are sampled randomly from the group $U(N)$, and finally we define $S = UDV^\dagger$ as a generated matrix with conditional number κ and size N.

We use classical Gram-Schmidt process and quantum Gram-schmidt process respectively to generate a set of orthogonal normalized vectors $S' = \{u_1, u_2, \cdots, u_T\}$. Then the loss of orthogonality of $\{u_1, u_2, \cdots, u_T\}$ is calculated. The parameter ϵ in the quantum Gram-Schmidt process is chosen to be 10^{-4}. We change the value of N and summarize the results in Fig. 3(a). It is noted that all our numerical simulations are realized with the noiseless state-vector simulator of Qiskit, a Python package developed by IBM. During the simulation, the Hamiltonian simulation step is realized ideally with no error. The evolution matrix of Hamiltonian H during time t is executed exactly as $\exp(-iHt)$.

Figure 3(a) shows that there is some loss of orthogonality of the set $\{u_1, u_2, \cdots, u_T\}$ which is generated by the quantum Gram-Schmidt process. It is because the whole quantum algorithm is simulated on a classical computer, so the effect of the rounding error cannot be neglected, which is a tiny difference between the ideal mathematical value and the value the computer actually uses. The rounding error mainly result from the inaccurate representation and calculation of the real number on a classical computer. Specifically, rounding error of the proposed quantum algorithm mainly include the calculation and representation of the evolution matrix $\exp(-iHt)$, the representation of the Hadamard gate, the representation of the quantum states, and multiplication of each quantum gate and the quantum state. All these rounding errors contribute to the error in the final result. The more computational steps there are, the greater the impact of the rounding error of the classical computer on the final result is.

The whole quantum algorithm is run on a classical state-vector based simulator, which needs to calculate the evolution operator $\exp(-iHt)$ to simulate the quantum circuit, the simulation of quantum circuits on a classical computer is inefficient. Thus, more computational steps are required by simulating quantum algorithms on a classical computer than doing the classical Gram-Schmidt process on a classical computer. Therefore, the error of quantum Gram-Schmidt process is a little larger than that of classical Gram-Schmidt process because more computational steps bring larger rounding error. However, even with a large rounding error, the error of the quantum Gram-Schmidt process is less than 10^{-10}, which indicates that there is no system error. Thus, the correctness of Algorithm 3 is proved.

We also study the performance of the quantum Gram-Schmidt process when $S = \{a_1, a_2, \cdots, a_N\}$ is an ill-conditional matrix with large conditional number. In this case, the conditional number of the matrix $S = (a_1, a_2, \cdots, a_N)$ is large. We randomly generate matrix S with different conditional numbers, fixing S to be an 8×8 matrix. We use quantum Gram-Schmidt process to generate a series of orthogonal normalized vectors $S' = \{u_1, u_2, \cdots, u_T\}$ for $S = (a_1, a_2, \cdots, a_N)$. The loss of orthogonality is calculated. The parameter ϵ in our quantum algorithm is chosen to be 10^{-4}. We change the value of the conditional number and summarize the results in Fig. 3(b). It is shown from Fig. 3(b) that the loss of orthogonality of $S' = \{u_1, u_2, \cdots, u_T\}$ remains small even linear independence of $S = \{a_1, a_2, \cdots, a_N\}$ is weak. This shows the robustness of Algorithm 3 in the case that the linear independence of input vectors is weak.

It is noted that we only examine whether Algorithm 3 generates a set of orthogonal normalized vectors, but the completeness of the generated vectors is not examined, i.e.,

$$span\{a_1, a_2, \cdots, a_M\} = span\{u_1, u_2, \cdots, u_T\} \quad (34)$$

is not proved numerically. This is because we will propose quantum Gram-Schmidt process based QR decomposition algorithm in the next section. When we numerically prove the correctness of our quantum QR decomposition algorithm, the completeness of the generated vectors is verified at the same time.

V. QUANTUM QR DECOMPOSITION ALGORITHM

In this section, we further propose a quantum QR decomposition algorithm based on our proposed quantum Gram-Schmidt process and a subroutine called quantum inner product estimation. We first introduce a quantum inner product estimation subroutine, and then propose our quantum QR decomposition algorithm.

A. Complexity of Quantum Inner Product Estimation

Quantum inner product estimation is a method used to calculate the overlap between any two quantum states. Given two N-dimension quantum states $|x\rangle$ and $|y\rangle$, a successful quantum inner product estimation algorithm gives the value $|\langle x|y\rangle|^2$ or $\langle x|y\rangle$ with probability $O(1)$. The most common algorithm is the SWAP test. However, SWAP test can only be used to calculate the overlap $|\langle x|y\rangle|^2$, and cannot calculate the value of the inner product $\langle x|y\rangle$ as a complex number. An algorithm for estimating $\langle x|y\rangle$ was proposed [42] and can be a subroutine for our quantum QR decomposition algorithm. However, its complexity has not been proven. We now prove the complexity of quantum inner product estimation [42] and apply it to the analysis of the complexity of our proposed quantum QR decomposition algorithm.

As shown in Fig. 4, suppose O_x and O_y are given as oracles for state preparation. $O_x|0\rangle = |x\rangle$, $O_y|0\rangle = |y\rangle$.

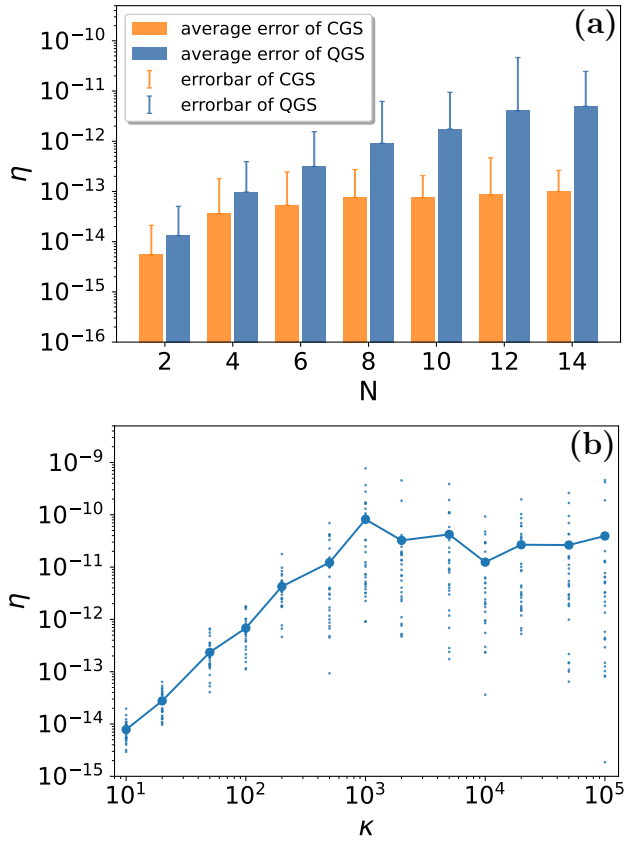

FIG. 3. (a) Loss of orthogonality for different dimensions of the input vector. CGS is for classical Gram-Schmidt process and QGS is for quantum Gram-Schmidt process. N is the system dimension and η is the loss of orthogonality. Error bar is also shown. (b) Loss of orthogonality for different conditional numbers of the matrix lined by input vectors. QGS is for quantum Gram-Schmidt process. κ is the conditional number of matrix $\mathcal{S}$ and η is the loss of orthogonality. Multiple tests for one conditional number are performed. The result of each test is marked with a small blue dot. The average error is marked with big blue dots.

Then before the quantum measurements, the output of the quantum circuit as in Fig. 4 is

$$|\phi\rangle = \frac{1}{2}(|0\rangle(|x\rangle + |y\rangle) + |1\rangle(|x\rangle - |y\rangle)). \tag{35}$$

Then the probability to obtain $|0\rangle$ is given by

$$p = \frac{1}{2}(1 + \mathrm{Re}(\langle x|y\rangle)) \tag{36}$$

when the first qubit is measured in the computational basis. Thus, the real part of $\langle x|y\rangle$ is encoded in the amplitude of the quantum state $|\phi\rangle$.

Similarly, by running the quantum circuit as in Fig. 5, the imaginary part of the quantum state $\langle x|y\rangle$ is encoded into the amplitude. Then the probability to obtain $|0\rangle$ is given by

$$p' = \frac{1}{2}(1 + \mathrm{Im}(\langle x|y\rangle)), \tag{37}$$

when the first qubit is measured in computational basis

It is clear that $\mathrm{Re}\langle x|y\rangle$ and $\mathrm{Im}\langle x|y\rangle$ can be estimated via the measurement on the quantum state $|0\rangle$ for the first qubit when quantum circuits as shown in Fig. 4 and Fig. 5 are separately run for many times. That is, we record the result of each measurement for the first qubit and use the average of the results to obtain the probabilities p and p' in Eq. (36) and Eq. (37), respectively. Thus, $\mathrm{Re}\langle x|y\rangle$ and $\mathrm{Im}\langle x|y\rangle$ can be obtained, and the task of estimating the inner product of quantum states is completed.

Lemma 6 (Quantum inner product estimation)**.** Given $\epsilon, \delta > 0$, for any two N-dimension quantum states $|x\rangle$ and $|y\rangle$, there exists a quantum algorithm that outputs $\widetilde{\langle x|y\rangle}$ as an estimation of inner product of these two quantum states, satisfying $|\widetilde{\langle x|y\rangle} - \langle x|y\rangle| \leq \epsilon$ with probability larger than $1-\delta$, using

$$O\left(\frac{1}{\epsilon^2}\log_2\left(\frac{1}{\delta}\right)\right) \tag{38}$$

calls to quantum oracles and $\lceil \log_2 n \rceil + 1$ qubits, where n is the size of $|x\rangle$ and $|y\rangle$.

Proof. We take the quantum circuits in Fig. 4 and Fig. 5 for estimating $\mathrm{Re}\langle x|y\rangle$ and $\mathrm{Im}\langle x|y\rangle$, respectively. Suppose the total number of running quantum circuit for estimating $\mathrm{Re}\langle x|y\rangle$ is N_r, so the total number of quantum oracle calls is $2N_r$. Each measurement result $X_i = 0$ or $1, i = 1, 2, \cdots, N_r$, then according to Hoeffding's inequality,

$$P\left(\left|\frac{1}{N_r}\sum_{i=1}^{N_r} X_i - \frac{1+\mathrm{Re}\langle x|y\rangle}{2}\right| \leq \epsilon\right) \geq 1 - 2\exp\left(-2N_r\epsilon^2\right). \tag{39}$$

Thus, when we take $N_r = (16/\epsilon^2)\log_2(4/\delta)$ and $\mathrm{Re}\widetilde{\langle x|y\rangle} = (2/N_r)\sum_{i=1}^{N_r} X_i - 1$, we have

$$P\left(|\mathrm{Re}\widetilde{\langle x|y\rangle} - \mathrm{Re}\langle x|y\rangle| \leq \frac{\epsilon}{2}\right) \geq 1 - \frac{\delta}{2}. \tag{40}$$

Similarly, we can estimate $\mathrm{Im}\langle x|y\rangle$ to error $\epsilon/2$ with probability larger than $\delta/2$ with $16\epsilon^{-2}\log_2(4/\delta)$ runs of

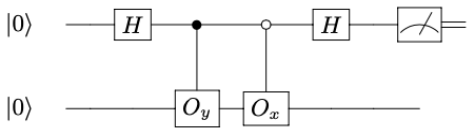

FIG. 4. Quantum circuit for estimating $\mathrm{Re}\langle x|y\rangle$, the probability that the measurement is 0 is $(1+\mathrm{Re}\langle x|y\rangle)/2$

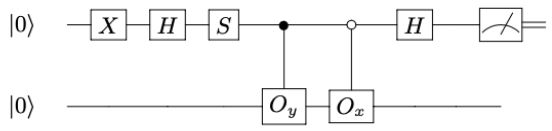

FIG. 5. Quantum circuit for estimating $\mathrm{Im}\langle x|y\rangle$, the probability that the measurement is 0 is $(1+\mathrm{Im}\langle x|y\rangle)/2$

circuit in Fig. 5, and obtain

$$P\left(|\widetilde{\mathrm{Im}\langle x|y\rangle}-\mathrm{Im}\langle x|y\rangle|\leq\frac{\epsilon}{2}\right)\geq 1-\frac{\delta}{2}. \tag{41}$$

Combining Eq. (40) and Eq. (41), we have

$$P\left(|\widetilde{\langle x|y\rangle}-\langle x|y\rangle|\leq\epsilon\right)\geq\left(1-\frac{\delta}{2}\right)^2>1-\delta. \tag{42}$$

The total number of runing quantum circuit is $32\epsilon^{-2}\log_2(4/\delta)$, thus the total number of quantum oracle calls is

$$64\epsilon^{-2}\log_2(4/\delta)=O\left(\frac{1}{\epsilon^2}\log_2\left(\frac{1}{\delta}\right)\right) \tag{43}$$

The quantum circuits use $\lceil\log_2 N\rceil$ qubits to encode $|x\rangle$ and $|y\rangle$. One auxiliary qubit is also needed. Thus, the total number of qubits required is $\lceil\log_2 N\rceil+1$. The proof of Lemma 6 is completed. □

B. Algorithm Description and Complexity Analysis

Algorithm 2 shows that the matrices Q and R of the matrix QR decomposition can be constructed through Gram-Schmidt process. The columns of matrix Q are obtained by orthogonal normalization of the columns of the matrix A. The non-diagonal matrix elements of matrix R can be obtained by calculating the inner product of the columns of matrix A and matrix Q. Based on classical QR decomposition algorithm, we now propose a quantum QR decomposition algorithm. The basic idea of the algorithm is that the matrix Q is constructed by our proposed quantum Gram-Schmidt orthogonalization process, and then the matrix R is calculated with quantum inner product estimation. The proposed algorithm takes a full-rank complex matrix $A\in\mathbb{C}^{N\times M}$ as input where $N\geq M$, and outputs an orthogonal matrix Q and an upper triangular matrix R.

