IEEE Transactions on Quantum Engineering

Received June 24, 2021; revised December 11, 2021; accepted December 19, 2021; date of publication January 4, 2022; date of current version January 21, 2022.

Digital Object Identifier 10.1109/TQE.2021.3140152

A Low-Complexity Quantum Principal Component Analysis Algorithm

CHEN HE[1] (Member, IEEE), JIAZHEN LI[1], WEIQI LIU[1], JINYE PENG[1], AND Z. JANE WANG[2] (Fellow, IEEE)

[1]School of Information Science and Technology, Northwest University, Xi'an 710069, China
[2]Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada

Corresponding authors: Chen He; Weiqi Liu (e-mail:chenhe@nwu.edu.cn; vickylwq1991@nwu.edu.cn).

This work was supported in part by the National Natural Science Foundation of China under Grant 6200138 and Grant 61701401, in part by the Key Research and Development Project of Shaanxi Province under Grant 2021KWZ-07, and in part by the Nova Program of Science and Technology of Shaanxi Province under Grant 2019KJXX-061.

ABSTRACT In this article, we propose a low-complexity quantum principal component analysis (qPCA) algorithm. Similar to the state-of-the-art qPCA, it achieves dimension reduction by extracting principal components of the data matrix, rather than all components of the data matrix, to quantum registers, so that the samples of measurement required can be reduced considerably. Both our qPCA and Lin's qPCA are based on quantum singular-value thresholding (QSVT). The key of Lin's qPCA is to combine QSVT, and modified QSVT is to obtain the superposition of the principal components. The key of our algorithm, however, is to modify QSVT by replacing the rotation-controlled operation of QSVT with the controlled-NOT operation to obtain the superposition of the principal components. As a result, this small trick makes the circuit much simpler. Particularly, the proposed qPCA requires three phase estimations, while the state-of-the-art qPCA requires five phase estimations. Since the runtime of qPCA mainly comes from phase estimations, the proposed qPCA achieves a runtime of roughly 3/5 of that of the state of the art. We simulate the proposed qPCA on the IBM quantum computing platform, and the simulation result verifies that the proposed qPCA yields the expected quantum state.

INDEX TERMS Quantum computing, quantum principal component analysis (qPCA), quantum singular-value threshold.

I. INTRODUCTION

Principal component analysis (PCA) is widely employed in signal processing and machine learning for dimension reduction and has a time complexity of $O(mN^2)$, where m is the number of features and N indicates the number of samples, which is usually greater than m. When the dimension of the data becomes very large, the classical PCA becomes nontractable. There are some algorithms to make the problem tractable for larger matrices, for example, Lanczos algorithm [1], quantum-inspired algorithm [2], and quantum PCA (qPCA) [3]. In this article, we continue to discuss the qPCA, which is in dimension exponentially faster than the classical PCA. According to [4], the complexity of the qPCA is about $O(\frac{1}{\sqrt{r}\epsilon}\text{poly}(\log Nm))$, where r is the rank of the matrix and ϵ is a relative error for the algorithm.

The qPCA is first proposed by Lloyd *et al.* [3] in 2014, which outputs the quantum state containing all the eigenvalues and eigenvectors of the data, and the top-t components (principal components) are obtained by sampling. However, this method contains all the r components of A; it requires a lot of samples to obtain the principal components. To avoid this disadvantage, Lin *et al.* [5] proposed an improved qPCA, as shown in Fig. 1, which yields a quantum state containing the components with top-t ($t \ll r$) largest eigenvalues only. For instance, a matrix $A_0 \in \mathbb{C}^{p\times q}$ has a quantum state of

$$|\psi_{A_0}\rangle = \frac{1}{\sqrt{N_1}}\sum_{k=1}^{r}\sigma_k|u_k\rangle|v_k\rangle \tag{1}$$

where $N_1 = \sum_{k=1}^{r}\sigma_k^2$, r is the rank of the matrix, A_0 and σ_k are the singular values of A_0, and u_k and v_k are the left and right singular vectors, respectively. With an input $|\psi_{A_0}\rangle$, the output of [5] is given by

$$|\psi'_A\rangle = \frac{1}{\sqrt{N_2}}\sum_{k=1}^{t}\sigma_k|\lambda_k\rangle|u_k\rangle|v_k\rangle \tag{2}$$

where $A = A_0A_0^\dagger = \sum_{k=1}^{r}\lambda_k u_k u_k^T$ and $N_2 = \sum_{k=1}^{t}\sigma_k^2$.

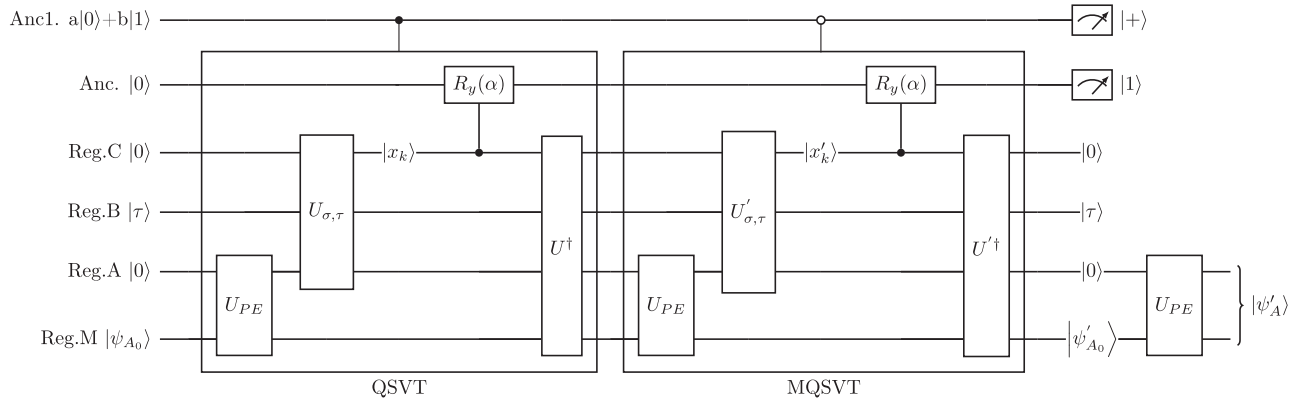

FIGURE 1. Quantum circuit of Lin's qPCA [5], which consists mainly two parts: QSVT [7] and MQSVT [5]. The register M is initialized by $|\psi_{A_0}\rangle$, the register A is initialized by $|0\rangle$ and stores eigenvalues $|\lambda_k\rangle$ obtained by U_{PE}, the register B saves the threshold $|\tau\rangle$, the register C is initialized by $|0\rangle$ and stores $|x_k\rangle$ and $|x'_k\rangle$ obtained by operations $U_{\sigma,\tau}$ and $U'_{\sigma,\tau}$, respectively, the register Anc. is initialed by $|0\rangle$, and the top register Anc1. is initialized by the superposition of $|0\rangle$ and $|1\rangle$. In this circuit, the threshold τ here is to filter out the small σ_ks, $x_k = (1-\tau/\sigma_k)_+$ and $x'_k = (1+\tau/\sigma_k)$; $U^\dagger_{\text{PE}}$ is the inverse of the phase estimation U_{PE}; $U_{\sigma,\tau}$ is an operation to filter the singular values, which is smaller than the threshold τ; $U^\dagger_{\sigma,\tau}$ is the inverse of $U_{\sigma,\tau}$; $R_y(\alpha)$ is a rotation operation about the y-axis with α; and $U^\dagger$ is the inverse operation of $U_{\sigma,\tau}U_{\text{PE}}$. Therefore, the circuit of the qPCA in [5] requires five phase estimations, i.e., the operations U_{PE}, $U^\dagger = U^\dagger_{\sigma,\tau}U^\dagger_{\text{PE}}$, U_{PE}, $U'^\dagger = U'^\dagger_{\sigma,\tau}U^\dagger_{\text{PE}}$, and U_{PE}.

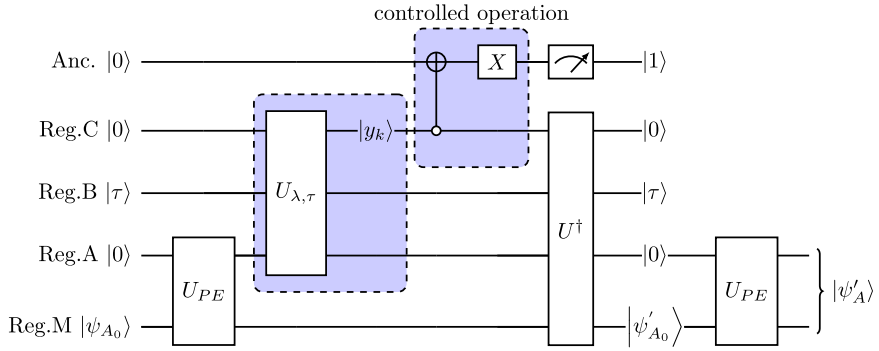

FIGURE 2. Quantum circuit of the proposed qPCA. In this circuit, the register M is initialized by $|\psi_{A_0}\rangle$, the register A is initialized by $|0\rangle$ and stores the eigenvalues $|\lambda_k\rangle$ obtained by U_{PE}, the register B is initialized by the threshold $|\tau\rangle$, the register C is initialized by $|0\rangle$ and saves $|y_k\rangle$ obtained by $U_{\lambda,\tau}$, and the top register Anc. saves an ancillary qubit, which is also initialized by $|0\rangle$. The threshold τ, here, is to filter out small λ_ks, $y_k = (1-\tau/\lambda_k)_+$, U_{PE} is a phase estimation operation, $U^\dagger_{\text{PE}}$ is the inverse of U_{PE}, $U_{\lambda,\tau}$ is an operation to filter the eigenvalues smaller than the threshold τ, $U^\dagger_{\lambda,\tau}$ is the inverse of $U_{\lambda,\tau}$, CU is a controlled-not operation, and $U^\dagger$ is the inverse operation of $U_{\lambda,\tau}U_{\text{PE}}$. Therefore, the circuit of the proposed qPCA requires three phase estimations, i.e., the operations U_{PE}, $U^\dagger = U^\dagger_{\lambda,\tau}U^\dagger_{\text{PE}}$, and U_{PE}. Compared with Lin's qPCA, the runtime of the proposed qPCA is roughly 3/5 of that of Lin's.