Our quantum QR decomposition algorithm is described as follows. We take each column of the matrix A as a N-dimensional vector. Thus, the matrix A has M vectors $\{a_1,\cdots,a_M\}$. The N components of each N-dimensional vector a_m is a_{nm} with $n=1,\cdots,N$. It is clear that a_{nm} is the matrix elements of the matrix A. Thus, we use the same steps of the quantum Gram-Schmidt orthogonalization algorithm to obtain orthogonal normalized vectors $\{\tilde{q}_1,\tilde{q}_2,\cdots,\tilde{q}_M\}$, which form the $\tilde{Q}$ matrix with the matrix elements $\tilde{q}_{nm}$. After obtaining the matrix $\tilde{Q}$, we perform quantum inner product estimation algorithm to calculate each element $\tilde{R}_{m_1m_2}=\tilde{q}_{m_2}^{\dagger}a_{m_1}$ of the matrix $\tilde{R}$. Thus, the quantum QR decomposition algorithm is completed.

It is noted that the QR decomposition exists only for matrices with full column rank. Usually, QR decomposition of a full-rank matrix is not unique. Following the classical Gram-Schmidt based QR decomposition algorithm, one solution $A=QR$ is generated. Our quantum algorithm generates matrix $\widetilde{Q}$ and matrix $\widetilde{R}$, which are ϵ approximation to matrix Q and matrix R generated by classical Gram-Schmidt based QR decomposition algorithm. Our algorithm is summarized in Algorithm 4.

Algorithm 4: Quantum QR Decomposition

Input: $A=\{a_1,a_2,\cdots,a_M\}\in\mathbb{C}^{N\times M}$ with full rank and $N\geq M$, error ϵ

Output: $\tilde{Q}=\{\tilde{q}_1,\tilde{q}_2,\cdots,\tilde{q}_M\}\in\mathbb{C}^{N\times M}$, $\tilde{R}\in\mathbb{C}^{M\times M}$, satisfying $\parallel\tilde{q}_m-q_m\parallel=O(\epsilon),\forall m$, $\widetilde{R}_{m_1m_2}=0,\forall m_1>m_2$, and $|\widetilde{R}_{m_1m_2}-R_{m_1m_2}|=O(\epsilon\parallel A\parallel)$, where matrix Q and matrix R is the exact unique decomposition solution via classical Gram-Schmidt process.

```
|q̃_1⟩ ← |a_1⟩;
H ← |q̃_1⟩⟨q̃_1|, t ← π, ε_0 ← ε/4κ^3;
for k = 1 to M − 1 do
    while True do
        construct e^{−iHt} to ε_0 with qubitization;
        run circuit with oracle O_{a_{k+1}};
        measure 1st register to get result x;
        if x = 0 then
            measure 2nd register and get |q_{k+1}⟩;
            H ← H + |q̃_{k+1}⟩⟨q̃_{k+1}|;
            break;
        end
        // If matrix A is not full rank with
           a_{k+1} ∈ span{a_1, a_2, ··· , a_k}, then this
           is an endless loop.
    end
end
Q̃ ← {|q̃_1⟩, |q̃_2⟩, ··· , |q̃_M⟩} ;
for m_1 = 1 to M do
    for m_2 = 1 to m_1 − 1 do
        use QIPE to calculate ⟨q̃_{m_2}|a_{m_1}⟩ with error
          rate ε, success probability larger than
          1 − ε/M^2;
        R̃_{m_2 m_1} ← ‖ a_{m_1} ‖ ⟨q̃_{m_2}|a_{m_1}⟩;
    end
    R̃_{m_1 m_1} ← ‖ a_{m_1} − Σ_{m_2=1}^{m_1−1} R̃_{m_2 m_1} q_{m_2} ‖ ;
end
return Q,R;
```

Theorem 2 (Quantum QR Decomposition)**.** Consider a matrix QR decomposition problem as defined in Problem

2. Given a full rank matrix $A \in \mathbb{C}^{N\times M}$ with $N \geq M$ and conditional number κ. Then there exists a quantum algorithm for calculating the QR decomposition of matrix A. Suppose the unique exact QR decomposition of matrix A is $A = QR$, then the algorithm generates matrix $\tilde{Q}$ and matrix $\tilde{R}$, satisfying $\parallel \tilde{q_m} - q_m \parallel = O(\epsilon), \forall m$, $R_{m_1m_2} = 0, \forall m_1 > m_2$, and $|\widetilde{R}_{m_1m_2} - R_{m_1m_2}| = O(\epsilon \parallel A \parallel)$, succeeding with probability larger than $1-\epsilon$. The query complexity of the algorithm is

$$O\left(\frac{M^2N\kappa^2}{\epsilon}\text{poly}\log\left(\frac{M\kappa}{\epsilon}\right), \frac{M^2}{\epsilon^2}\text{poly}\log\left(\frac{M}{\epsilon}\right)\right), \tag{44}$$

and $\lceil\log_2 M\rceil + \lceil\log_2 N\rceil + 3$ qubits are needed. The total number of additional two-qubit quantum gates is larger than the query complexity by a factor $\log_2 M$.

Proof. We now analyze the complexity of Algorithm 4, which proves the theorem above. First, Algorithm 4 takes Algorithm 3 as a subroutine. Our quantum QR decomposition algorithm outputs matrix $\tilde{Q}$ as an approximation to matrix Q with Algorithm 3. From theorem 1 we know that

$$O\left(\frac{M^2N\kappa^2}{\epsilon}\text{poly}\log\left(\frac{M\kappa}{\epsilon}\right)\right) \tag{45}$$

calls to quantum oracles, $\lceil\log_2 M\rceil + \lceil\log_2 N\rceil + 3$ qubits are needed, and from Eq. (B46) we know

$$\parallel \tilde{q}_i - q_i \parallel = O(\epsilon). \tag{46}$$

And the probability that the quantum Gram-Schmidt process runs correctly is larger than $1-\epsilon/2$.

For each nonzero item in matrix R, we use quantum inner product estimation algorithm for estimating it. For quantum inner product estimation, we take the estimation accuracy to be ϵ and the success probability to be $1-\epsilon/M^2$. As there are at most $M^2/2$ nonzero items in matrix R, so the probability that we estimate each of the item to accuracy ϵ is larger than

$$\prod_{n=1}^{M^2/2}\left(1-\frac{\epsilon}{M^2}\right) > 1-\frac{\epsilon}{2}. \tag{47}$$

So, the probability that each item $\widetilde{R}_{m_1m_2}$ satisfies

$$\begin{aligned}
&|\widetilde{R}_{m_1m_2} - R_{m_1m_2}| \\
<\ &|\widetilde{R}_{m_1m_2} - \langle\tilde{q}_{m_1}|a_{m_2}\rangle \parallel a_{m_2} \parallel| \\
&+ |\langle\tilde{q}_{m_1}|a_{m_2}\rangle \parallel a_{m_2} \parallel - \langle q_{m_1}|a_{m_2}\rangle \parallel a_{m_2} \parallel| \\
<\ &\parallel a_{m_2} \parallel \epsilon + \parallel a_{m_2} \parallel * \parallel |\tilde{q}_{m_1}\rangle - |q_{m_1}\rangle \parallel \\
<\ &2\epsilon \parallel a_{m_2} \parallel \\
=\ &O(\epsilon \parallel A \parallel)
\end{aligned} \tag{48}$$

is larger than $1-(\epsilon/2)$. For each item estimation, from Lemma 6 we know that

$$O\left(\frac{1}{\epsilon^2}\log_2\left(\frac{M^2}{\epsilon}\right)\right) \tag{49}$$

calls to quantum oracles and $\lceil\log_2 N\rceil + 1$ qubits are needed. So to estimate all the nonzero items, a total of

$$O\left(\frac{M^2}{\epsilon^2}\log_2\left(\frac{M^2}{\epsilon}\right)\right) \tag{50}$$

calls to quantum oracles and $\lceil\log_2 N\rceil + 1$ qubits are needed.

Summarily, the probability that $\parallel \tilde{q_m} - q_m \parallel = O(\epsilon), \forall m$ is larger than $1-\epsilon/2$ and the probability that $|\widetilde{R}_{m_1m_2} - R_{m_1m_2}| = O(\epsilon \parallel A \parallel), \forall m_1, m_2$ is also larger than $1-\epsilon/2$. So the probability that the two results hold true at the same time is larger than $1-\epsilon/2$. In this case, the whole algorithm is successful. The total qubit number needed is

$$\max\left(\lceil\log_2 M\rceil + \lceil\log_2 N\rceil + 3, \lceil\log_2 N\rceil + 1\right), \tag{51}$$

i.e., $\lceil\log_2 M\rceil + \lceil\log_2 N\rceil + 3$. The total number of calls to quantum oracles is

$$\max\left(O\left(\frac{M^2N\kappa^2}{\epsilon}\text{poly}\log\left(\frac{M\kappa}{\epsilon}\right)\right), O\left(\frac{M^2}{\epsilon^2}\log_2\left(\frac{M^2}{\epsilon}\right)\right)\right), \tag{52}$$

i.e.,

$$O\left(\frac{M^2N\kappa^2}{\epsilon}\text{poly}\log\left(\frac{M\kappa}{\epsilon}\right), \frac{M^2}{\epsilon^2}\text{poly}\log\left(\frac{M}{\epsilon}\right)\right). \tag{53}$$

The total number of additional two-qubit quantum gates is larger than the query complexity by a factor $\log_2 M$. Thus, the proof of Theorem 2 is completed. □

C. Validations of algorithm

In this section, we further show the correctness and the performance of our proposed quantum QR decomposition algorithms for the matrix A under different situations. We first calculate the error of the quantum QR decomposition algorithm which outputs an orthogonal matrix $\tilde{Q}$ and an upper triangle matrix $\tilde{R}$. The error of the quantum QR decomposition algorithm is defined as

$$\eta = \parallel A - \tilde{Q}\tilde{R} \parallel . \tag{54}$$

We change the dimension of the input matrix A and calculate the error η. Without loss of generality, we take matrix A to be a square matrix.

We randomly generate the matrix $A \in \mathbb{C}^{N\times N}$. The conditional number of matrix A is fixed to be 100. We use classical Gram-Schmidt process based QR decomposition algorithm and quantum Gram-schmidt process based QR decomposition algorithm respectively to decompose matrix A into matrix $\tilde{Q}$ and matrix $\tilde{R}$. Then the error $\eta = \parallel A - \tilde{Q}\tilde{R} \parallel$ is calculated. The parameter ϵ in the quantum Gram-Schmidt process is chosen to be 10^{-4}.

We change the value of N and summarize the results in Fig. 6(a).

It is shown in Fig. 6(a) that the error of quantum Gram-Schmidt process based QR decomposition algorithm is a little larger than that of the classical Gram-Schmidt process based QR decomposition algorithm. As discussed above in Section IV C, it is because the whole process is simulated on a classical computer, so there is a rounding error in each step of calculations, and simulating quantum algorithms requires more computational steps. With rounding error, the error of the quantum Gram-Schmidt process based QR decomposition algorithm is less than 10^{-11}, which indicates that there is no system error.

It is noted that we have calculated the loss of orthogonality of the matrix $\tilde{Q}$ in Section IV C. We show that the loss of orthogonality of the matrix $\tilde{Q}$, i.e., $\parallel \tilde{Q}^{\dagger}\tilde{Q} - I \parallel$ is less than 10^{-10}. Meanwhile, the matrix $\tilde{R}$ generated from Algorithm 4 is strictly an upper triangle matrix. Thus, we prove numerically that Algorithm 4 generates an orthogonal matrix $\tilde{Q}$ and an upper triangle matrix $\tilde{R}$. Combining the fact that the error $\eta = \parallel A - \tilde{Q}\tilde{R} \parallel$ is less than 10^{-11}, the correctness of Algorithm 4 is proved.

We also study the performance of the quantum Gram-Schmidt process based QR decomposition algorithm for ill-conditional matrix A, of which the conditional number κ is large. We randomly generate the matrix A with different conditional numbers, fixing matrix A to be an 8×8 matrix. We use Algorithm 4 to generate matrix $\tilde{Q}$ and matrix $\tilde{R}$ for an input matrix A and calculate the error η. We change the conditional number κ and the parameter ϵ and summarize the results in Fig. 6(b). For drawing, we define $\chi = \log_{10}\eta = \log_{10} \parallel A - \tilde{Q}\tilde{R} \parallel$, $\kappa' = \log_{10}\kappa$, and $\epsilon' = \log_{10}\epsilon$.

For a fixed ϵ, it is noticed that as the conditional number grows larger, the error η first remains under 10^{-11}, then grows rapidly and finally becomes stable at the value about ϵ. This is because when the conditional number becomes larger, the linear independence of the columns of the matrix A is weak. Thus, at certain steps of the quantum Gram-Schmidt process, our algorithm may mistake the linearly independent case as the linearly dependent one. This leads to the rapid growth of the error. It is shown that this error can be reduced by using smaller ϵ. From the numerical simulation, it is shown that when the conditional number $\kappa \in (0, \epsilon^{-1})$, the error is under 10^{-11}, in this case, the quantum QR decomposition algorithm generates $\tilde{Q}$ which satisfies

$$span\{\tilde{q_1}, \tilde{q_2}, \cdots, \tilde{q_M}\} = span\{a_1, a_2, \cdots, a_M\}. \quad (55)$$

However, when $\kappa \in (\epsilon^{-1}, +\infty)$, it is shown numerically that the error η satisfies

$$\eta < \epsilon. \quad (56)$$

Therefore, the performance of Algorithm 4 under different size and conditional number of the input matrix is examined. The numerical simulations show the correctness and robustness of Algorithm 4.

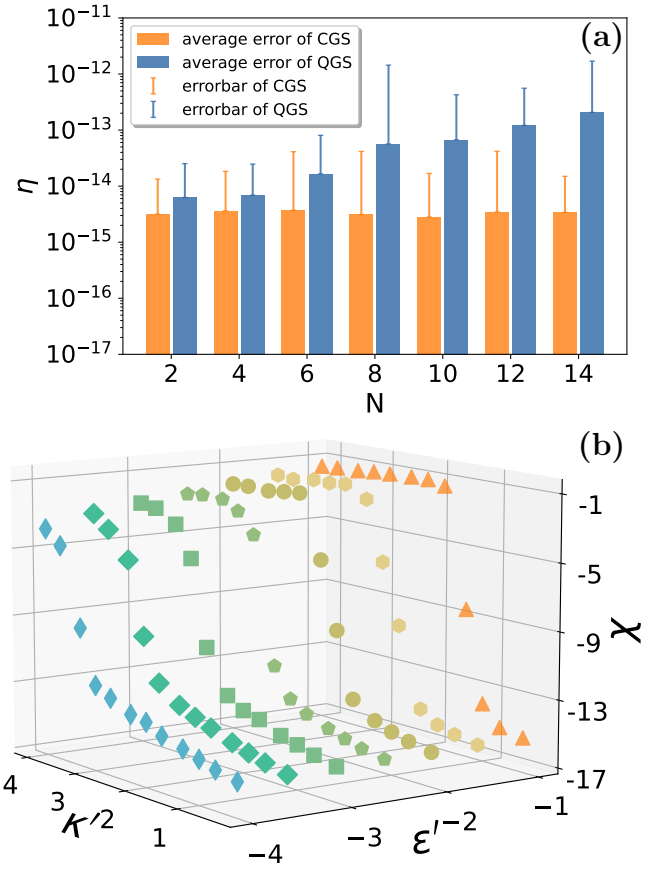

FIG. 6. (a) CGS is for classcial Gram-Schmit process based QR decomposition algorithm and QGS is for quantum Gram-Schmit process based QR decomposition algorithm. N is the system dimension and η is the error of QR decomposition. The average error and error bar are calculated and shown. (b) Error of quantum QR decomposition algorithm for different conditional numbers of input matrix and different values of ϵ in the algorithm. The conditional number is denoted as κ. We here take $\chi = \log_{10}\eta = \log_{10} \parallel A - \tilde{Q}\tilde{R} \parallel$, $\kappa' = \log_{10}\kappa$, and $\epsilon' = \log_{10}\epsilon$.

VI. APPLICATIONS

The quantum Gram-Schmidt algorithm can be independently applied to solve vector orthogonal normalization problem. It can also be a subroutine for quantum QR decomposition algorithm. In this section, we mainly show the applications of the proposed quantum QR decomposition algorithm by several examples, e.g., linear least squares regression, solving linear equations, and eigenvalues.

A. Linear Least Squares Regression

The least squares problem is a kind of regression problem, and the general form of least squares problem is

$$\min_x |f(x)|^2, \tag{57}$$

where $f(x)$ is the residual function and represents the difference between the predicted and measured value, and the loss function is $f(x)^2$. When $f(x) = Ax - b$, the least squares problem is a linear least squares problem. The accurate solution can be derived, by letting

$$\frac{\partial \parallel Ax - b \parallel^2}{\partial x} = A^T Ax - A^T b = 0. \tag{58}$$

Thus $x = \left(A^T A\right)^{-1} A^T b$. But to obtain $(A^T A)^{-1}$ is hard. However, when matrix A is a full column-rank matrix, we can use QR decomposition $A = QR$ to simplify the solution, i.e.,

$$\begin{aligned} x &= \left(A^T A\right)^{-1} A^T b = \left(R^T Q^T QR\right)^{-1} R^T Q^T b \\ &= R^{-1} Q^T b. \end{aligned} \tag{59}$$

As matrix R is an upper triangle matrix, the inversion of matrix R can be obtained classically within $O(M^2)$, where M is the column number of the matrix A. Thus, the whole process is much faster than calculating the inversion of $A^T A$. Therefore, we can get an algorithm for linear least squares regression using quantum QR decomposition algorithm as a subroutine. We decompose matrix A with quantum QR decomposition algorithm proposed in Algorithm 4, then solve the linear equations $Rx = Q^T b$ classically to get x.

It is noted that by solving Eq. (58) directly with the techniques for solving linear equations [24, 25, 43], the least squares fitting problem can also be solved. We compare our methods for solving the least squares fitting problem with these methods and summarized the results in Table. I. It is shown that our quantum QR decomposition based algorithm does not require the preprocess step for calculating $A^\dagger A$, making it more efficient.

Let us now verify our algorithm using an example of polynomial fitting. We assume that a set of data $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ is given. The data set is generated by a polynomial function

$$y_i = g(x_i) = \sum_{l=0}^{r} a_l x_i^l. \tag{60}$$

Based on this data set, we find a polynomial $f(x)$ to fit the data by least square fitting using quantum QR decomposition as a subroutine so that the mean square error is minimized. We assume

$$f(x_i) = \sum_{l=0}^{k} m_l x_i^l. \tag{61}$$

Then the coefficients m_l can be obtained with least square fitting method with

$$\begin{aligned} \overrightarrow{m} &= \min_{\overrightarrow{m}} \sum_{n=1}^{N} \left(\sum_{l=0}^{k} m_l x_i^l - y_i \right)^2 \\ &= \min_{\overrightarrow{m}} \parallel X\overrightarrow{m} - Y \parallel^2, \end{aligned} \tag{62}$$

where

$$X = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2 & x_2^2 & \cdots & x_2^k \\ 1 & x_3 & x_3^2 & \cdots & x_3^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^k \end{pmatrix} \tag{63}$$

and

$$Y = \begin{pmatrix} y_1 & y_2 & y_3 & \cdots & y_n \end{pmatrix}^T \tag{64}$$

The items in the polynomial base set $\{1, x, x^2, \cdots, x^k\}$ are orthogonal to each other, the matrix X as in Eq. (63) is full rank and has a QR decomposition. Thus, our quantum QR decomposition algorithm can be applied to least square fitting. It is noted that to fit polynomial function $g(x)$ as in Eq. (60), it is important that the order of polynomial function k as in Eq. (61) is chosen appropriately. For fixed r as in Eq. (60), the fitting function $f(x)$ in Eq. (61) is under fitting when $k < r$, appropriate when $k = r$, and over fitting when $k > r$.

In Fig. 7, we show the performance of our algorithm. The training dataset is generated with a randomly chosen polynomial function $g(x) = 0.86x^2 + 0.50x + 0.43$. We generate ten data points $\{(x_1, y_1), (x_2, y_2), \cdots, (x_{10}, y_{10})\}$, which satisfy

$$y_i = g(x_i) = 0.43x_i^2 + 0.5x_i + 0.86, \forall i = 1, 2, \cdots, 10 \tag{65}$$

as in Eq. (60). Then we use a linear function, a quadratic function, and a cubic function to fit the generated training data points respectively, i.e., the value of k as in Eq. (61) is chosen as 1, 2, and 3, respectively. Quantum QR decomposition algorithm is used to calculate the coefficients in Eq. (62). We plot the generated data points, the correct polynomial function, and the fitted polynomial function in Fig. 7 for $k = 1$, 2, 3, respectively. We also generate fifteen test data points, which satisfy the correct polynomial function $g(x)$ in Eq. (60). We use test data points to observe if the fitted polynomial function is under-fitting, appropriate, or over-fitting.

The results show in Fig. 7 that the linear function cannot fit both the training data and the test data and is the under-fitting case. The quadratic function fits both the training data and the test data well, which is the appropriate case. The cubic function fits well on the training data but cannot fit the test data and is the over-fitting case. The results above agree with the conclusion of classical machine learning. Therefore, the correctness of our quantum QR decomposition algorithm is verified.

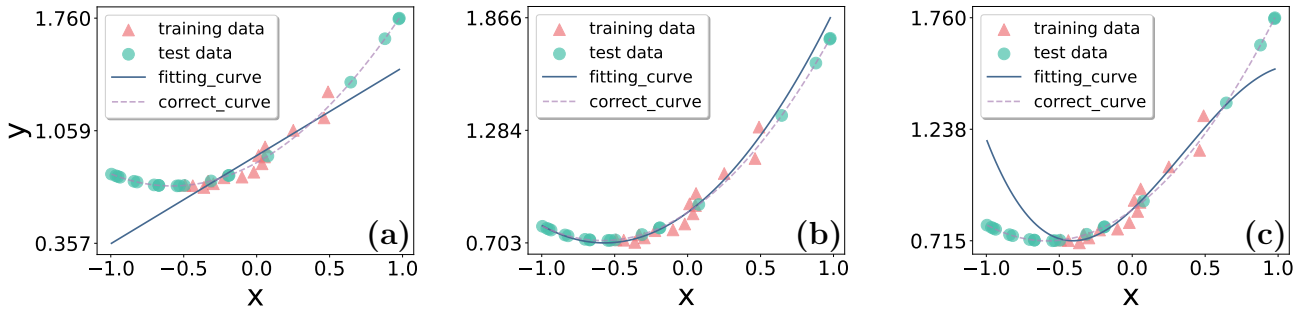

FIG. 7. Fitting 10 generated data points (the red triangles) with quantum QR decomposition algorithm based least squares fitting method. The order of the polynomial function to be fitted (the purple dashed curve) is chosen to be 2. The order of the fitting polynomial function (the deep blue line) is chosen to be k. And test data points (the green dots) are shown. The correct polynomial function is $g(x) = 0.43x^2 + 0.5x + 0.86$. (a) $k = 1$, under-fitting case. The fitting function is $f(x) = 0.54x + 0.90$, and the relative error of test data points is 0.249. (b) $k = 2$, appropriate fitting case. The fitting function is $f(x) = 0.48x^2 + 0.55x + 0.86$, and the relative error of test data points is 0.042.(c) $k = 3$, over-fitting case. The fitting function is $f(x) = -0.47x^3 + 0.50x^2 + 0.54x + 0.86$, and the relative error of test data points is 0.193..