Actually, the qPCA in [5] can be divided into three steps. The first step is to convert (1) to a linear combination, i.e.,

$$|\psi_{A_0}\rangle = \frac{1}{\sqrt{N_2}}\sum_{k=1}^{t}\sigma_k|u_k\rangle|v_k\rangle + \frac{1}{\sqrt{N_3}}\sum_{k=t+1}^{r}\sigma_k|u_k\rangle|v_k\rangle \quad (3)$$

where $N_2 = \sum_{k=1}^{t}\sigma_k^2$ and $N_3 = \sum_{k=t+1}^{r}\sigma_k^2$. The second step is a measurement that makes the superposition state collapses to $\frac{1}{\sqrt{N_2}}\sum_{k=1}^{t}\sigma_k|u_k\rangle|v_k\rangle$, i.e., the first term of (3). The third step is to perform the phase estimation and obtain the top-t eigenvalues in (2).

To obtain (3) in the first step, as shown in Fig. 1, Lin's algorithm [5] combines quantum singular-value thresholding (QSVT) and modified QSVT (MQSVT), each of which requires two phase estimations. In the third step requiring one phase estimation, Lin's algorithm requires five phase estimations in total for the circuit runtime. In this article, as shown in Fig. 2, we obtain (3) by modifying QSVT via replacing the rotation-controlled operation with the controlled-NOT operation; as a result, the MQSVT is avoided. This small trick makes the total number of phase estimations down to three. As the runtime of the phase estimations dominates that of other operations [6], the proposed qPCA achieves a runtime of roughly 3/5 of that of [5].

The rest of this article is organized as follows. In Section II, we review the qPCA algorithm proposed by Lin *et al.* In Section III, we propose and validate the low-complexity qPCA algorithm via theoretical analysis. In Section IV, we analytically study the complexity and the accuracy of the proposed qPCA. In Section V, we simulate and validate the proposed algorithm on IBM Quantum Experience (QX). Finally, Section VI concludes this article.

II. STATE OF THE ART

The qPCA proposed by Lin *et al.* [5], as shown in Fig. 1, consists of three steps. The first step contains two major operations: QSVT and MQSVT, i.e.,

$$U_{\{Lin,Step1\}} = (|0><0| \otimes U_{\text{QSVT}} + |1><1| \otimes U_{\text{MQSVT}}) \tag{4}$$

where

$$U_{\{\text{QSVT}\}} = (I \otimes U_{\text{PE}}^{\dagger})(I \otimes U_{\sigma,\tau}^{\dagger} \otimes I)(R_y(\alpha) \otimes I) \times (I \otimes U_{\sigma,\tau} \otimes I)(I \otimes U_{\text{PE}}) \tag{5}$$

$$U_{\{\text{MQSVT}\}} = (I \otimes U_{\text{PE}}^{\dagger})(I \otimes U_{\sigma,\tau}^{\prime\dagger} \otimes I)\left(R_y\left(\frac{\alpha}{2}\right) \otimes I\right) \times (I \otimes U_{\sigma,\tau}^{\prime} \otimes I)(I \otimes U_{\text{PE}}) \tag{6}$$

where U_{PE} is the phase estimation, $U_{\sigma,\tau}$ is the operation to filter the smaller singular values, $U'_{\sigma,\tau}$ is the operation to add the smaller singular values, and $\dagger$ represents the inverse of an operation. Moreover, $R_y(\alpha)$ and $R_y(\alpha/2)$ are rotation operations defined in (20). The first step is to convert the r components of $|\psi_{A_0}\rangle$ in (1) to a $(t + (r - t))$ linear combination in (3). The second step is to make a projection measurement on Ancl. and Anc. in Fig. 1 to obtain $\frac{1}{\sqrt{N_2}} \sum_{k=1}^{t} \sigma_k |u_k\rangle|v_k\rangle$. The third step is a phase estimation for obtaining the large singular values, which can be expressed as

$$U_{\{Lin,Step3\}} = I \otimes U_{\text{PE}}. \tag{7}$$

The corresponding circuit of QSVT in (5) is shown on the left-hand side of Fig. 1. When the measurement of top register is 1 in the QSVT circuit, the rest registers collapse to

$$|\psi_s\rangle = \frac{1}{\sqrt{N_2}} \sum_{k=1}^{t} (\sigma_k - \tau)|u_k\rangle|v_k\rangle. \tag{8}$$

QSVT can be decomposed into the following building blocks: the initialization of quantum registers, the phase estimation U_{PE} for extracting the eigenvalues λ_ks, the unitary operation $U_{\sigma,\tau}$ for determining whether or not the singular values are greater than the threshold τ, and the operations $U^{\dagger}$ containing the reverse operations $U_{\text{PE}}^{\dagger}$ and $U_{\sigma,\tau}^{\dagger}$ for removing unnecessary registers.

A. INITIALIZATION

The preparation of the initial quantum state in quantum registers is shown in Fig. 1

$$|0\rangle^{\text{Anc}}|0\rangle^{\text{C}}|\tau\rangle^{\text{B}}|0\rangle^{\text{A}}|\psi_{A_0}\rangle^{\text{M}}. \tag{9}$$

The register A, register C, and the top ancillary register Anc. are all initialized by $|0\rangle$, the register B is initialized by the threshold $|\tau\rangle$, and the register M is initialized by the quantum state $|\psi_{A_0}\rangle$. Then, using Gram–Schmidt decomposition, we have [6]

$$|\psi_{A_0}\rangle = \sum_{i=1}^{p}\sum_{j=1}^{q} a_{ij}|i\rangle|j\rangle = \sum_{k=1}^{r} \sigma_k |u_k\rangle|v_k\rangle \tag{10}$$

and prepare $|\psi_{A_0}\rangle$ by the quantum random access memory (QRAM) [8].

Time complexity: The time complexity of this step is mainly determined by the time complexity of preparing $|\psi_{A_0}\rangle$. Note that $|\psi_{A_0}\rangle$ can be prepared by the QRAM. In the QRAM input model, the time to run the circuit is $O(\log(pq))$; therefore, the time complexity of this step is $O(\log(pq))$.

Space complexity: Since the size of the data matrix A_0 is $p \times q$, we can infer that the number of qubits required for register M, which stores A_0, is $O(\log(pq))$.

B. PHASE ESTIMATION $I \otimes U_{PE}$

The purpose of the phase estimation U_{PE} is to extract eigenvalues λ_ks to the quantum register A. Suppose $|\psi_{A_0}\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$, where $\mathcal{H}_1$ and $\mathcal{H}_2$ are two Hilbert spaces; the mixed state $\rho_{A_0A_0^{\dagger}}$ can be obtained by the partial trace of $|\psi_{A_0}\rangle\langle\psi_{A_0}|$ over $\mathcal{H}_2$ [9]

$$\rho = \rho_{A_0A_0^{\dagger}} = \text{tr}_2(|\psi_{A_0}\rangle\langle\psi_{A_0}|) = \frac{1}{N_1}\sum_{k=1}^{r} \lambda_k |u_k\rangle\langle u_k|. \tag{11}$$

Via phase estimation, the eigenvalues λ_ks of the data matrix A_0 can be estimated as [10]

$$|0\rangle^A|\psi_{A_0}\rangle^M \xrightarrow{U_{\text{PE}}} \frac{1}{\sqrt{N_1}}\sum_{k=1}^{r} \sigma_k |\lambda_k\rangle^A|u_k\rangle|v_k\rangle \tag{12}$$

where

$$U_{\text{PE}} = (\text{QFT}^{\dagger} \otimes I)(I \otimes e^{2\pi i \rho t_0})(H \otimes I) \tag{13}$$

and QFT is the quantum Fourier transform and $\text{QFT}^{\dagger}$ denotes the inverse of QFT.

Time complexity: Now, we study the time complexity of U_{PE}. The Hamilton simulation $e^{2\pi i \rho t_0}$, with $t_0 = O(\kappa/\epsilon)$ (κ is the condition number of the matrix A and ϵ is the error of the Hamilton simulation), can be simulated by using $O(t_0^2\epsilon^{-1})$ copies of $e^{2\pi i \rho \Delta t}$ for the time slice Δt[6]. Based on (11), the time complexity of each of the above copies is determined by the time complexity of preparing $|\psi_{A_0}\rangle$, which is $O(\log(pq))$. Therefore, the time complexity of the Hamilton simulation is $O(t_0^2\epsilon^{-1}\log(pq)) = O(\kappa^2\epsilon^{-3}\log(pq))$. However, suppose that κ is the condition number of the matrix A and $\lambda_k \geq 1/\kappa$ for all k; the complexity of the QFT is only $O(n^2) = O((\log\kappa)^2)$ or $O(n\log n) = O(\log\kappa\log(\log\kappa))$ (the space complexity below will explain $n = O(\log\kappa)$), which is much smaller than that of the Hamilton simulation. Therefore, the time complexity for obtaining U_{PE} is $O(t_0^2\epsilon^{-1}\log(pq))$. The state-of-the-art quantum phase estimation (QPE) [11] has a lower time complexity of $O(t_0 + \log(1/\epsilon))$. Even for this complexity, QPE is the dominating procedure of the qPCA algorithm in terms of the running time.

Space complexity: In general, the number of qubits required for storing $|\lambda_k\rangle$s is $O(n) = O(\log(\kappa))$ [7], where κ is the condition number of A_0.

C. UNITARY OPERATION $I \otimes U_{\sigma,\tau} \otimes I$

The purpose of this step is to filter out small singular values in the quantum register C. The way to achieve this is to set $U_{\sigma,\tau}$, which is a unitary transformation from $\lambda_k = \sigma_k^2$ to $x_k = (1 - \frac{\tau}{\sqrt{\sigma_k^2}})_+$.