Moreover, we use quantum QR decomposition algorithm based least squares fitting method to observe under-fitting, appropriate fitting, and over-fitting phenomena for more general cases. For fixed r as in Eq. (60), we randomly choose a r-order polynomial function $g(x)$ and generate 10 data points according to $g(x)$. Different orders k as in Eq. (61) of polynomial functions are chosen for fitting the given data. We calculate the fitting polynomial $f(x)$ using quantum QR decomposition algorithm. We further generate 100 test data and calculate the fitting error on test data. Then we change r and repeat the process. The relative test error ϵ_r is recorded and shown in Fig. 8.

From classical machine learning theory, we know that the lower left region of the figure is the under-fitting region, satisfying $k < r$. The upper right region of the figure is the over-fitting region, satisfying $k > r$. The test error is large in both of these areas. The polynomial order used for fitting is selected appropriately on the diagonals of the figure satisfying $k = r$, and the test error is minimized. Our quantum QR decomposition algorithm based least squares fitting method reproduces this result as in Fig. 8, which indicates the correctness of our algorithm.

B. Solving Linear Equations

Solving linear equations is a fundamental problem in linear algebra. Given a matrix $A \in \mathbb{C}^{N\times M}$ and a column vector $b \in \mathbb{C}^N$, define a system of linear equations

$$Ax = b. \tag{66}$$

For the general case, we need to judge whether the system has a unique solution, infinite solutions, or no solution, and give the solution of the system when the system has a unique solution. In general, the matrix A is not necessarily a square matrix. Based on matrix analysis and linear algebra, we know that the system of linear equations has solutions when the given right-side vector b is in the linear space spanned by the columns of matrix A. In this case, the system of linear equations has a unique solution when matrix A is invertible, and it has infinite solutions when matrix A is singular. Conversely, if the given vector b is not in the column space of matrix A, the system of linear equations has no solution. Based on quantum Gram-Schmidt orthogonalization and quantum QR decomposition, we propose a quantum algorithm to solve linear equations, judge whether the system of linear equations has a unique, infinite, or no solution, and give the solution when the system of linear equations has a unique solution.

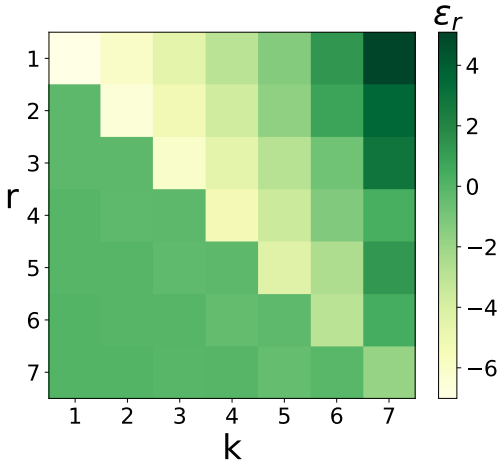

FIG. 8. The logarithmic relative error $\epsilon_r$of test data is calculated and plotted. k is the order of fitting polynomial function $f(x)$ and r is the order of fitted polynomial function $g(x)$. Underfitting regime is in the lower triangular area. Overfitting regime is in the upper triangular area. Appropriate fitting regime is on the diagonal.

Let us assume that the vector b is in the column space

TABLE I. Comparison of the performance of different methods for solving the problems of least squares fitting and linear equations

Problem	Method	Complexity	Limitations
Least squares fitting	Quantum QR decomposition	$O(N^3 \log N\kappa^2 \text{poly}\log(N\kappa/\epsilon)/\epsilon)$	None
	HHL algorithm [24]	$O\left(N^3 \log N\kappa^2/\epsilon^2, N^3\right)$[a]	Preprocess needed[b]
	CKS algorithm [25]	$O\left(N^2\kappa \text{poly}\log(N\kappa/\epsilon^2), N^3\right)$	Preprocesss needed[b]
	WZP algorithm [43]	$O(N\kappa^2 \text{poly}\log(N)\|\|A\|\|_F/\epsilon^2, N^3)$	Preprocess needed[b]
Solving linear equations	Quantum QR decomposition	$O(N^3 \log N\kappa^2 \text{poly}\log(N\kappa/\epsilon)/\epsilon)$	None
	HHL algorithm [24]	$O\left(N^3 \log N\kappa^2/\epsilon^2\right)$	Preprocess needed[c]
	CKS algorithm [25]	$O\left(N^2\kappa \text{poly}\log(N\kappa/\epsilon^2)\right)$	Preprocess needed[c]
	WZP algorithm [43]	$O(N\kappa^2 \text{poly}\log(N)\|\|A\|\|_F/\epsilon^2)$	Preprocess needed[c],$\|\|A\|\|_F$ in complexity

[a] Hereafter, we consider the complexity of the algorithms for general dense matrices, i.e., the sparity s in the complexity is all taken to be N.
[b] Need to calculate $A^\dagger A$ with classical time complexity $O(N^3)$.
[c] Need to expand A to matrix $(\mathbf{0}, A^\dagger; A, \mathbf{0})$ for non-Hermitian or non-square A.

of matrix A. Following the quantum circuit used in quantum Gram-Schmidt orthogonalization as in Fig. 2, we take the state preparation oracle

$$O_b|0\rangle = |b\rangle, \tag{67}$$

where $|b\rangle$ is the amplitude encoding of the vector $b/\parallel b \parallel$ as shown in Eq. (13). We take the Hamiltonian in the circuit as

$$H = \sum_{m=1}^{M} |a_m\rangle\langle a_m|, \tag{68}$$

via the qubitization. Where $|a_m\rangle$ is the amplitude encoding of the vector $a_m/\parallel a_m \parallel$ as shown in Eq. (13), and a_m is a vector formed by all matrix elements a_{nm} of the mth column of the matrix A. When $b \in span\{a_1, a_2, \cdots, a_M\}$, i.e

$$|b\rangle = \sum_{m=1}^{M} \langle a_m|b\rangle|a_m\rangle, \tag{69}$$

we have

$$\begin{aligned} H|b\rangle &= \sum_{m_1=1}^{M}\sum_{m_2=1}^{M} |a_{m_1}\rangle\langle a_{m_2}|a_{m_1}\rangle\langle a_{m_2}|b\rangle \\ &= \sum_{m_1=1}^{M}\sum_{m_2=1}^{M} \delta_{m_1 m_2}|a_{m_1}\rangle\langle a_{m_2}|b\rangle = |b\rangle. \end{aligned} \tag{70}$$

Thus, the output of the circuit for the quantum Gram-Schmidt orthogonalization algorithm is

$$|1\rangle|b\rangle, \tag{71}$$

and the probability that we measure 1 in the first register is 1. Conversely, if 0 is measured in the first register, then $b \notin span\{a_1, a_2, \cdots, a_M\}$, the system of linear equations has no solution. From Eq. (B14), we know that if we run the circuit for $(1/\epsilon)\ln(1/\epsilon)$ times and all the measurement results is 1, then we can infer that $b \in span\{a_1, a_2, \cdots, a_M\}$, as

$$P\left(1 - \sum_{n=1}^{m} |\langle a_i|b\rangle|^2 < \epsilon\right) > 1 - \epsilon. \tag{72}$$

By using the quantum Gram-Schmidt orthogonalization algorithm, we can know if b is or is not in the column space of matrix A. If b is not in the column space of matrix A, then linear equations have no solution. If b is in the column space of matrix A, then we perform quantum QR decomposition on the matrix A. When the matrix A is not full rank, then linear equations have infinite solutions. In this case, quantum QR decomposition algorithm will go into an endless loop. Conversely, if matrix A is a full-rank matrix, then linear equations have a unique solution. We can obtain the QR decomposition of the matrix A; then, the unique solution can be calculated classically with time complexity $O(MN)$ for

$$x = R^{-1}Q^T b. \tag{73}$$

There have been many quantum algorithms for solving the linear equations. We compare the performance of our algorithm and the previously best quantum algoritms [24, 25, 43] and summarize the resluts in Table I. It is noted that all quantum complexities listed include the complexity of the state readout procedure. It is shown that though the complexity of the proposed quantum QR decomposition based algorithm is not optimal, it has fewer limitations, robust to the case when A is a non-square matrix or the equation of $Ax = b$ has no solution or infinite solution.

We now further show the correctness of the quantum QR decomposition algorithm for solving the linear equations by an example of solving Laplace equation

$$\nabla^2 u = 0. \tag{74}$$

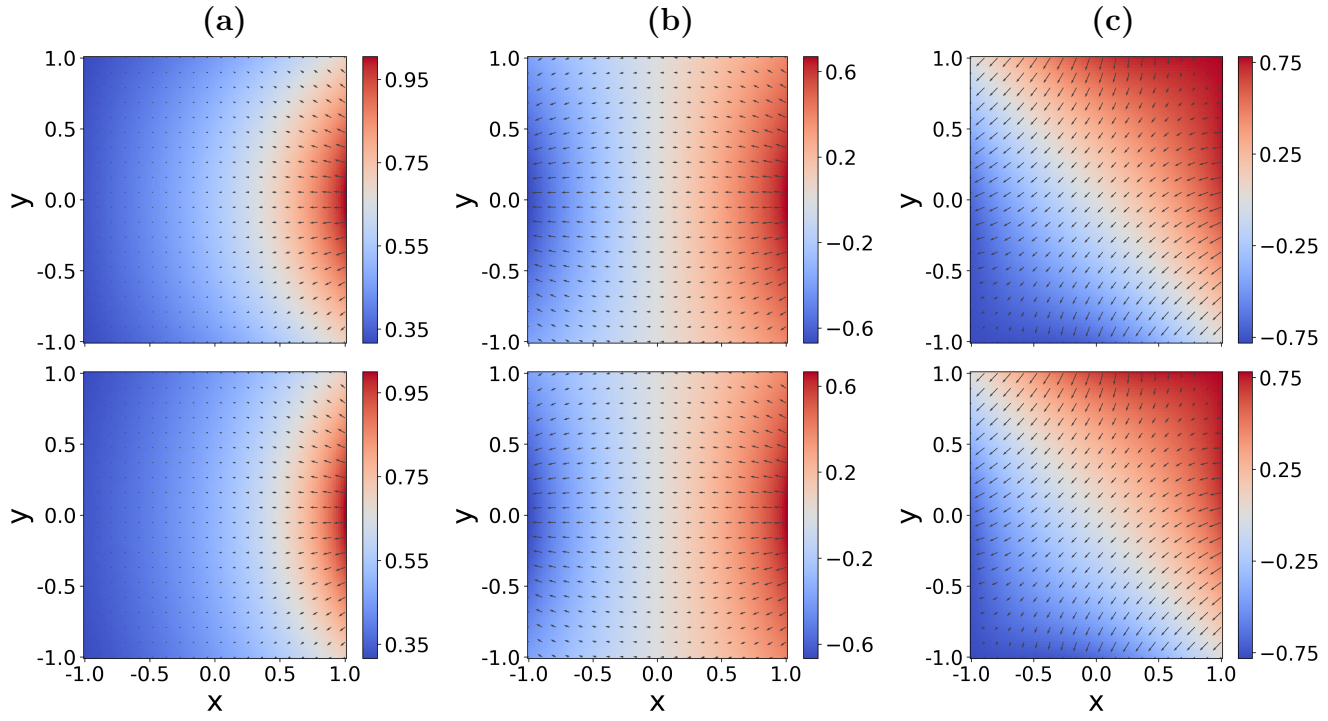

FIG. 9. The electric potential is plotted as a two-dimensional heatmap. The black arrows represent the electric field intensity, where the directions of arrows represent the directions of electric field intensity, and the length of arrows represent the magnitude of electric field intensity. The top three figures are the solutions from quantum QR decomposition based numerical simulations. The bottom three figures are the exact solutions, which agree with numerical solutions. (a) Electric monopole case, the relative error is 0.053. (b) Electric dipole case, the relative error is 0.057. (c) Electric quadrupole case, the relative error is 0.057.

At present, the most commonly used method to solve differential equations is the finite difference method. Using methods like the Runge-Kutta method, a differential equation is transformed into difference form, and then transformed into linear equations, which can be solved with quantum QR decomposition algorithm. We verify the performance of our algorithm on a two-dimensional Laplace equation with Dirichlet boundary condition.

We calculate the electric potential in vacuum. The electric potential ϕ satisfies Maxwell's equations

$$-\nabla^2\phi(x,y) = \frac{\rho(x,y)}{\epsilon_0}. \tag{75}$$

We calculate $\phi(x,y)$ on area $\Omega = \{(x,y)|-1 < x,y < 1\}$, where there is no charge. That is, Eq. (75) is changed to a Laplace equation

$$\nabla^2\phi(x,y) = 0 \tag{76}$$

for $\forall(x,y) \in \Omega$. We assume that the electric potential in this area is caused by electric monopole, electric dipole, and electric quadrupole, respectively. In the electric monopole case, we calculate the case that there is a positively charged particle at $(2,0)$. Thus, the boundary condition and exact solution for equation $\nabla^2\phi(x,y) = 0$ is

$$\phi(x,y) = \frac{kq}{\sqrt{(x-2)^2+y^2}}, \tag{77}$$

where k is the Coulomb constant and q is the quantity of electric charge. In the electric dipole case, we calculate the case that there is a positively charged particle at $(2,0)$ and a negatively charged particle at $(-2,0)$. Thus, the boundary condition and exact solution is

$$\phi(x,y) = \frac{kq}{\sqrt{(x-2)^2+y^2}} - \frac{kq}{\sqrt{(x+2)^2+y^2}}. \tag{78}$$

In the electric quadrupole case, we calculate the case that there are positively charged particles at $(2,0)$ and $(0,2)$, and there are negatively charged particles at $(-2,0)$ and $(0,-2)$. Thus, the boundary condition and exact solution is

$$\begin{aligned}\phi(x,y) =& \frac{kq}{\sqrt{(x-2)^2+y^2}} - \frac{kq}{\sqrt{(x+2)^2+y^2}} \\ &+ \frac{kq}{\sqrt{x^2+(y-2)^2}} - \frac{kq}{\sqrt{x^2+(y+2)^2}}.\end{aligned} \tag{79}$$

Without loss of generality, we take $k = q = 1$. We use quantum QR decomposition algorithm to solve the electric potential for the monopole case, dipole case, and quadrupole case respectively, and compare with the exact solution. The result is shown in Fig. 9. It is shown that the numerical results agree well with the exact solution, which shows the correctness of our quantum algorithm and its feasibility in solving differential equations.

C. Solving Eigenvalues

Our algorithm can also be applied to solve eigenvalues of a full rank matrix by replacing classical algorithm with quantum one. Classically, the eigenvalues of a full rank matrix can be solved by QR iteration algorithm as follows. Suppose we have a full rank matrix A and we want to compute its eigenvalues. We first take $A_1 = A$, then start to iterate the matrix. At the k-step (starting with $k = 1$), we compute the QR decomposition such that $A_k = Q_k R_k$ with an orthogonal matrix Q_k and an upper triangular matrix R_k, and obtain A_{k+1} as

$$A_{k+1} = R_k Q_k. \tag{80}$$

When A_k converges to the upper triangular matrix, the iteration is completed, and the principal diagonal elements are the eigenvalues of matrix A. The classical QR iteration algorithm is almost the most efficient method to solve all eigenvalues of matrices up to now. However, we find that this classical iteration QR algorithm can be replaced by the quantum one, in which classical QR decomposition in each iteration is replaced by the quantum one. We have proved that the computational complexity of the quantum QR decomposition algorithm is less than the classical one. Thus, the computational complexity of the quantum QR iteration algorithm is also less than the classical one.

To more concretely show the application of our algorithm to the problem, we now take examples of solving energy spectra of quantum Ising and Heisenberg models, which have extensively been studied in quantum many-body physics and condensed matter physics [44–48]. We will show that our quantum algorithm can be applied to solve these problems, which are not easy to be solved in the large scale systems by classical numerical one. We know that N-site open boundary one dimensional Ising model has the Hamiltonian

$$H = -h \sum_{n=1}^{N} \sigma_x^i - J \sum_{n=1}^{N-1} \sigma_z^i \sigma_z^{i+1}. \tag{81}$$

However, N-site open boundary one dimensional Heisenberg model has the Hamiltonian

$$H = -J \sum_{n=1}^{N-1} \left(\sigma_x^i \sigma_x^{i+1} + \sigma_y^i \sigma_y^{i+1} + \sigma_z^i \sigma_z^{i+1} \right). \tag{82}$$

We calculate the energy spectra of one dimensional Ising model and Heisenberg model taking the size to be 5 and all parameters to be 1. Using quantum QR decomposition algorithm, we perform iterative QR algorithm to obtain the eigenvalues of one dimensional Ising model and Heisenberg model.

We first study the convergence of calculated eigenvalues of Ising model. We take the diagonal elements of the matrix A_k obtained by the iterative QR algorithm at each step, take the lowest values of the diagonal elements as the calculated values of the lowest energy levels, and study how these values change with the increase of the number of iterative steps. The convergence of a few lowest energy levels through the iteration of QR process is shown in Fig. 10(a), where the dashed lines represent the exact values of Ising model energy spectra calculated by exact diagonalization method. It is shown that the calculated values of the few lowest energy levels converge to the exact values accurately. For an Ising model with the size N, the dimension of the Hamiltonian for Ising model is 2^N. It is shown in Fig. 10(a) that QR algorithm converges fast with $O(2^N)$ iterations.

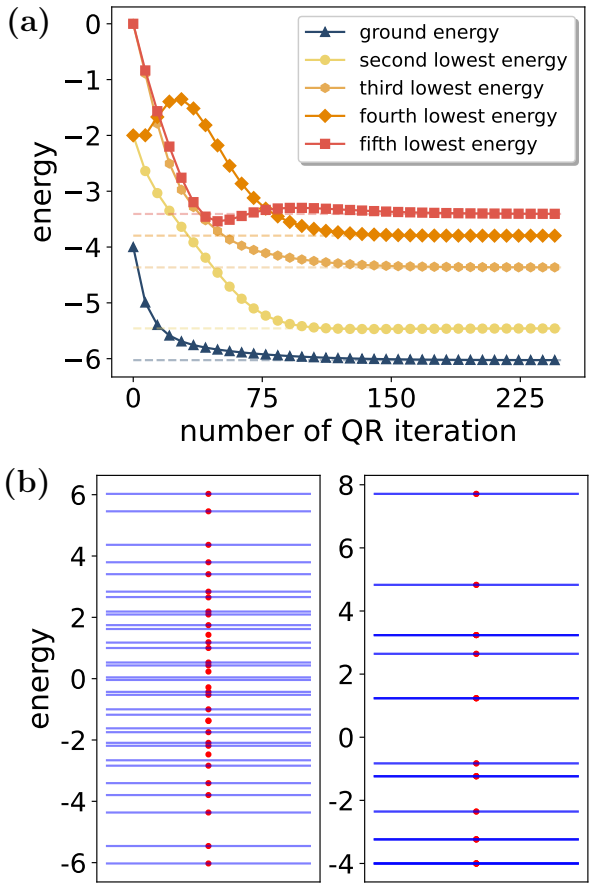

FIG. 10. (a) Convergence of few lowest eigenvalues of Ising model calculated by using quantum QR decomposition based QR algorithm. (b) Comparison of energy spectra of Ising model and Heisenberg model calculated by using quantum QR decomposition based QR algorithm with exact solutions of energy spectra.

Then we compare the calculated energy spectra of Ising model and Heisenberg model with exact energy spectra, which are calculated by exact diagonalization method. The results are shown in Fig. 10(b). The blue lines are the exact energy spectra of Ising model (left) and Heisenberg model (right), and the red dots are the energy spectra calculated by using quantum QR decomposition based QR algorithm. It is shown that the red dots agree with the blue lines for both Ising model and Heisenberg

model, showing that our quantum QR decomposition based QR algorithm can successfully calculate the energy spectra of both these two models accurately. Thus, our algorithm can be applied to eigenvalue finding with high accuracy.

However, it is worth mentioning that the time complexity of the quantum QR decomposition algorithm for solving all eigenvalues is at least $O(N^3 \mathrm{poly}\log N)$ in the parameter N. Compared to quantum phase estimation, whose time complexity is $O(N^2 \mathrm{poly}\log N)$, and other quantum algorithms [27, 49, 50] serving as the eigenvalue solver, the quantum QR decomposition based algorithm is less efficient. This is because the eigenvalue solver based on quantum QR decomposition is a natural extension of the classical QR algorithm. The main principle of the classical QR algorithm is unchanged. Therefore, its quantum advantage is limited.

VII. DISCUSSIONS AND CONCLUSIONS

In summary, we propose quantum algorithms for vector set orthogonal normalization and matrix QR decomposition respectively, which are basic tasks in matrix analysis and linear algebra with various applications in many fields. The proposed algorithms achieve polynomial speedup over the best previous quantum algorithms, scaling $O(N^3)$ in the system dimension N. It is noted that the classical algorithms also scale $O(N^3)$ in the system dimension N. Thus, the speedup of the proposed quantum algorithm over the classical algorithms is limited with a potential acceleration. This limitation is mainly brought by the Hamiltonian simulation step and the state readout step. Specifically, to simulate the LCU Hamiltonian $H = \sum_{l=1}^{d} V_l$ with the time t, the complexity is at least $O(dt)$. And to readout an N-dimensional quantum state, the complexity is at least $O(N)$. The linear scaling of the LCU Hamitonian and state readout contributes and leads to the $O(N^3)$ scaling in the final complexity. We look forward to new LCU Hamiltonian simulation tools with a sublinear scaling in d for $H = \sum_{l=1}^{d} V_l$, and state readout protocols with sublinear sampling comlexity in the dimension N of the quantum state. With these better subroutines, our algorithm can achieve a scaling lower than $O(N^3)$ in the system dimension N, which achieves a greater acceleration over the classical algorithms.

Also, the scaling $O(1/\epsilon^2)$ in the tolerant error ϵ of our quantum QR decomposition algorithm may be further improved. We use repetitive measurements to estimate the inner product of two quantum states, leading to this $O(1/\epsilon^2)$ scaling. Quantum amplitude estimation methods can be applied to estimate quantum inner product [51–53], which may improve the scaling in tolerant error ϵ to $O(1/\epsilon)$.

In this work, we also study several applications of our quantum algorithms, including the linear least squares regression, solving linear equations and eigenvalues of the matrix. The correctness and efficiency of our algorithms are also proved with detailed proof and numerical simulations. Due to the need of the QRAM oracle, the proposed algorothms cannot be realized with NISQ devices temporarily. But when the fault-tolerant quantum computers are available, we believe that our algorithms have broad applications for solving various important problems.

ACKNOWLEDGMENTS

This work was supported by Innovation Program for Quantum Science and Technology (Grant No. 2021ZD0300201).

Appendix A: Derivation of Quantum Phase Estimation

The quantum circuit for quantum phase estimation is given in Fig. 11 when we limit the number of qubits in the first register to be 1. Suppose the Hamiltonian H has a spectral decomposition

$$H = \sum_{n=1}^{N} \lambda_n |u_n\rangle\langle u_n|, \tag{A1}$$

where N is the dimension of H and each λ_n can be 0. Then $\{|u_1\rangle, |u_2\rangle, \cdots, |u_N\rangle\}$ is a complete set, so the input state of the second register $|u\rangle_s$ can be written as

$$|u\rangle_s = \sum_{n=1}^{N} \langle u_n|u\rangle_s |u_n\rangle \tag{A2}$$

So the input state of the circuit is

$$|0\rangle_f |u\rangle_s = \sum_{n=1}^{N} |0\rangle_f \langle u_n|u\rangle_s |u_n\rangle. \tag{A3}$$

After first Hadamard gate, the state of system is

$$\sum_{n=1}^{N} \frac{|0\rangle + |1\rangle}{\sqrt{2}} \langle u_n|u\rangle_s |u_n\rangle. \tag{A4}$$

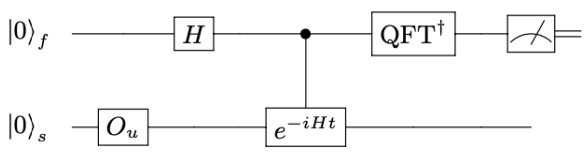

FIG. 11. Quantum circuit for quantum phase estimation when the size of first register is 1. To avoid confusion, the top H represents Hadamard gate and the bottom H is Hamiltonian.