Now, let us see how to obtain $U_{\sigma,\tau}$ as the transformation from σ_k^2 to x_k. We first set the intermediate variable $z_k = z(\sigma_k^2) = 1/\sqrt{\sigma_k^2}$. Because there are only two arithmetic operations in quantum circuits, i.e., multiplication and addition, we employ Newton's iteration to make a mathematical approximation for the division of the reciprocal. Then, we apply the Newton method to the function $f(z_k) = 1/z_k^2 - \sigma_k^2$, and we can obtain the iteration function

$$z_k^{(i+1)} = z_k^{(i)} - \frac{f(z_k^{(i)})}{f'(z_k^{(i)})} = \frac{1}{2}(3z_k^{(i)} - (z_k^{(i)})^3\sigma_k^2). \tag{14}$$

After obtaining z_k by the Newton's iteration, whose each iteration can be obtained by additions or multiplications, the unitary operation $U_{\lambda,\tau}$ is obtained by $x_k = 1 - \tau z_k$. This step can be represented as

$$\begin{aligned} &|0\rangle^C \frac{1}{\sqrt{N_1}} \sum_{k=1}^{r} \sigma_k |\lambda_k\rangle|u_k\rangle|v_k\rangle \\ &\xrightarrow{U_{\sigma,\tau}} \frac{1}{\sqrt{N_1}} \sum_{k=1}^{r} \sigma_k |x_k\rangle^C |\lambda_k\rangle|u_k\rangle|v_k\rangle. \end{aligned} \tag{15}$$

Time complexity: The time complexity of $U_{\sigma,\tau}$ is determined by the Newton's iteration. The initial state of the iteration (32) is $z_k^{(0)} = 2^{\frac{p-1}{2}}$, where p satisfies $2^{-p} \le \sigma_k < 2^{1-p}$. The number of iterations for obtaining $U_{\sigma,\tau}$ is approximately $s = O(\log n)$ [12], where n is the number of qubits in the register storing $|\lambda_k\rangle$. For each of the above iterations, $O(\text{poly}(n))$ additions or multiplications are required. Therefore, the time complexity of obtaining $U_{\sigma,\tau}$ by the Newton's iteration is $O(\text{poly}(n) \times s) = O(\text{poly}(\log \kappa) \log(\log \kappa))$. Compared with the time complexity of the phase estimation, which is $O(\kappa^2 \epsilon^{-3} \log(pq))$, the time complexity of the Newton's iteration can be ignored.

Space complexity: In general, the registers storing $|x_k\rangle$, $|\lambda_k\rangle$, $|\tau\rangle$ require the same number of qubits. Therefore, in this step, we can infer that the numbers of qubits required in registers B and C are both $O(\log(\kappa))$.

Accuracy: There are two types of errors incurred by the Newton's iteration: 1) the expected error e_s, brought by the finite number (i.e., s) of iterations, and 2) the average truncation error e_n, brought by the finite number of available digits. We define $\hat{\lambda_k} = \lambda_k - e_n$; then, we have

$$\begin{aligned} z_k^{(i)} - \frac{1}{\sqrt{\hat{\lambda_k}}} &= \frac{1}{2}(3z_k^{(i-1)} - \hat{\sigma_k}^2 (z_k^{(i-1)})^3) - \frac{1}{\sqrt{\hat{\sigma_k}^2}} \\ &= -\frac{1}{2}(z_k^{(i-1)} - \frac{1}{\sqrt{\hat{\sigma_k}^2}})^2 \sqrt{\hat{\sigma_k}^2}(z_k^{(i-1)}\sqrt{\hat{\sigma_k}^2} + 2) \\ &\le 0 \end{aligned} \tag{16}$$

and the Newton iteration convergence is quadratic; the error e_s can be written as

$$\begin{aligned} e_s^{(i)} &= \left| z_k^{(i)} - \frac{1}{\sqrt{\hat{\lambda_k}}} \right| = \frac{1}{2} e_s^{(i-1)2} \sqrt{\hat{\sigma_k}^2} \left(z_k^{(i-1)} \sqrt{\hat{\sigma_k}^2} + 2 \right) \\ &\le \frac{3}{2} e_s^{(i-1)2} \sqrt{\hat{\sigma_k}^2} \le \frac{2}{3\sqrt{\hat{\sigma_k}^2}} \left(\frac{3}{2} \sqrt{\hat{\sigma_k}^2} e^{(0)} \right)^{2^s}. \end{aligned} \tag{17}$$

In addition, according to the initial state $z_k^{(0)}$, we have

$$\begin{aligned} \sqrt{\hat{\sigma_k}^2} e_s^{(0)} &= \left| z_k^{(0)} - \frac{1}{\sqrt{\hat{\sigma_k}^2}} \right| \sqrt{\hat{\sigma_k}^2} = \left| z_k^{(0)} \sqrt{\hat{\sigma_k}^2} - 1 \right| \\ &= 1 - z_k^{(0)} \sqrt{\hat{\sigma_k}^2} = 1 - 2^{p-1} \sqrt{\hat{\sigma_k}^2} \\ &\le 1 - 2^{p-1} \times 2^{-p} = \frac{1}{2}. \end{aligned} \tag{18}$$

Therefore, the total error $e_s + e_n$ is given by

$$\begin{aligned} e_s + e_n &\le \frac{2}{3\sqrt{\hat{\sigma_k}^2}} \left(\frac{3}{2} \sqrt{\hat{\sigma_k}^2} e^{(0)} \right)^{2^s} + s2^{-n} \\ &\le \frac{2}{3\sqrt{\hat{\sigma_k}^2}} \left(\frac{3}{4} \right)^{2^s} + s2^{-n} \\ &\le \frac{2\sqrt{\kappa}}{3} \left(\frac{3}{4} \right)^{2^s} + s2^{-n}. \end{aligned} \tag{19}$$

D. ROTATION OPERATION $R_y(\alpha) \otimes I$

The purpose of this step is to employ the ancillary qubit to perform $R_y(\alpha)$, where

$$R_y(\alpha)|x_k\rangle = \sin \frac{x_k \alpha}{2} |1\rangle + \cos \frac{x_k \alpha}{2} |0\rangle. \tag{20}$$

This procedure can be expressed as

$$\begin{aligned} &\frac{1}{\sqrt{N_1}} |0\rangle^{\text{Anc}} \sum_{k=1}^{r} \sigma_k |x_k\rangle|\lambda_k\rangle|u_k\rangle|v_k\rangle \xrightarrow{R_y(\alpha)} \\ &\frac{1}{\sqrt{N_1}} \left[\sum_{k=1}^{t} (\sin(x_k\alpha)|1\rangle + \cos(x_k\alpha)|0\rangle) \sigma_k |x_k\rangle|\lambda_k\rangle|u_k\rangle|v_k\rangle \right. \\ &\qquad \left. + \sum_{k=t+1}^{r} \sigma_k |0\rangle|x_k\rangle|\lambda_k\rangle|u_k\rangle|v_k\rangle \right]. \end{aligned} \tag{21}$$

Time complexity: The number of quantum gates of $R_y(\alpha)$ is $O(d)$, where d is the number of qubits of the register storing the eigenvalues $|\lambda_k\rangle$. Therefore, the time cost in this step is very small.

Space complexity: In this step, the top register Anc. requires only one qubit.

E. UNITARY REVERSE OPERATION $I \otimes U^\dagger$

The purpose of this step is to remove the unnecessary registers storing $|x_k\rangle$ and $|\lambda_k\rangle$. Since $U^\dagger = U_{\mathrm{PE}}^\dagger U_{\sigma,\tau}^\dagger$, we have

$$\begin{aligned}
&\frac{1}{\sqrt{N_1}}\left[\sum_{k=1}^{t}(\sin(x_k\alpha)|1\rangle+\cos(x_k\alpha)|0\rangle)\sigma_k|x_k\rangle|\lambda_k\rangle|u_k\rangle|v_k\rangle\right.\\
&\qquad\left.+\sum_{k=t+1}^{r}\sigma_k|0\rangle|x_k\rangle|\lambda_k\rangle|u_k\rangle|v_k\rangle\right]\xrightarrow{U^\dagger}\\
&\frac{1}{\sqrt{N_1}}\left[\sum_{k=1}^{t}(\sin(x_k\alpha)|1\rangle+\cos(x_k\alpha)|0\rangle)\sigma_k|u_k\rangle|v_k\rangle\right.\\
&\qquad\left.+\sum_{k=t+1}^{r}\sigma_k|0\rangle|u_k\rangle|v_k\rangle\right]. \qquad (22)
\end{aligned}$$

Time complexity: In Sections II-B and II-C, it has been shown that the time complexities for obtaining U_{PE} and $U_{\sigma,\tau}$ are $O(t_0^2\epsilon^{-1}\log(pq))$ and $O(\mathrm{poly}(\log\kappa)\log(\log\kappa))$ ($t_0 = O(\kappa/\epsilon)$), respectively. Therefore, the time complexity for obtaining $U^\dagger$ is $O(t_0^2\epsilon^{-1}\log(pq))$.