As

$$e^{-iHt}|u_n\rangle = e^{-i\lambda_n t}|u_n\rangle, \tag{A5}$$

so after the controlled-e^{-iHt} gate, the state of system is

$$\sum_{n=1}^{N}\langle u_n|u\rangle_s \frac{|0\rangle_f + e^{-i\lambda_n t}|1\rangle_f}{\sqrt{2}}|u_n\rangle. \tag{A6}$$

After the final Hadamard gate, the state of the system is

$$\sum_{n=1}^{N}\langle u_n|u\rangle_s \frac{(|0\rangle_f + |1\rangle_f) + e^{-i\lambda_n t}(|0\rangle_f - |1\rangle_f)}{2}|u_n\rangle. \tag{A7}$$

Rewrite the equation.

$$\sum_{n=1}^{N}\langle u_n|u\rangle_s \left(\frac{1+e^{-i\lambda_n t}}{2}|0\rangle_f + \frac{1-e^{-i\lambda_n t}}{2}|1\rangle_f\right)|u_n\rangle. \tag{A8}$$

If the eigenvalues of Hamiltonian H satisfy

$$\begin{aligned}\lambda_n &= 1, \quad n = 1, 2, \cdots, k\\ \lambda_n &= 0, \quad n = k+1, \cdots, N\end{aligned} \tag{A9}$$

as in the quantum Gram-Schmidt process and the evolution time t of the Hamiltonian H is

$$t = \pi, \tag{A10}$$

The final state of the quantum circuit then is

$$\begin{aligned}&\sum_{n=1}^{k}\langle u_n|u\rangle_s \left(\frac{1+e^{-i\pi}}{2}|0\rangle_f + \frac{1-e^{-i\pi}}{2}|1\rangle_f\right)|u_n\rangle\\ &+ \sum_{n=k+1}^{N}\langle u_n|u\rangle_s \left(\frac{1+e^{-i\pi*0}}{2}|0\rangle_f + \frac{1-e^{-i\pi*0}}{2}|1\rangle_f\right)|u_n\rangle,\end{aligned} \tag{A11}$$

i.e,

$$|1\rangle_f\left(\sum_{n=1}^{k}\langle u_n|u\rangle_s|u_n\rangle\right) + |0\rangle_f\left(\sum_{n=k+1}^{N}\langle u_n|u\rangle_s|u_n\rangle\right). \tag{A12}$$

So the probability that measured result from the first register is 0 is

$$1 - \sum_{n=1}^{k}|\langle u_n|b\rangle|^2. \tag{A13}$$

When we measure 0 from the first register, then the state of the second register is then

$$|\psi\rangle = \frac{\sum_{n=k+1}^{N}\langle u_n|b\rangle|u_n\rangle}{\| \sum_{n=k+1}^{N}\langle u_n|b\rangle|u_n\rangle \|}, \tag{A14}$$

which satisfies

$$|\langle\psi|u_n\rangle|^2 = 0, \forall n = 1, 2, \cdots, k. \tag{A15}$$

.

Appendix B: Proof of Lemmas in the section IV B

1. Proof of Lemma 3

Proof. First we have

$$\begin{aligned}\| A\otimes B \|_2^2 &= \left(\sqrt{\lambda_{max}\left(A\otimes B\right)^\dagger\left(A\otimes B\right)}\right)^2\\ &= \lambda_{max}\left(A\otimes B\right)^\dagger\left(A\otimes B\right)\\ &= \lambda_{max}\left(A^\dagger A\otimes B^\dagger B\right)\\ &= \lambda_{max}\left(A^\dagger A\right)\cdot\lambda_{max}\left(A^\dagger A\right)\\ &= \| A \|_2^2\cdot\| B \|_2^2.\end{aligned} \tag{B1}$$

Therefore $\| A\otimes B \|_2 = \| A \|_2\cdot\| B \|_2$ is proved.

Suppose the qubit number of the first register is 1. Then the exact and errant QPE unitary is

$$U_{exact} = \left(QFT^\dagger\otimes I\right)(C-U)(H\otimes I) \tag{B2}$$

$$U_{real} = \left(QFT^\dagger\otimes I\right)(C-\exp(-iHt))(H\otimes I) \tag{B3}$$

with $\| U-\exp(-iHt) \| < \epsilon_0$.

Define $\Delta U = (C-U) - (C-\exp(-iHt))$. From Lemma 2, we get the following equation.

$$\begin{aligned}\| \Delta U \| &= \| (C-U) - (C-\exp(-iHt)) \|\\ &= \| |1\rangle\langle 1|\otimes(U-\exp(-iHt)) \|\\ &= \| |1\rangle\langle 1| \| \| U-\exp(-iHt) \|\\ &< \epsilon_0.\end{aligned} \tag{B4}$$

Therefore,

$$\begin{aligned}&\| U_{real}-U_{exact} \|\\ &\le \| \left(QFT^\dagger\otimes I\right) \| \| (H\otimes I) \| \| \Delta U \|\\ &= \| \Delta U \|\\ &< \epsilon_0\end{aligned} \tag{B5}$$

.

Therefore, we prove Lemma 3. □

2. Proof of Lemma 4

In Algorithm 3, we generate a series states successively, so we use mathematical induction to prove Lemma 4.

Proof. Firstly, as we take $|u_1\rangle \equiv |a_1\rangle$, So $span\{a_1\} = span\{u_1\}$. It is clear that $|u_t\rangle$ and $|a_m\rangle$ is the amplitude encoding of $N-$dimension vector u_t and a_m.

$$|a_m\rangle = \sum_{n=1}^{N}\frac{a_{nm}}{\| a_m \|}|n-1\rangle. \tag{B6}$$

$$|u_t\rangle = \sum_{n=1}^{N}\frac{u_{nt}}{\| u_t \|}|n-1\rangle. \tag{B7}$$

Suppose after the first k steps of quantum Gram-Schmidt process, we have generated $\{|u_1\rangle, |u_2\rangle, \cdots, |u_t\rangle\}$ for $\{a_1, a_2, \cdots, a_k\}$, satisfying $span\{a_1, a_2, \cdots, a_k\} = span\{u_1, u_2, \cdots, u_t\}$ with probability larger than $1 - k\epsilon/M$. It is clear that t is not always equal to k. It is because $\{a_1, a_2, \cdots, a_k\}$ is not always a set of linearly independent vectors. Therefore, $k \geq t$.

In the $(k+1)$th step of the quantum Gram-Schmidt process, the input state of the second register is $|a_{k+1}\rangle$ which can be written as

$$|a_{k+1}\rangle = \sum_{n=1}^{t} \langle u_n|a_{k+1}\rangle |u_n\rangle + c|\psi\rangle, \tag{B8}$$

where we denote c to be a real number

$$c = \sqrt{1 - \sum_{n=1}^{t} \| \langle u_n|a_{k+1}\rangle \|^2}. \tag{B9}$$

After running the quantum circuit of the $(k+1)$th step of quantum Gram-Schmidt process, if we measure the first register and the result is 0, we can read out the $(k+1)$th state $|u_{t+1}\rangle$ from the second register as $|u_{t+1}\rangle \equiv |\psi\rangle$. So

$$a_{k+1} \in span\{u_1, u_2, \cdots, u_t, u_{t+1}\} \tag{B10}$$

and

$$span\{a_1, a_2, \cdots, a_k, a_{k+1}\} = span\{u_1, u_2, \cdots, u_t, u_{t+1}\} \tag{B11}$$

When c as in Eq. (B8) is small, we may need to run the quantum circuit many times to measure 0 in the first register. When $a_{k+1} \in span\{a_1, a_2, \cdots, a_k\}$, then c is 0 so we cannot measure 0 from the first register. However, we cannot know this information in advance, so we need to run the quantum circuit repeatedly and make measurements to determine whether it is the linearly dependent case.

Denote p the probability of measuring 0 in the first register. Then

$$p = c^2 = 1 - \sum_{n=1}^{t} |\langle u_n|a_{k+1}\rangle|^2 \tag{B12}$$

Denote κ as the conditional number of the matrix $A' = (a_1/||a_1||, a_2/||a_2||, \cdots, a_M/||a_M||)$, which is defined as the ratio of the maximum singular value $\sigma_{\max}$ and the minimum non-zero singular value $\sigma_{\min}$ of A'

$$\kappa = \frac{\sigma_{\max}(A')}{\sigma_{\min}(A')}. \tag{B13}$$

An important result that will be used for the proof below is that, suppose $a_{k+1} \notin span\{a_1, a_2, \cdots, a_k\}$, then

$$p > \frac{1}{\kappa^2}. \tag{B14}$$

We prove Eq. (B14) in Appendix. B 3.

We use hypothesis testing to determine whether it is the linearly dependent or independent case. Two hypotheses are listed below:

$$\begin{aligned} H_0 &: a_{k+1} \notin span\{a_1, a_2, \cdots, a_k\}, \\ H_1 &: a_{k+1} \in span\{a_1, a_2, \cdots, a_k\}. \end{aligned} \tag{B15}$$

The proposed hypothesis testing method is, we repeatedly run the quantum circuit and measure the ancillary qubit in the computational basis for at most $\kappa^2 \ln(M/\epsilon)$ times, where M is the number of input vectors. If in one iteration, the measurement outcome is 0, then we have hypothesis H_0 verified immediately. If all $\kappa^2 \ln(M/\epsilon)$ outcomes are all 1, then we conclude the hypothesis H_1 is correct, with successful probability larger than $1 - \epsilon/M$. Next we prove the correctness of the hypothesis tesing method.

Denote after W iteations, the total of W measurement outcomes $\overrightarrow{x} = (x_1, x_2, \cdots, x_W)$. Follow our method, we have $x_1 = x_2 = \cdots = x_{W-1} = 1$. Following the Neyman-Pearson theorem, i.e., if the function $L(\overrightarrow{x})$ satisfies

$$L(\overrightarrow{x}) = \frac{p(\overrightarrow{x}; H_1)}{p(\overrightarrow{x}; H_0)} > \gamma, \tag{B16}$$

then we conclude that H_1 is correct, else we conclude that H_0 is correct. We take the threshold γ to be

$$\gamma = \left(\frac{1}{\epsilon}\right)^{1+\frac{1}{2\kappa^2}}. \tag{B17}$$

And in our case, the function of $p(\overrightarrow{x}; H_0)$ and $p(\overrightarrow{x}; H_1)$ is

$$\begin{aligned} p(\overrightarrow{x}; H_0) &= (1-p)^{W-1+x_W} p^{x_W} \\ p(\overrightarrow{x}; H_1) &= \delta(x_W = 1) \end{aligned} \tag{B18}$$

Thus, if after W iteration, we have $x_1 = x_2 = \cdots = x_{W-1} = 1$ and $x_W = 0$, we have

$$L(\overrightarrow{x}) = \frac{\delta(x_W = 1)}{(1-p)^{W-1}p} = 0 < \gamma, \tag{B19}$$

which indicates that H_0 is correct, i.e., $a_{k+1} \notin span\{a_1, a_2, \cdots, a_k\}$. If after $\kappa^2 \ln(M/\epsilon)$ iterations, and we still have $x_1 = x_2 = \cdots = x_W = 1$, then

$$\begin{aligned} L(\overrightarrow{x}) = \frac{1}{(1-p)^W} &= (1-p)^{-\kappa^2 \ln(M/\epsilon)} \\ &> \left(1 - \frac{1}{\kappa^2}\right)^{-\kappa^2 \ln(M/\epsilon)} \\ &= \exp\left(-\kappa^2 \ln\left(\frac{M}{\epsilon}\right) \ln\left(1 - \frac{1}{\kappa^2}\right)\right) \\ &> \exp\left(-\kappa^2 \ln\left(\frac{M}{\epsilon}\right) \left(-\frac{1}{\kappa^2} - \frac{1}{2\kappa^4}\right)\right) \\ &= \left(\exp\left(\ln\left(\frac{M}{\epsilon}\right)\right)\right)^{1+\frac{1}{2\kappa^2}} = \gamma, \end{aligned} \tag{B20}$$

thus in this case we conclude that hypothesis H_1 is correct. The false alert probability p_{FA} satisfies

$$\begin{aligned} p_{FA} &= \int_{\overrightarrow{x}:L(\overrightarrow{x})>\gamma} p(\overrightarrow{x};H_0)d\overrightarrow{x} \\ &= \sum_{x_1=0}^{1}\cdots\sum_{x_W=0}^{1} p(\overrightarrow{x};H_0)\delta(L(\overrightarrow{x})>\gamma) \\ &= p(\overrightarrow{1};H_0)\delta(L(\overrightarrow{1})>\gamma) \\ &= (1-p)^{\kappa^2\ln(M/\epsilon)} \\ &< (1-p)^{-\ln(\epsilon/M)/p} < (1-p)^{\ln(\epsilon/M)/\ln(1-p)} = \epsilon/M. \end{aligned} \tag{B21}$$

Thus, we have proved that the probability that we miskate the case $a_{k+1}\notin span\{a_1,a_2,\cdots,a_k\}$ to be $a_{k+1}\in span\{a_1,a_2,\cdots,a_k\}$ is smaller than ϵ/M. In this case, $span\{a_1,a_2,\cdots,a_{k+1}\}\neq span\{u_1,u_2,\cdots,u_t\}$. Elsewise we have $span\{a_1,a_2,\cdots,a_{k+1}\}=span\{u_1,u_2,\cdots,u_{t+1}\}$ for the linearly independent case and $span\{a_1,a_2,\cdots,a_{k+1}\}=span\{u_1,u_2,\cdots,u_t\}$ for the linealy dependent case. Thus, if $span\{a_1,a_2,\cdots,a_k\}=span\{u_1,u_2,\cdots,u_t\}$ for the first k steps, then $span\{a_1,a_2,\cdots,a_{k+1}\}=span\{u_1,u_2,\cdots,u_t\}$ or $span\{a_1,a_2,\cdots,a_{k+1}\}=span\{u_1,u_2,\cdots,u_{t+1}\}$ holds true with probability larger than $1-\epsilon/M$.

To sum up, after the first k steps, if $span\{a_1,a_2,\cdots,a_k\}=span\{u_1,u_2,\cdots,u_t\}$ holds with probability larger than $1-k\epsilon/M$, then we prove after the first $k+1$ steps, the linear subspace spanned by $\{u_1,u_2,\cdots,u_t\}$ or $\{u_1,u_2,\cdots,u_{t+1}\}$ is equal to $span\{a_1,a_2,\cdots,a_{k+1}\}$ with probability larger than $(1-\epsilon/M)(1-k\epsilon/M)$, which is larger than $1-(k+1)\epsilon/M$. Thus after all M steps, i.e., after the whole algorithm is accomplished, the probability that $span\{a_1,a_2,\cdots,a_M\}=span\{u_1,u_2,\cdots,u_T\}$ is larger than $1-\epsilon$. The proof of Lemma 4 is completed. □

3. Proof of Eq. (B14)

Proof. For a set of input vectors $\{a_1,a_2,\cdots,a_M\}$ with rank T, there exists a set of linearly independent vectors $\{a_{k_1},a_{k_2},\cdots,a_{k_T}\}$, where each $k_t\in\{1,2,\cdots,M\}, t=1,\cdots,T$, and

$$span\{a_1,a_2,\cdots,a_M\}=span\{a_{k_1},a_{k_2},\cdots,a_{k_T}\}. \tag{B22}$$

We define the set of the linearly independent complete set $\{a_{k_1},a_{k_2},\cdots,a_{k_T}\}$ in the following way. Denote p_{k+1} as the probability that zero is the measurement outcome in the first qubit register in the $(k+1)$th step of the quantum Gram-Schimdt process. Then when $a_{k+1}\in\{a_1,a_2,\cdots,a_k\}$, $p_{k+1}=0$. And when $a_{k+1}\notin\{a_1,a_2,\cdots,a_k\}$, we have

$$p_{k+1}=\parallel|a_{k+1}\rangle-\mathcal{P}_k|a_{k+1}\rangle\parallel^2, \tag{B23}$$

where $\mathcal{P}_k$ is the projector onto the linear subspace $span\{a_1,a_2,\cdots,a_k\}$ and $|a_{k+1}\rangle$ is the quantum state that encodes a_{k+1}. From $k=0$ to $k=M-1$, if $a_{k+1}\notin\{a_1,a_2,\cdots,a_M\}$, then we add a_{k+1} to linearly independent complete set, and we finally get the set of $\{a_{k_1},a_{k_2},\cdots,a_{k_T}\}$. We define a matrix $C=(a_{k_1}/||a_{k_1}||,a_{k_2}/||a_{k_2}||,\cdots,a_{k_T}/||a_{k_T}||)$, which is a full column-rank matrix. We define the conditional number of matrix $A'=(a_1/||a_1||,a_2/||a_2||,\cdots,a_M/||a_M||)$ and $C=(a_{k_1}/||a_{k_1}||,a_{k_2}/||a_{k_2}||,\cdots,a_{k_T}/||a_{k_T}||)$ to be $\kappa_{A'}$ and κ_C, respectively. The conditional number of a matrix is denoted to be the ratio of the maximum singular value and the minimum non-zero singluar value of the matrix. There is a transformation matrix $\mathcal{T}$ between A' and C, as

$$A'\mathcal{T}=C. \tag{B24}$$

$\mathcal{T}$ is a matrix with size $M\times T$. The (t,k_t)th item of matrix $\mathcal{T}$ is one, $\forall t=1,2,\cdots,T$ and all other items are zero. We now calculate the conditional number of C.

$$\kappa_C=\frac{\sigma_{\max}(C)}{\sigma_{\min}(C)}=\frac{\lambda_{\max}(C^\dagger C)}{\lambda_{\min}(C^\dagger C)}. \tag{B25}$$

Following the Courant-Fischer minimax theorem, we have

$$\begin{aligned} \lambda_{\max}(C^\dagger C) &= \min_{dim(V)=T}\max_{x\in V,||x||_2=1}\parallel Cx\parallel_2^2 \\ &= \min_{dim(V)=T}\max_{x\in V,||x||_2=1}\parallel A'\mathcal{T}x\parallel_2^2 \\ &\leq \min_{dim(V)=T}\max_{x\in V,||x||_2=1}\sigma_{\max}(A')\parallel\mathcal{T}x\parallel_2^2 \\ &= \sigma_{\max}(A'), \end{aligned} \tag{B26}$$

and

$$\begin{aligned} \lambda_{\min}(C^\dagger C) &= \min_{dim(V)=1}\max_{x\in V,||x||_2=1}\parallel Cx\parallel_2^2 \\ &= \min_{dim(V)=1}\max_{x\in V,||x||_2=1}\parallel A'\mathcal{T}x\parallel_2^2 \\ &\geq \min_{dim(V)=1}\max_{x\in V,||x||_2=1}\sigma_{\min}(A')\parallel\mathcal{T}x\parallel_2^2 \\ &= \sigma_{\min}(A'). \end{aligned} \tag{B27}$$

Thus, we have

$$\kappa_C=\frac{\lambda_{\max}(C^\dagger C)}{\lambda_{\min}(C^\dagger C)}\leq\frac{\sigma_{\max}(A')}{\sigma_{\min}(A')}=\kappa_{A'}. \tag{B28}$$

Next, we bound the conditional number of C with $\{p_{k_1},p_{k_2},\cdots,p_{k_T}\}$. As the matrix C is a full rank matrix, we can perform the QR decomposition on the matrix C. Denote $C=Q_cR_c$ to be the QR decomposition of matrix C. Following the Algorithm. 2, the diagonal terms of matrix R_c is

$$\begin{aligned} (R_c)_{k_tk_t} &= \parallel a_{k_t}/||a_{k_t}||-\mathcal{P}_{k_{t-1}}a_{k_t}/||a_{k_t}||\parallel \\ &= \parallel|a_{k_t}\rangle-\mathcal{P}_{k_{t-1}}|a_{k_t}\rangle\parallel \\ &= \sqrt{p_{k_t}},\quad\forall t=1,2,\cdots,T. \end{aligned} \tag{B29}$$

Thus, the conditional number of R_c satisfies,

$$\kappa_{R_c} = \frac{\sigma_{\max}(R_c)}{\sigma_{\min}(R_c)} > \frac{\max_t (R_c)_{k_t k_t}}{\min_t (R_c)_{k_t k_t}} = \frac{\max_t \sqrt{p_{k_t}}}{\min_t \sqrt{p_{k_t}}} = \frac{1}{\min_t \sqrt{p_{k_t}}}. \quad \text{(B30)}$$

C is a full column-rank matrix with $C = Q_c R_c$, thus $Q_c^\dagger Q_c = I$. And

$$C^\dagger C = R_c^\dagger Q_c^\dagger Q_c R_c = R_c^\dagger R_c. \quad \text{(B31)}$$

Thus

$$\kappa_C = \frac{\lambda_{\max}(C^\dagger C)}{\lambda_{\min}(C^\dagger C)} = \frac{\lambda_{\max}(R_c^\dagger R_c)}{\lambda_{\min}(R_c^\dagger R_c)} = \kappa_{R_c}. \quad \text{(B32)}$$

Combining Eq. (B28), Eq. (B30), and Eq. (B32) together, we have

$$\frac{1}{\min_t \sqrt{p_{k_t}}} < \kappa_{R_c} = \kappa_C \leq \kappa_{A'}. \quad \text{(B33)}$$

Summarily, for each $p \in \{p_1, p_2, \cdots, p_M\}$ satisfying $p > 0$, we have proved that

$$\frac{1}{\sqrt{p}} < \kappa_{A'}, \quad \text{(B34)}$$

i.e.,

$$p > \frac{1}{\kappa_{A'}^2}. \quad \text{(B35)}$$

The proof of Eq. (B14) is completed. □

4. Proof of Lemma 5

In Algorithm 3, we generate a series states successively, so we again use mathematical induction to prove Lemma 5.

Proof. Firstly, as we take $|u_1\rangle \equiv |a_1\rangle$, So u_1 itself is a normalized vector. It is clear that $|u_t\rangle$ and $|a_m\rangle$ is the amplitude encoding of $N-$dimension vector u_t and a_m.

$$|a_m\rangle = \sum_{n=1}^{N} \frac{a_{nm}}{\| a_m \|} |n-1\rangle. \quad \text{(B36)}$$

$$|u_t\rangle = \sum_{n=1}^{N} \frac{u_{nt}}{\| u_t \|} |n-1\rangle. \quad \text{(B37)}$$

Suppose in the first k steps of quantum Gram-Schmidt process, we have generated $\{u_1, u_2, \cdots, u_t\}$ for $\{a_1, a_2, \cdots, a_k\}$, satisfying $u_{t_1}^\dagger u_{t_2} = O(\epsilon)$ for $\forall t_1 \neq t_2$. It is clear that t is not always equal to k. It is because $\{a_1, a_2, \cdots, a_k\}$ is not always a set of linearly independent vectors so $k \geq t$. We want to prove the constructed u_{t+1} is nearly orthogonal to all previous constructed vectors $\{u_1, u_2, \cdots, u_t\}$ with the errant Hamiltonian simulation step of the QPE circuit, i.e.,

$$u_{t+1}^\dagger u_{t'} = O(\epsilon), \forall t' = 1, 2, \cdots, t \quad \text{(B38)}$$

Considering the $(k+1)$th step of quantum Gram-Schmidt process, we assume that $a_{k+1} \notin \{a_1, a_2, \cdots, a_k\}$. For the case that $a_{k+1} \in \{a_1, a_2, \cdots, a_k\}$, in this step we do not construct a new $|u_{t+1}\rangle$ so it is a trival case and follow the mathematical induction immediately. So we only consider the case that $a_{k+1} \notin \{a_1, a_2, \cdots, a_k\}$. In the $(k+1)$th step, the $(t+1)$th state $|u_{t+1}\rangle$ is constructed based on previous constrcuted states $\{|u_1\rangle, |u_2\rangle, \cdots, |u_t\rangle\}$ and a_{k+1}. $|a_{k+1}\rangle$ which can be written as

$$|a_{k+1}\rangle = \sum_{n=1}^{t} \langle u_n | a_{k+1} \rangle |u_n\rangle + c|\psi\rangle, \quad \text{(B39)}$$

where we denote c to be a real number

$$c = \sqrt{1 - \sum_{n=1}^{t} \| \langle u_n | a_{k+1} \rangle \|^2}. \quad \text{(B40)}$$

Suppose the first qubit register is measured and the outcome is zero, and we construct a new state $|u_{t+1}\rangle$ in the $(k+1)$th step, we want to prove that

$$|\langle u_{t+1} | u_t' \rangle| = O(\epsilon), \forall t' = 1, 2, \cdots, t, \quad \text{(B41)}$$

given the condition that

$$|\langle u_{t_1} | u_{t_2} \rangle| = O(\epsilon), \forall t_1, t_2 = 1, \cdots, t, t_1 \neq t_2. \quad \text{(B42)}$$

We first define three states for the case that the first qubit register is measured to be zero: the quantum state $|u_{exact}\rangle$ which is the quantum state in the second register with ideal Hamiltonian simulation, the quantum state $|u_{real}\rangle$ which is the quantum state in the second register with errant Hamiltonian simulation and exact readout, and $|u_{t+1}\rangle$ which is the actual state we readout with errant Hamiltonian simulation and imperfect state tomography. Now we start our proof.