The MQSVT changes $x_k = (1-\tau/\sigma_k)_+$ to $x_k' = 1+\tau/\sigma_k$ and changes the parameter α in the rotation operation $R_y(\alpha)$ to $\alpha/2$ in the rotation operation $R_y(\frac{\alpha}{2})$. According to the analysis of QSVT above, we can obtain the output of the MQSVT

$$\begin{aligned}
&\frac{1}{\sqrt{N_1}}\left[\sum_{k=1}^{t}\sigma_k\left(\sin\frac{x_k'\alpha}{2}|1\rangle+\cos\frac{x_k'\alpha}{2}|0\rangle\right)|u_k\rangle|v_k\rangle\right.\\
&\qquad\left.+\sum_{k=t+1}^{r}\sigma_k|0\rangle|u_k\rangle|v_k\rangle\right]. \qquad (23)
\end{aligned}$$

According to the state-of-the-art qPCA, shown in Fig. 1, when the measurement of the top register Anc1. is 1, the QSVT is performed; otherwise, the MQSVT is performed; then, we can obtain the state

$$\begin{aligned}
&\frac{1}{\sqrt{N_1}}\left[\frac{a}{\sqrt{a^2+b^2}}\sum_{k=1}^{t}\sigma_k|0\rangle(\sin x_k\alpha|1\rangle+\cos x_k\alpha|0\rangle)|u_k\rangle|v_k\rangle\right.\\
&+\frac{b}{\sqrt{a^2+b^2}}\sum_{k=1}^{t}\sigma_k|1\rangle\left(\sin\frac{x_k'\alpha}{2}|1\rangle+\cos\frac{x_k'\alpha}{2}|0\rangle\right)|u_k\rangle|v_k\rangle\\
&+\frac{a}{\sqrt{a^2+b^2}}\sum_{k=t+1}^{r}\sigma_k|0\rangle|0\rangle|u_k\rangle|u_k\rangle\\
&\left.+\frac{b}{\sqrt{a^2+b^2}}\sum_{k=t+1}^{r}\sigma_k|1\rangle|0\rangle|u_k\rangle|u_k\rangle\right] \qquad (24)
\end{aligned}$$

when the measurement of the register Anc is 1, we can obtain the state

$$\begin{aligned}
&\frac{1}{\sqrt{N_2}}\sum_{k=1}^{t}\sigma_k\left[\frac{a}{\sqrt{a^2+b^2}}|0\rangle\sin x_k\alpha+\frac{b}{\sqrt{a^2+b^2}}|1\rangle\sin\frac{x_k'\alpha}{2}\right]\\
&\qquad|1\rangle|u_k\rangle|v_k\rangle. \qquad (25)
\end{aligned}$$

Lin *et al.* figured out that by projecting the above quantum state onto a specific state to collapse to $|\psi_{A_0}\rangle$, e.g., setting $a = \frac{1}{\sqrt{5}}$, $b = \frac{2}{\sqrt{5}}$, and project onto $|+\rangle^{\mathrm{Anc1}}|1\rangle^{\mathrm{Anc}}$, one can obtain

$$\begin{aligned}
&\frac{1}{\sqrt{N_2}}\sum_{k=1}^{t}\sigma_k\left(\sin x_k\alpha+2\sin\frac{x_k'\alpha}{2}\right)|u_k\rangle|v_k\rangle\\
&\approx\frac{1}{\sqrt{N_2}}\sum_{k=1}^{t}\sigma_k(x_k+x_k')|u_k\rangle|v_k\rangle\\
&=\frac{1}{\sqrt{N_2}}\sum_{k=1}^{t}\sigma_k|u_k\rangle|v_k\rangle \qquad (26)
\end{aligned}$$

and then perform a phase estimation on $|\psi_{A_0}\rangle$ to obtain the output of the qPCA, i.e.,

$$|\psi_A'\rangle = \frac{1}{\sqrt{N_2}}\sum_{k=1}^{t}\sigma_k|\lambda_k\rangle|u_k\rangle|v_k\rangle \qquad (27)$$

where the top-t components are extracted.

III. PROPOSED QPCA

In this section, we propose a low-complexity qPCA as shown in Fig. 2, which consists of three steps. The first step can be expressed as

$$\begin{aligned}
U_{\{Simple,step1\}} &= (I\otimes U_{\mathrm{PE}}^\dagger)(I\otimes U_{\lambda,\tau}^\dagger\otimes I)(CU\otimes I)\\
&\quad\times(I\otimes U_{\lambda,\tau}\otimes I)(I\otimes U_{\mathrm{PE}}) \qquad (28)
\end{aligned}$$

which converts the r components shown in (1) to a $(t+(r-t))$ linear combination shown in (3). The second step is a measurement on auxiliary quits (Anc. in Fig. 2) to obtain the superposition state $\frac{1}{\sqrt{N_2}}\sum_{k=1}^{t}\sigma_k|u_k\rangle|v_k\rangle$. The third step can be expressed as

$$U_{\{Simple,step3\}} = I\otimes U_{\mathrm{PE}} \qquad (29)$$

which a phase estimation on registers A and M in Fig. 2 for obtaining the top-t eigenvalues.

When the measurement of the top qubit (register Anc.) is 1, the rest registers will be collapsed to $|\psi_A'\rangle$. Compared with the state-of-the-art qPCA, which requires five phase estimations, the proposed qPCA, as shown in Fig. 2, requires only three-phase estimations.

The proposed qPCA in (28) can be decomposed into following building blocks: the initialization of quantum registers, the first phase estimation U_{PE} for extracting the eigenvalues λ_ks, the unitary operation $U_{\lambda,\tau}$ for determining whether or not the eigenvalues are greater than the threshold τ, the operations $U^\dagger$ containing the reverse operations $U_{\mathrm{PE}}^\dagger$ and $U_{\lambda,\tau}^\dagger$ for removing unnecessary registers, the measurement, and the third phase estimation U_{PE} for extracting the target eigenvalues that are greater than the threshold τ.

A. INITIALIZATION

The preparation of the initial quantum state in quantum registers is shown in Fig. 2

$$|0\rangle^{\mathrm{Anc}}|0\rangle^{\mathrm{C}}|\tau\rangle^{\mathrm{B}}|0\rangle^{\mathrm{A}}|\psi_{A_0}\rangle^{\mathrm{M}}. \tag{30}$$

Time complexity and space complexity: Similar to the step in Section II-A, the time complexity of this step is $O(\log(pq))$. The number of qubits required for register M, which stores A_0, is $O(\log(pq))$.

B. PHASE ESTIMATION $I \otimes U_{PE}$

The purpose of the phase estimation U_{PE} is to extract eigenvalues λ_ks to quantum register A. Similar to the step in Section II-B, the eigenvalues λ_ks of the data matrix A_0 can be estimated as [10]

$$|0\rangle^{\mathrm{A}}|\psi_{A_0}\rangle^{\mathrm{M}} \xrightarrow{U_{\mathrm{PE}}} \frac{1}{\sqrt{N_1}}\sum_{k=1}^{r}\sigma_k|\lambda_k\rangle^{\mathrm{A}}|u_k\rangle|v_k\rangle. \tag{31}$$

Time complexity and space complexity: Similar to the step in Section II-B, the time complexity for obtaining U_{PE} is $O(t_0^2\epsilon^{-1}\log(pq))$. The number of qubits required for storing $|\lambda_k\rangle$s is $O(\log(\kappa))$ [7], where κ is the condition number of matrix A and $\lambda_k \geq 1/\kappa$ for all k.

C. UNITARY OPERATION $I \otimes U_{\lambda,\tau} \otimes I$

The purpose of this step is to filter out small eigenvalues in the quantum register C. One way to achieve this is to set $U_{\lambda,\tau}$ to be the transformation from λ_k to $y_k = (1 - \frac{\tau}{\lambda_k})_+$. Note that another way is to set $U_{\lambda,\tau}$ to be the transformation from λ_k to $w_k = (\lambda_k - \tau)_+$.

In this article, λ_k, τ, and y_k (or w_k) are used in binary to participate in quantum circuit calculations; therefore, we need to convert λ_k, τ, and y_k (or w_k) from decimal to binary. Assuming that there are six qubits stored λ_k, that is to say, the resolution of the binary code is 0.000001 (in binary). For example, let $\lambda_k = 0.2100$ in decimal and $\tau = 0.2000$ in decimal; then, $y_k \approx 0.0476$ in decimal, which equals $y_k \approx 0.000011$ in binary, and $w_k = 0.0100$ in decimal, which equals $w_k \approx 0.000001$ in binary.

We proposed the above two methods to filter the eigenvalues smaller than threshold. w_k can be implemented in a simpler quantum circuit with more qubits, and y_k can be implemented with fewer qubits in a more complicated circuit, but this additional running time of this part circuit can be negligible, compared with the overall running time of qPCA. Sometimes, it is preferable to have shallower circuits at the cost of employing some extra ancillary qubits.

Now, let us see how to obtain $U_{\lambda,\tau}$ as the transformation from λ_k to y_k. We first set the intermediate variable $z_k = z(\lambda_k) = 1/\lambda_k$ and apply the Newton method to the function $f(z_k) = 1/z_k - \lambda_k$; then, we can obtain the iteration function [13]

$$z_k^{(i+1)} = z_k^{(i)} - \frac{f(z_k^{(i)})}{f'(z_k^{(i)})} = 2z_k^{(i)} - (z_k^{(i)})^2\lambda_k. \tag{32}$$

After obtaining z_k by the Newton's iteration, as shown in Fig. 3(b), $y_k = (1 - \tau z_k)_+$ can be obtained by Fig. 3(c). In addition, the complete quantum circuit of the unitary operation $U_{\lambda,\tau}$ is given in Fig. 3. In summary, the unitary operation $U_{\lambda,\tau}$ can be represented as

$$\begin{aligned}&|0\rangle^C\frac{1}{\sqrt{N_1}}\sum_{k=1}^{r}\sigma_k|\lambda_k\rangle|u_k\rangle|v_k\rangle\\ &\xrightarrow{U_{\lambda,\tau}} \frac{1}{\sqrt{N_1}}\sum_{k=1}^{r}\sigma_k|y_k\rangle^C|\lambda_k\rangle|u_k\rangle|v_k\rangle.\end{aligned} \tag{33}$$

Time complexity: Similar to the step in Section II-C, the time complexity of $U_{\lambda,\tau}$ is determined by the Newton's iteration. The initial state of the iteration (32) is $z_k^{(0)} = 2^{p-1}$, where p satisfies $2^{-p} \leq \lambda_k < 2^{1-p}$. The number of iterations for obtaining $U_{\lambda,\tau}$ is approximately $s = O(\log n)$, where n is the number of qubits in the register storing $|\lambda_k\rangle$. The time complexity of obtaining $U_{\lambda,\tau}$ by the Newton's iteration is $O(\mathrm{poly}(n) \times s) = O(\mathrm{poly}(\log\kappa)\log(\log\kappa))$. Compared with the time complexity of the phase estimation, which is $O(\kappa^2\epsilon^{-3}\log(pq))$, the time complexity of the Newton's iteration can be ignored.

Space complexity: In general, the registers storing $|y_k\rangle$, $|\lambda_k\rangle$, $|\tau\rangle$ require the same number of qubits. Therefore, in this step, we can infer that the number of qubits required in registers B and C are both $O(\log(\kappa))$.

Accuracy: There are two types of errors incurred by the Newton's iteration: 1) the expected error e_s, brought by the finite number (i.e., s) of iterations, and 2) the average truncation error e_n, brought by the finite number of available digits. We define $\hat{\lambda_k} = \lambda_k - e_n$; then, we have

$$\begin{aligned}z_k^{(i)} - \frac{1}{\hat{\lambda_k}} &= 2z_k^{(i-1)} - \hat{\lambda_k}(z_k^{(i-1)})^2 - \frac{1}{\hat{\lambda_k}}\\ &= -\left(z_k^{(i-1)} - \frac{1}{\hat{\lambda_k}}\right)^2\hat{\lambda_k}\\ &\leq 0\end{aligned} \tag{34}$$

and the Newton iteration convergence is quadratic, because the error e_s can be written as

$$e_s^{(i)} = \left|z_k^{(i)} - \frac{1}{\hat{\lambda_k}}\right| = \left|2z_k^{(i-1)} - \hat{\lambda_k}z_k^{(i-1)2} - \frac{1}{\hat{\lambda_k}}\right| = e_s^{(i-1)2}\hat{\lambda_k}. \tag{35}$$

In addition, according to the initial state $z_k^{(0)}$, we have

$$\begin{aligned}e_s^{(0)}\hat{\lambda_k} &= \left|z_k^{(0)} - \frac{1}{\hat{\lambda_k}}\right|\hat{\lambda_k} = |z_k^{(0)}\hat{\lambda_k} - 1|\\ &= 1 - z_k^{(0)}\hat{\lambda_k} = 1 - 2^{p-1}\hat{\lambda_k}\\ &\leq 1 - 2^{p-1}\times 2^{-p} = \frac{1}{2}.\end{aligned} \tag{36}$$

IEEE Transactions on Quantum Engineering

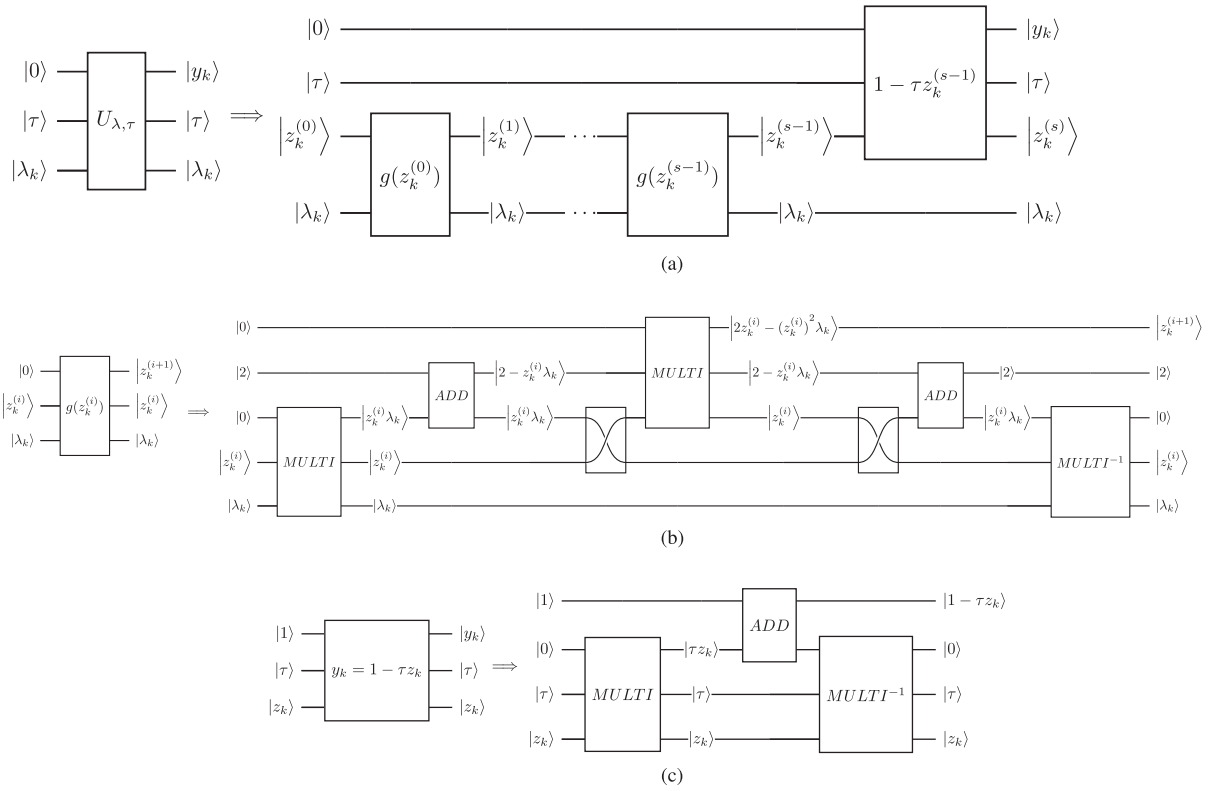

FIGURE 3. (a) Quantum circuit of the unitary operation $U_{\lambda,\tau}$. (b) Quantum circuit for one Newton's iteration. (c) Quantum circuit for computing $y_k = 1 - \tau z_k$.

Therefore, the total error $e_s + e_n$ is given by

$$\begin{aligned} e_s + e_n &= e_s^{(i-1)^2}\hat{\lambda_k} + s2^{-n} \\ &= \frac{1}{\hat{\lambda_k}}(e_s^{(0)}\hat{\lambda_k})^{2^s} + s2^{-n} \\ &\leq 2^{(-2^s)}\kappa + 2^{-n}s = \frac{\kappa + \log n}{2^n}. \end{aligned} \tag{37}$$

D. UNITARY CONTROLLED-NOT OPERATION $CU \otimes I$

The purpose of this step is to employ unitary controlled-NOT operation and ancillary qubit to tell whether or not the eigenvalue in the measured quantum bits corresponds to a principal component. If $y_k > 0$ (i.e., $\lambda_k > \tau$), the unitary controlled-NOT operation will reverse the top qubit (i.e., the ancillary qubit); otherwise, it will do nothing. In other words, this step is to split the r components into $(t + (r - t))$ linear combinations. This procedure can be expressed as

$$\begin{aligned} &\frac{1}{\sqrt{N_1}}|0\rangle^{\text{Anc}}\sum_{k=1}^{r}\sigma_k|y_k\rangle|\lambda_k\rangle|u_k\rangle|v_k\rangle \\ \xrightarrow{CU} &\frac{1}{\sqrt{N_1}}(|1\rangle^{\text{Anc}}\sum_{k=1}^{t}\sigma_k|y_k\rangle|\lambda_k\rangle|u_k\rangle|v_k\rangle \\ &+|0\rangle^{\text{Anc}}\sum_{k=t+1}^{r}\sigma_k|0\rangle|\lambda_k\rangle|u_k\rangle|v_k\rangle). \end{aligned} \tag{38}$$

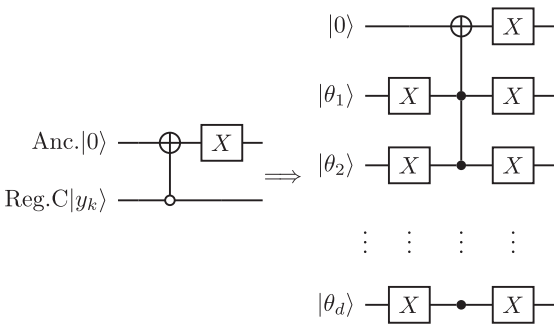

FIGURE 4. Quantum circuit to implement the *CU* operation.

Time complexity: The controlled-NOT operation can be decomposed into a combination of general quantum gates in two ways, with or without work qubits, and these two cases yield different complexities.

For the case without using work qubits, the complexity of the decomposition of the controlled-NOT operation is $O(d^2) + O(2d + 1) = O((\log\kappa)^2)$. For the case with work qubits being used, the complexity of decomposing controlled-NOT operation is $O(d) + O(2d + 1) = O(\log\kappa)$.

The details of the complexity analysis are as follows: As shown in Fig. 4, to decompose the CN operation, $(2d + 1)$ X quantum gates and a $C^d(X)$ operation are required, where $C^d(X)$ is a NOT quantum gate controlled by d qubits. Then,

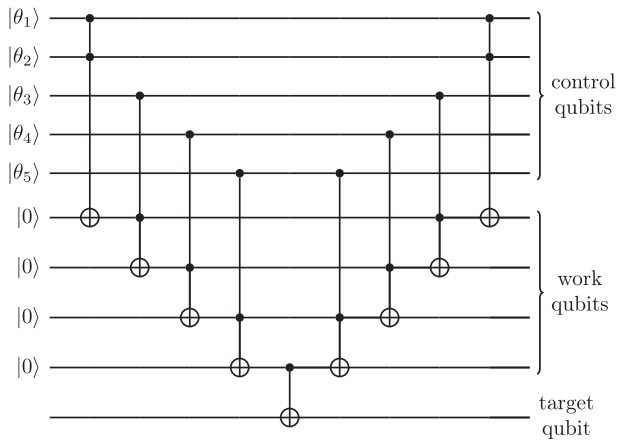

FIGURE 5. Quantum circuit to implement the C^dX operation, using work qubits.

we can find a circuit containing $O(d^2)$ Toffoli and X gates to implements the $C^d(X)$ operation with using no work qubit. We can also find a circuit containing $O(d)$ Toffoli and X gates to implement $C^d(X)$ operation with using $(d-1)$ work qubits, as shown in Fig. 5.

Space complexity: In this step, the top register Anc. requires only one qubit.