We first calculate the distance between $|u_{exact}\rangle$ and $|u_{real}\rangle$. The ideal output of the circuit assuming there is no error in the Hamiltonian simulation step is

$$|\psi_{exact}\rangle = U_{exact} \left(|0\rangle \otimes |a_{k+1}\rangle \right) = \sum_{n=1}^{t} \langle u_n | a_{k+1} \rangle |1\rangle |u_n\rangle + c|0\rangle|\psi\rangle \quad \text{(B43)}$$

But with errant Hamilonian simulation step, the actual output state is $|\psi_{real}\rangle$.

$$|\psi_{real}\rangle = U_{real} \left(|0\rangle \otimes |a_{k+1}\rangle \right). \quad \text{(B44)}$$

From Lemma 3 we know, the error of QPE circuit is bounded by

$$\| U_{real} - U_{exact} \| < \epsilon_0. \tag{B45}$$

So

$$\begin{aligned} \| |\psi_{real}\rangle - |\psi_{exact}\rangle \| =& \| (U_{real} - U_{exact}) |0\rangle \otimes |a_{k+1}\rangle \| \\ \leq& \| U_{real} - U_{exact} \| \\ <& \epsilon_0. \end{aligned} \tag{B46}$$

When we measure the first register and get result 0, we have,

$$\begin{aligned} \frac{|0\rangle\langle 0| \otimes I|\psi_{exact}\rangle}{\sqrt{\langle\psi_{exact}|0\rangle\langle 0| \otimes I|\psi_{exact}\rangle}} &= |0\rangle \otimes |u_{exact}\rangle \\ \frac{|0\rangle\langle 0| \otimes I|\psi_{real}\rangle}{\sqrt{\langle\psi_{real}|0\rangle\langle 0| \otimes I|\psi_{real}\rangle}} &= |0\rangle \otimes |u_{real}\rangle. \end{aligned} \tag{B47}$$

Denote the probability that 0 is measured from the first register $p(0)_{ideal}$ and $p(0)_{real}$ in the ideal case and in the real case, i.e.,

$$p(0)_{exact} = \langle\psi_{exact}|0\rangle\langle 0| \otimes I|\psi_{exact}\rangle, \tag{B48}$$

$$p(0)_{real} = \langle\psi_{real}|0\rangle\langle 0| \otimes I|\psi_{real}\rangle. \tag{B49}$$

So,

$$\begin{aligned} & |p(0)_{exact} - p(0)_{real}| \\ =& |\langle\psi_{real}|0\rangle\langle 0| \otimes I|\psi_{real}\rangle - \langle\psi_{exact}|0\rangle\langle 0| \otimes I|\psi_{exact}\rangle| \\ <& 2\epsilon_0 \end{aligned} \tag{B50}$$

Thus,

$$\begin{aligned} & \| |u_{exact}\rangle - |u_{real}\rangle \| \\ =& \| |0\rangle|u_{exact}\rangle - |0\rangle|u_{real}\rangle \| \\ =& \| \frac{|0\rangle\langle 0| \otimes I|\psi_{exact}\rangle}{\sqrt{p(0)_{exact}}} - \frac{|0\rangle\langle 0| \otimes I|\psi_{real}\rangle}{\sqrt{p(0)_{real}}} \| \end{aligned} \tag{B51}$$

Expand $1/\sqrt{p(0)_{real}}$ as

$$\begin{aligned} \frac{1}{\sqrt{p(0)_{real}}} &= \frac{1}{\sqrt{p(0)_{exact}}} \frac{1}{\sqrt{1 + \frac{p(0)_{real} - p(0)_{exact}}{p(0)_{exact}}}} \\ &= \frac{1}{\sqrt{p(0)_{exact}}} \left(1 + \frac{p(0)_{real} - p(0)_{exact}}{2p(0)_{exact}}\right) \\ &\quad + O\left(\frac{\epsilon_0^2}{p(0)_{exact}^{3/2}}\right) \end{aligned} \tag{B52}$$

So,

$$\begin{aligned} & \| |u_{exact}\rangle - |u_{real}\rangle \| \\ \leq& \| \frac{|0\rangle\langle 0| \otimes I\left(|\psi_{exact}\rangle - |\psi_{real}\rangle\right)}{\sqrt{p(0)_{exact}}} \| \\ & + |\frac{p(0)_{real} - p(0)_{exact}}{2p(0)_{exact}}| * \| \frac{|0\rangle\langle 0| \otimes I|\psi_{real}\rangle}{\sqrt{p(0)_{exact}}} \| \\ & + O\left(\frac{\epsilon_0^2}{p(0)_{exact}^{3/2}}\right) * \| |0\rangle\langle 0| \otimes I|\psi_{real}\rangle \| \\ <& \frac{\epsilon_0}{\sqrt{p(0)_{exact}}} + \frac{2\epsilon_0}{2 * p(0)_{exact}^{3/2}} + O\left(\frac{\epsilon_0^2}{p(0)_{exact}^{3/2}}\right) \\ <& \frac{2\epsilon_0}{p(0)_{exact}^{3/2}} \end{aligned} \tag{B53}$$

From Eq. (B14) we know $1/p(0)_{exact} < \kappa^2$, thus

$$\| |u_{exact}\rangle - |u_{real}\rangle \| < 2\epsilon_0\kappa^3. \tag{B54}$$

Thus when we take $\epsilon_0 = \epsilon/2\kappa^3$, we have

$$\| |u_{exact}\rangle - |u_{real}\rangle \| < \epsilon. \tag{B55}$$

For the state readout procedure, as discussed in the main text, it is demand that

$$|\langle u_{t+1}|u_{real}\rangle| > 1 - \epsilon^2. \tag{B56}$$

And for state $|u_{exact}\rangle$, when the condition of Eq. (B42) is satisfied, we have $|\langle u_n|u_{exact}\rangle| = O(\epsilon)$, $\forall n = 1, 2, \cdots, t$. Combining this with Eq. (B55), we have

$$\begin{aligned} |\langle u_n|u_{real}\rangle| &\leq |\langle u_n|u_{exact}\rangle| + \| |u_{exact}\rangle - |u_{real}\rangle \| \\ &= O(\epsilon), \forall n = 1, 2, \cdots, t. \end{aligned} \tag{B57}$$

And combining this equation with Eq. (B56) we have

$$|\langle u_n|u_{t+1}\rangle| = O(\epsilon), \forall n = 1, 2, \cdot, t. \tag{B58}$$

Thus we have proved that in the $(k+1)$th step of the quantum Gram-Schmidt process, the constructed $|u_{t+1}\rangle$ with errant Hamiltonian simulation step is nearly orthogonal to all previous constructed states, i.e.,

$$|\langle u_n|u_{t+1}\rangle_{real}| = O(\epsilon), \quad \forall n \leq t. \tag{B59}$$

Therefore, in the $(k+1)th$ step of quantum Gram-Schmidt process, a state $|u_{t+1}\rangle$ is constructed, encoding a vector u_{t+1}. And $\{u_1, u_2, \cdots, u_t, u_{t+1}\}$ satisfies

$$u_{t_1}^\dagger u_{t_2} = O(\epsilon), \forall t_1 \neq t_2. \tag{B60}$$

Thus, the proof of Lemma 5 is completed. □

[1] R. A. Horn and C. R. Johnson, *Matrix analysis* (Cambridge university press, 2012).
[2] R. Bhatia, *Matrix analysis*, Vol. 169 (Springer Science & Business Media, 2013).
[3] P. Businger and G. H. Golub, Numerische Mathematik **7**, 269 (1965).
[4] S. J. Leon, Å. Björck, and W. Gander, Numerical Linear Algebra with Applications **20**, 492 (2013).
[5] B. N. Parlett, Computing in science & engineering **2**, 38 (2000).
[6] W. Zheng, C. Zou, and L. Zhao, in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 2 (IEEE, 2004) pp. 403–406.
[7] D. Wang, H. Zhang, R. Liu, X. Liu, and J. Wang, Neurocomputing **173**, 845 (2016).
[8] I. Aizenberg, A. Luchetta, and S. Manetti, Soft Computing **16**, 563 (2012).
[9] J.-K. Zhang, A. Kavcic, and K. M. Wong, IEEE Transactions on Information Theory **51**, 154 (2005).
[10] J. Cheng and M. D. Sacchi, Geophysics **81**, V89 (2016).
[11] A. Srinivasa, International Journal of Engineering Science **60**, 1 (2012).
[12] C. R. Goodall, (1993).
[13] L. Giraud, J. Langou, and M. Rozloznik, Computers & Mathematics with Applications **50**, 1069 (2005).
[14] L. Giraud, J. Langou, M. Rozložník, and J. v. d. Eshof, Numerische Mathematik **101**, 87 (2005).
[15] W. M. Gentleman, Linear Algebra and its Applications **10**, 189 (1975).
[16] R. P. Feynman, Optics News **11**, 11 (1985).
[17] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (Cambridge university press, 2010).
[18] J. Preskill, Quantum **2**, 79 (2018).
[19] E. Gibney, Nature **574**, 461 (2019).
[20] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, *et al.*, Science **370**, 1460 (2020).
[21] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, Science **351**, 1068 (2016).
[22] P. W. Shor, in *Proceedings 35th annual symposium on foundations of computer science* (IEEE, 1994) pp. 124–134.
[23] A. Prakash, *Quantum algorithms for linear algebra and machine learning* (University of California, Berkeley, 2014).
[24] A. W. Harrow, A. Hassidim, and S. Lloyd, Physical Review Letters **103**, 150502 (2009).
[25] A. M. Childs, R. Kothari, and R. D. Somma, SIAM Journal on Computing **46**, 1920 (2017).
[26] P. Rebentrost, M. Mohseni, and S. Lloyd, Physical Review Letters **113**, 130503 (2014).
[27] S. Lloyd, M. Mohseni, and P. Rebentrost, Nature Physics **10**, 631 (2014).
[28] G. H. Low, T. J. Yoder, and I. L. Chuang, Physical Review A **89**, 062315 (2014).
[29] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, Physical Review X **8**, 021050 (2018).
[30] K. Zhang, M.-H. Hsieh, L. Liu, and D. Tao, Physical Review Research **3**, 043095 (2021).
[31] V. Giovannetti, S. Lloyd, and L. Maccone, Physical Review Letters **100**, 160501 (2008).
[32] G. Ma, H. Li, and J. Zhao, Quantum Information Processing **19**, 1 (2020).
[33] V. Giovannetti, S. Lloyd, and L. Maccone, Physical Review A **78**, 052310 (2008).
[34] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, Mathematics of Computation **30**, 772 (1976).
[35] Å. Björck, Linear Algebra and Its Applications **197**, 297 (1994).
[36] G. H. Low and I. L. Chuang, Quantum **3**, 163 (2019).
[37] P. W. Shor, SIAM review **41**, 303 (1999).
[38] L. Hales and S. Hallgren, in *Proceedings 41st Annual Symposium on Foundations of Computer Science* (IEEE, 2000) pp. 515–525.
[39] A. Miessen, P. J. Ollitrault, F. Tacchino, and I. Tavernelli, Nature Computational Science **3**, 25 (2023).
[40] J. Haah, A. W. Harrow, Z. Ji, X. Wu, and N. Yu, in *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing* (2016) pp. 913–925.
[41] A. Anshu and S. Arunachalam, Nature Reviews Physics **6**, 59 (2024).
[42] L. Zhao, Z. Zhao, P. Rebentrost, and J. Fitzsimons, Quantum Machine Intelligence **3**, 21 (2021).
[43] L. Wossnig, Z. Zhao, and A. Prakash, Physical review letters **120**, 050502 (2018).
[44] U. Schollwöck, Reviews of modern physics **77**, 259 (2005).
[45] U. Schollwöck, Annals of physics **326**, 96 (2011).
[46] R. Orus and G. Vidal, Physical Review B—Condensed Matter and Materials Physics **78**, 155117 (2008).
[47] R. Orús, Annals of physics **349**, 117 (2014).
[48] H. Fehske, R. Schneider, and A. Weisse, *Computational many-particle physics*, Vol. 739 (Springer, 2007).
[49] 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.
[50] G. H. Low and I. L. Chuang, Physical review letters **118**, 010501 (2017).
[51] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Contemporary Mathematics **305**, 53 (2002).
[52] T. Giurgica-Tiron, I. Kerenidis, F. Labib, A. Prakash, and W. Zeng, Quantum **6**, 745 (2022).
[53] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, npj Quantum Information **7**, 52 (2021).