E. UNITARY REVERSE OPERATION $I \otimes U^\dagger$

The purpose of this step is to remove the unnecessary registers storing $|y_k\rangle$ and $|\lambda_k\rangle$. Since $U^\dagger = U_{\mathrm{PE}}^\dagger U_{\lambda,\tau}^\dagger$, we have

$$\begin{aligned}
&\frac{1}{\sqrt{N_1}}(|1\rangle \sum_{k=1}^{t} \sigma_k |y_k\rangle^C |\lambda_k\rangle^A |u_k\rangle|v_k\rangle \\
&\qquad + |0\rangle \sum_{k=t+1}^{r} \sigma_k |0\rangle^C |\lambda_k\rangle^A |u_k\rangle|v_k\rangle) \\
\xrightarrow{U^\dagger} &\frac{1}{\sqrt{N_1}}(|1\rangle|0\rangle^C|0\rangle^A \sum_{k=1}^{t} \sigma_k |u_k\rangle|v_k\rangle \\
&\qquad + |0\rangle|0\rangle^C|0\rangle^A \sum_{k=t+1}^{r} \sigma_k |u_k\rangle|v_k\rangle). \qquad (39)
\end{aligned}$$

Time complexity: In Sections III-B and III-C, it has been shown that the time complexities for obtaining U_{PE} and $U_{\lambda,\tau}$ are $O(t_0^2\epsilon^{-1}\log(pq))$ and $O(\mathrm{poly}(\log\kappa)\log(\log\kappa))$ ($t_0 = O(\kappa/\epsilon)$), respectively; therefore, the time complexity for obtaining $U^\dagger$ is $O(t_0^2\epsilon^{-1}\log(pq))$.

F. MEASUREMENT

In this step, the register Anc. is measured to remove the eigenvalues that are smaller than the threshold τ from $|\psi_{A_0}\rangle$. If the measurement is 1, the state of remaining qubits will

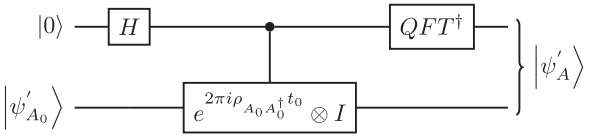

FIGURE 6. Quantum circuit of the phase estimation.

collapse to

$$|\psi'_{A_0}\rangle = \frac{1}{\sqrt{N_2}} \sum_{k=1}^{t} \sigma_k |u_k\rangle|v_k\rangle \qquad (40)$$

where $N_2 = \sum_{k=1}^{t} \sigma_k^2$. We can see that $|\psi'_{A_0}\rangle$ only contains top-t components of A.

G. PHASE ESTIMATION $I \otimes U_{PE}$

In this step, a phase estimation is performed on $|\psi'_{A_0}\rangle$ to extract the top-t largest eigenvalues, as shown in Fig. 6, i.e.,

$$|\psi'_{A_0}\rangle \xrightarrow{U_{\mathrm{PE}}} |\psi'_A\rangle. \qquad (41)$$

As we can see in (2), the top-t eigenvalues $|\lambda_k\rangle$ are stored in the register of the state $|\psi'_A\rangle$.

Time complexity: The time complexity of this step is the time complexity of U_{PE}, which has been shown to be $O(t_0^2\epsilon^{-1}\log(pq))$ in Section III-B.

IV. COMPLEXITY AND ACCURACY ANALYSIS

In this section, we analyze the time complexity, the space complexity, and also the accuracy for the proposed qPCA. Moreover, we compared them with those of the state-of-the-art qPCA in [5].

A. TIME COMPLEXITY

Similar to Lin's qPCA, the time complexity of the proposed qPCA is mainly determined by the Hamilton simulation $e^{2\pi i\rho t_0}$ in the phase estimation. However, Lin's qPCA requires five phase estimations, as shown in Fig. 1, while there are only three phase estimations for the proposed qPCA, as shown in Fig. 2. Therefore, the time complexity of the proposed qPCA is approximately 3/5 of that of Lin's qPCA.

It is worth to mention here that the focus of this work, as well as other works on qPCA [5]–[7], is on runtime of the qPCA circuit, which has a time complexity of $O(\mathrm{poly}(\log(pq)))$. However, the time complexity to compute the circuit given the classical representation of the matrix is $O(pq)$.

B. SPACE COMPLEXITY

As shown in Section III, registers Anc., A, B, C, and M require one qubit, $O(\log(\kappa))$ qubits, $O(\log(\kappa))$ qubits, $O(\log(\kappa))$ qubits, and $O(\log(pq))$ qubits, respectively, while Lin's qPCA, shown in Fig. 1, requires an additional ancillary register Anc1, which contains two qubits. Therefore, Lin's qPCA requires one more register than the proposed qPCA.

Algorithm 1: Low-Complexity qPCA Algorithm.

Input:

A quantum state $|\psi_{A_0}\rangle = \frac{1}{\sqrt{N_1}} \sum_{k=1}^{r} \sigma_k |u_k\rangle|v_k\rangle$;

A unitary operation $U_{\mathrm{PE}} = e^{2\pi i \rho_{A_0 A_0^\dagger} t_0}$;

A threshold constant τ.

Output:

A quantum state $|\psi'_A\rangle = \frac{1}{\sqrt{N_2}} \sum_{k=1}^{t} \sigma_k |\lambda_k\rangle|u_k\rangle|v_k\rangle$.

Procedure:

1: Prepare quantum state $|\psi_1\rangle = |0\rangle^{\mathrm{Anc}}|0\rangle^C|0\rangle^A|\psi_{A_0}\rangle^M$.

2: Perform the phase estimation U_{PE} to obtain $|\psi_2\rangle = \frac{1}{\sqrt{N_1}}|0\rangle^{\mathrm{Anc}}|0\rangle^C \sum_1^r \sigma_k |\lambda_k\rangle^A |u_k\rangle|v_k\rangle$.

3: Perform the unitary operation $U_{\lambda,\tau}$ to obtain $|\psi_3\rangle = \frac{1}{\sqrt{N_1}}(|0\rangle^{\mathrm{Anc}} \sum_{k=1}^{t} \sigma_k |y_k\rangle^C |\lambda_k\rangle^A |u_k\rangle|v_k\rangle + |0\rangle^{Anc} \sum_{k=t+1}^{r} \sigma_k |0\rangle^C |\lambda_k\rangle^A |u_k\rangle|v_k\rangle)$.

4: Perform the controlled-NOT operation CU to obtain $|\psi_4\rangle = \frac{1}{\sqrt{N_1}}(\sum_{k=1}^{t} \sigma_k |1\rangle^{\mathrm{Anc}} |y_k\rangle^C |\lambda_k\rangle^A |u_k\rangle|v_k\rangle + \sum_{k=t+1}^{r} \sigma_k |0\rangle^{\mathrm{Anc}} |0\rangle^C |\lambda_k\rangle^A |u_k\rangle|v_k\rangle)$.

5: Employ unitary operation $U^\dagger$ to obtain $|\psi_5\rangle = \frac{1}{\sqrt{N_1}}(\sum_{k=1}^{t} \sigma_k |1\rangle^{\mathrm{Anc}} |u_k\rangle|v_k\rangle + \sum_{k=t+1}^{r} \sigma_k |0\rangle^{\mathrm{Anc}} |u_k\rangle|v_k\rangle)$.

6: Measurement. When the measurement of the top qubit is 1, the rest registers will collapse to $|\psi'_{A_0}\rangle = \frac{1}{\sqrt{N_2}} \sum_{k=1}^{t} \sigma_k |u_k\rangle|v_k\rangle$.

7: Extract top-t largest eigenvalues from a phase estimation and get $|\psi'_A\rangle = \frac{1}{\sqrt{N_2}} \sum_{k=1}^{t} \sigma_k |\lambda_k\rangle|u_k\rangle|v_k\rangle$.

C. ACCURACY

We compare the accuracy of Lin's algorithm and ours under the same fidelity. Fidelity is a measure of the distance between two quantum states. For Lin's algorithm, the fidelity can be adjusted by adjusting α in (21). For our algorithm, the fidelity can be adjusted by adjusting the number of iterations for approximating y_k in (33). The error of the proposed qPCA is brought by the Newton iterations and the phase estimations. The error from Newton iterations has been shown to be $O(2^{(-2^s)}\kappa + 2^{-n}s)$ in Section III-C. Then, we analyzed the error from the phase estimation. For λ_k obtained by the phase estimation U_{PE}, we have $\lambda_k = j_k/t_0$, where $t_0 = O(\kappa/\epsilon)$, j_k is the binary form of λ_k, and the error of λ_k is $O(1/t_0)$ [6]. Then, the error of λ_k propagates to $y_k = 1 - \frac{\tau}{\lambda_k}$, which becomes $O(1/(\lambda_k t_0))$. Since $1/(\lambda_k t_0) \gg 2^{(-2^s)}\kappa + 2^{-n}s$ for large enough s, i.e., enough number of iterations, the error of the proposed qPCA is $O(1/(\lambda_k t_0))$. However, in Lin's qPCA shown in Fig. 1, the error of λ_k propagates to $x_k = 1 - \frac{\tau}{\sigma_k} = 1 - \frac{\tau}{\sqrt{\lambda_k}}$ or $x'_k = 1 + \frac{\tau}{\sigma_k} = 1 + \frac{\tau}{\sqrt{\lambda_k}}$, which becomes $O(1/(\sqrt{\lambda_k} t_0))$. Therefore, the error of the purposed qPCA is smaller than that of Lin's qPCA under the same fidelity.

V. EXPERIMENT

In this section, we perform simulation experiments on the IBM quantum computing platform [14], i.e., IBM QX [15]–[17], to validate the proposed qPCA.

A. SIMULATION FOR 2 × 2 MATRIX

First, we take the 2×2 matrix

$$A_0 = \begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{bmatrix} \tag{42}$$

as an example, whose quantum state is given by [18]

$$|\psi_{A_0}\rangle = \frac{3}{\sqrt{20}}|00\rangle + \frac{1}{\sqrt{20}}|01\rangle + \frac{1}{\sqrt{20}}|10\rangle + \frac{3}{\sqrt{20}}|11\rangle. \tag{43}$$

Note that the eigenvalues and the corresponding eigenvectors obtained by classical PCA are given by

$$\lambda_1 = 2, \quad u_1 = [1, 1]^T$$
$$\lambda_2 = 1, \quad u_2 = [-1, 1]^T.$$

When we take a threshold $\tau = 1.1$ for eigenvalues, the eigenvalue in binary $|\lambda_1\rangle = |10\rangle$ and the corresponding eigenvector $|u_1\rangle = [1, 1]^T = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ are considered as principal components. Therefore, the output of the proposed algorithm should be given by

$$\begin{aligned} |\psi'_A\rangle &= |\lambda_1\rangle|u_1\rangle|u_1\rangle \\ &= |10\rangle \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \\ &= \frac{1}{2}|1000\rangle + \frac{1}{2}|1001\rangle + \frac{1}{2}|1010\rangle + \frac{1}{2}|1011\rangle. \end{aligned} \tag{44}$$

The simulation circuit for the proposed qPCA on the IBM Quantum Composer is shown in Fig. 7. The IBM Quantum Composer is a graphical quantum programming tool of IBM QX. For the 2×2 matrix A_0 in (42), five qubits are required in total. The qubit q[0] is an ancillary qubit. Before the first dash line of the quantum circuit, the qubits q[3–4] are initialized in the quantum state $|\psi_{A_0}\rangle$. Between the first dash and the second dash lines, the qubits q[1–2] save the eigenvalues $|\lambda_k\rangle$ from the phase estimation U_{PE}. Between the second and the third dash lines, $|\lambda_k\rangle$ is converted to $|y_k\rangle$ by $U_{\lambda,\tau}$. Between the third and the fourth dash lines, the controlled-NOT operation CU is performed. Between the fourth and the sixth dash lines, the operation $U^\dagger = U^\dagger_{\lambda,\tau} U^\dagger_{\mathrm{PE}}$ is performed. The rest of the circuit is U_{PE} and the measurement. When the measurement of q[0] yields 1, q[3–4] will collapse to the quantum state $|\psi'_A\rangle$.

The IBM QX platform provides the result by running "ibmq_qasm_simulator," a general-purpose simulator for simulating quantum circuits [14]. The result is shown in Fig. 8. For each sequence on the horizontal axis, counting from the right to the left, the first bit (i.e., q[0]) is the ancillary qubit; the second and third qubits (i.e., q[1–2]) represent the

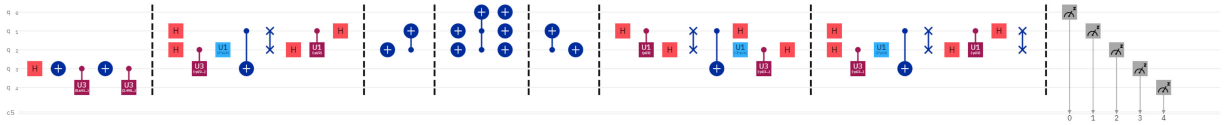

FIGURE 7. Simulation circuit of our qPCA for the 2 × 2 matrix A_0 on IBM QX.

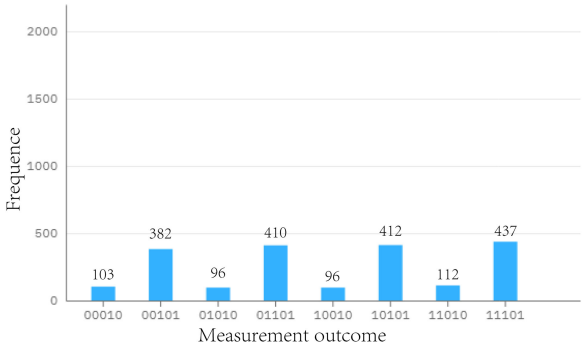

FIGURE 8. Simulation result for the matrix A_0 on IBM QX.

eigenvalues; the fourth and fifth qubits (i.e., q[3–4]) represent the eigenvectors. The outputs "00101," "01101," "10101," and "11101," whose first qubits (counting from right) are 1, represent the principal component of the matrix A_0, and the corresponding states of $|\lambda_k\rangle|u_k\rangle|u_k\rangle$ are $|1000\rangle$, $|1001\rangle$, $|1010\rangle$, and $|1011\rangle$. Therefore, $|\psi'_A\rangle_s$, the simulated state of $|\psi'_A\rangle$, is given by

$$\begin{aligned}|\psi'_A\rangle_s &= \sqrt{\frac{382}{1641}}|1000\rangle + \sqrt{\frac{410}{1641}}|1001\rangle \\ &\quad + \sqrt{\frac{412}{1641}}|1010\rangle + \sqrt{\frac{437}{1641}}|1011\rangle \\ &\approx \frac{1}{2}|1000\rangle + \frac{1}{2}|1001\rangle + \frac{1}{2}|1010\rangle + \frac{1}{2}|1011\rangle \quad (45)\end{aligned}$$

which matches $|\psi'_A\rangle$ in (44).

B. SIMULATION FOR 4 × 4 MATRIX

Now, we take the 4 × 4 matrix

$$C_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \quad (46)$$

as an example, whose quantum state is given by [18]

$$|\psi_{C_0}\rangle = \frac{1}{\sqrt{14}}|0101\rangle + \frac{2}{\sqrt{14}}|1010\rangle + \frac{3}{\sqrt{14}}|1111\rangle. \quad (47)$$

Note that the eigenvalues and the corresponding eigenvectors obtained by classical PCA are given by

$$\begin{aligned}\lambda_1 &= 0, \quad u_1 = [1, 0, 0, 0]^T \\ \lambda_2 &= 1, \quad u_2 = [0, 1, 0, 0]^T \\ \lambda_3 &= 2, \quad u_3 = [0, 0, 1, 0]^T \\ \lambda_4 &= 3, \quad u_4 = [0, 0, 0, 1]^T.\end{aligned}$$

When we take a threshold $\tau = 1.1$ for eigenvalues, the eigenvalues $|\lambda_3\rangle = 2 = |10\rangle$ and $|\lambda_4\rangle = 3 = |11\rangle$ and the corresponding eigenvectors $|u_3\rangle = [0, 0, 1, 0]^T = |10\rangle$ and $|u_4\rangle = [0, 0, 0, 1]^T = |11\rangle$ are considered as principal components. Therefore, the output of the proposed algorithm should be given by

$$\begin{aligned}|\psi'_C\rangle &= \frac{\lambda_3}{\sqrt{\lambda_3^2 + \lambda_4^2}}|\lambda_3\rangle|u_3\rangle|u_3\rangle + \frac{\lambda_4}{\sqrt{\lambda_3^2 + \lambda_4^2}}|\lambda_4\rangle|u_4\rangle|u_4\rangle \\ &= \frac{2}{\sqrt{13}}|10\rangle \otimes |10\rangle \otimes |10\rangle + \frac{3}{\sqrt{13}}|11\rangle \otimes |11\rangle \otimes |11\rangle \\ &= \frac{2}{\sqrt{13}}|101010\rangle + \frac{3}{\sqrt{13}}|111111\rangle. \quad (48)\end{aligned}$$

The simulation circuit for the proposed qPCA for the 4 × 4 matrix C_0 is shown in Fig. 9. The circuit requires eight qubits in total. The qubit q[0] is the ancillary qubit; the last qubit q[7] is an additional qubit to participate in the computing of $U_{\lambda,\tau}$. Before the first dash line of the quantum circuit, the qubits q[3–6] are initialized in the quantum state $|\psi_{C_0}\rangle$. Between the first and the second dash lines, the qubits q[1–2] save eigenvalues $|\lambda_k\rangle$ from the phase estimation U_{PE}. Between the second and the third dash lines, $|\lambda_k\rangle$ is converted to $|y_k\rangle$ by $U_{\lambda,\tau}$. Between the third and the fourth dash lines, the controlled-NOT operation CU is performed. Between the fourth and the sixth dash lines, the operation $U^\dagger = U^\dagger_{\lambda,\tau}U^\dagger_{\mathrm{PE}}$ is performed. The rest of the circuit is U_{PE} and measurement. When the measurement of q[0] is 1, q[3–6] will collapse to $|\psi'_C\rangle$.

In addition, the preparation of the initial state $|\psi_{C_0}\rangle$ and the construction of the phase estimation operation are not straightforward. Therefore, we design the binary tree to prepare $|\psi_{C_0}\rangle$, as shown in Fig. 11 of Appendix A, and the corresponding quantum circuit is shown in Fig. 12. The quantum circuit of the phase estimation is shown in Fig. 13 of Appendix B.

The result is shown in Fig. 10. For each sequence on the horizontal axis, counting from the right to the left, the first bit (i.e., q[0]) is the ancillary qubit, the second and third qubits (i.e., q[1–2]) represent the eigenvalues, the fourth to seventh qubits (i.e., q[3–6]) represent the eigenvectors, and the last bit (i.e., q[7]) is only an addition qubit to participate in computing of $U_{\lambda,\tau}$. The outputs "01010101" and "01111111," whose first qubits (counting from right) are 1, represent the principal component of the matrix C_0, and the corresponding states of $|\lambda_k\rangle|u_k\rangle|u_k\rangle$ are $|101010\rangle$ and $|111111\rangle$. Therefore, $|\psi'_C\rangle_s$, the

IEEE Transactions on Quantum Engineering

FIGURE 9. Simulation circuit of our qPCA for the 4 × 4 matrix C_0 on IBM QX.

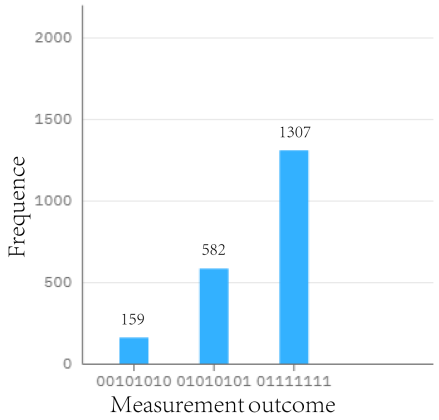

FIGURE 10. Simulation result for the matrix C_0 on IBM QX.

simulated state of $|\psi_C'\rangle$, is given by

$$|\psi_C'\rangle_s = \sqrt{\frac{582}{1889}}|101010\rangle + \sqrt{\frac{1307}{1889}}|111111\rangle \approx \frac{2}{\sqrt{13}}|101010\rangle + \frac{3}{\sqrt{13}}|111111\rangle \quad (49)$$

which matches $|\psi_C'\rangle$ in (48).

As we can see from the simulation results, the proposed qPCA yields the expected quantum state.

VI. CONCLUSION

In this article, we proposed a low-complexity qPCA algorithm, which yielded the quantum state only containing the principal components. This is similar to the state-of-the-art qPCA algorithm in [5]. However, instead of indirectly filtering out the components with small eigenvalues, the proposed qPCA can achieve this directly. As a result, the circuit of the proposed qPCA was much simpler than that of the state of the art. Particularly, the proposed qPCA only required three phase estimations, while the state-of-the-art qPCA required five, and this led to a roughly 3/5 runtime of that of the state of the art. Simulations on IBM QX verified that the proposed qPCA yielded the expected quantum state.

APPENDIX A
PREPARATION OF STATE $|\psi_{C_0}\rangle$

Since the size of the matrix C_0 is large, the corresponding quantum state $|\psi_{C_0}\rangle$ in (47) is not straightforward to prepare on IBM QX. According to [19], a state can be generated by a binary tree, whose internal nodes store the amplitudes of the state. Therefore, we design the binary tree as shown in Fig. 11 to prepare the quantum state $|\psi_{C_0}\rangle$, and the values in

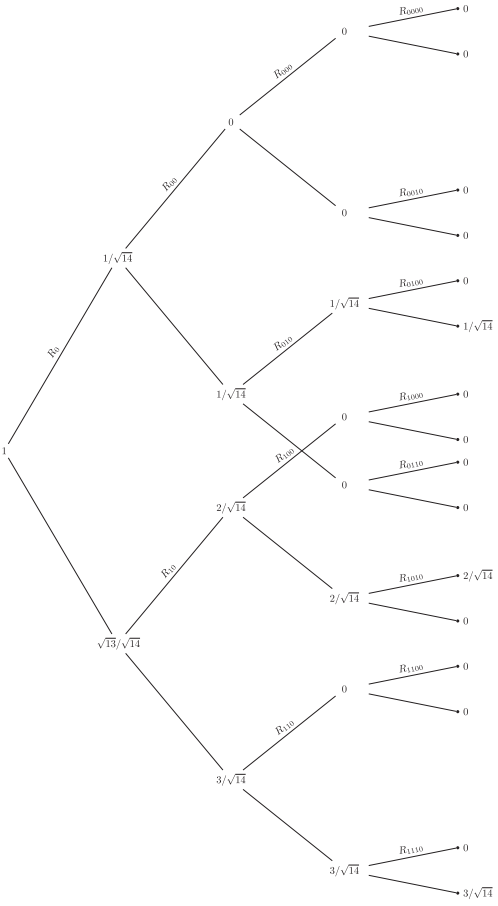

FIGURE 11. Binary tree to prepare $|\psi_{C_0}\rangle$.

the internal nodes equal the amplitudes of $|\psi_{C_0}\rangle$. Each branch in the binary tree is $R_y(\theta)$ unitary operation, where

$$R_y(\theta) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}. \quad (50)$$

The corresponding quantum circuit of the binary tree is shown in Fig. 12.

APPENDIX B
PHASE ESTIMATION OF MATRIX C

Since the size of the matrix C_0 is large, the quantum circuit of the phase estimation for matrix C_0 is not straightforward to design on IBM QX. Therefore, we decompose the phase estimation operation into several unitary operations, which can be implemented by simple quantum gates [20]. The unitary operations in the phase estimation of C_0 are $U_1 = e^{\frac{2\pi i C_0}{4}}$

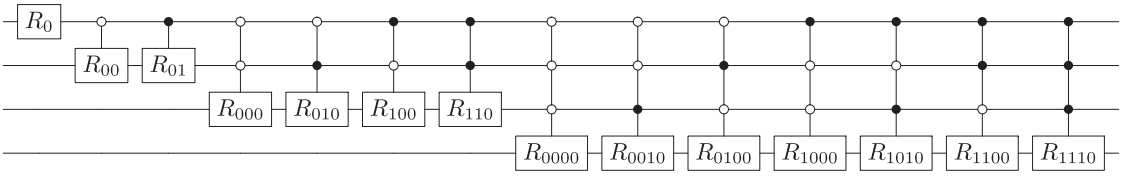

FIGURE 12. Quantum circuit to prepare $|\psi_{C_0}\rangle$.

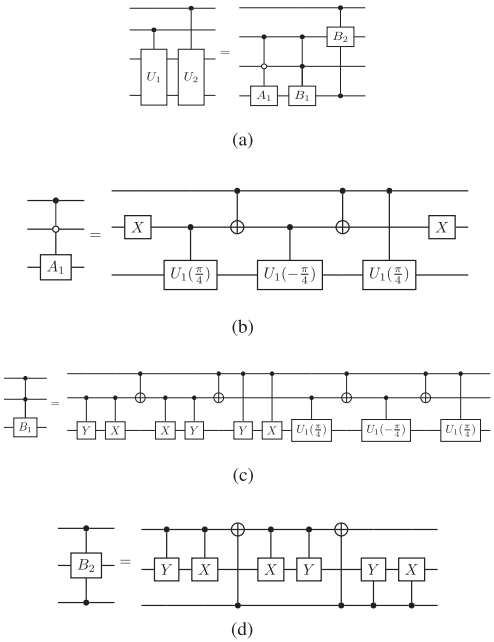

FIGURE 13. Decomposed circuit of the matrix U_1 and U_2. (a) Unitary decomposition of U_1 and U_2. (b) Unitary decomposition of A_1. (c) Unitary decomposition of B_1. (d) Unitary decomposition of B_2.

and $U_2 = e^{\frac{2\pi i C_0}{2}}$ [21], i.e.,

$$U_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -i \end{bmatrix}, \; U_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \tag{51}$$

The corresponding decomposition circuits of the operations U_1 and U_2 are shown in Fig. 13.

REFERENCES

[1] W. Eberly and E. Kaltofen, "On randomized Lanczos algorithms," in *Proc. Int. Symp. Symbolic Algebr. Comput.*, 1997, pp. 176–183, doi: 10.1145/258726.258776.

[2] M. Moore and A. Narayanan, "Quantum-inspired computing," Dept. Comput. Sci., Univ. Exeter, Exeter, U.K., 1995.

[3] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nat. Phys.*, vol. 10, pp. 631–633, 2014, doi: 10.1038/nphys3029.

[4] A. Bellante, A. Luongo, and S. Zanero, "Quantum algorithms for data representation and analysis," 2021, *arXiv:2104.08987*.

[5] J. Lin, W.-S. Bao, S. Zhang, T. Li, and X. Wang, "An improved quantum principal component analysis algorithm based on the quantum singular threshold method," *Phys. Lett. A*, vol. 383, no. 24, pp. 2862–2868, 2019, doi: 10.1016/j.physleta.2019.06.026.

[6] B. Duan, J. Yuan, Y. Liu, and D. Li, "Quantum algorithm for support matrix machines," *Phys. Rev. A*, vol. 96, no. 3, 2017, Art. no. 032301, doi: 10.1103/PhysRevA.96.032301.

[7] B. Duan, J. Yuan, Y. Liu, and D. Li, "Efficient quantum circuit for singular-value thresholding," *Phys. Rev. A*, vol. 98, no. 1, 2018, Art. no. 012308, doi: 10.1103/PhysRevA.98.012308.

[8] V. Giovannetti, S. Lloyd, and L. Maccone, "Quantum random access memory," *Phys. Rev. Lett.*, vol. 100, no. 16, 2008, Art. no. 160501, doi: 10.1103/PhysRevLett.100.160501.

[9] M. Schuld, I. Sinayskiy, and F. Petruccione, "Prediction by linear regression on a quantum computer," *Phys. Rev. A*, vol. 94, no. 2, 2016, Art. no. 022342, doi: 10.1103/PhysRevA.94.022342.

[10] B. Soni, N. Khare, K. K. Soni, and A. Rasool, "Classical equivalent quantum based efficient data preprocessing algorithm," in *Proc. 11th Int. Conf. Comput., Commun. Netw. Technol.*, 2020, pp. 1–7, doi: 10.1109/ICCCNT49239.2020.9225473.

[11] G. H. Low and I. L. Chuang, "Hamiltonian simulation by qubitization," *Quantum*, vol. 3, 2019, Art. no. 163, doi: 10.22331/q-2019-07-12-163.

[12] M. K. Bhaskar, S. Hadfield, A. Papageorgiou, and I. Petras, "Quantum algorithms and circuits for scientific computing," *Quantum Inf. Comput.*, vol. 16, pp. 197–236, 2016, doi: 10.26421/QIC16.3-4-2.

[13] J.-M. Liang, S.-Q. Shen, and M. Li, "Quantum algorithms and circuits for linear equations with infinite or no solutions," *Int. J. Theor. Phys.*, vol. 58, no. 8, pp. 2632–2640, 2019, doi: 10.1007/s10773-019-04151-2.

[14] IBM, "IBM Quantum Experience." Jan. 5, 2022. [Online]. Available: https://quantum-computing.ibm.com/

[15] A. Cross, "The IBM Q experience and QISKit open-source quantum computing software," *Bull. Amer. Phys. Soc.*, vol. 2018, 2018. [Online]. Available: https://meetings.aps.org/Meeting/MAR18/Session/L58.3

[16] P. Balasubramanian, B. Behera, and P. Panigrahi, "Circuit implementation for rational quantum secure communication using IBM Q experience beta platform," doi: 10.13140/RG.2.2.28733.31207.

[17] D. García-Martín and G. Sierra, "Five experimental tests on the 5-qubit IBM quantum computer," *J. Appl. Math. Phys.*, vol. 6, no. 7, pp. 1460–1475, 2018, doi: 10.4236/jamp.2018.67123.

[18] P. J. Coles *et al.*, "Quantum algorithm implementations for beginners," 2018, *arXiv:1804.03719*.

[19] A. Prakash, "Quantum algorithms for linear algebra and machine learning," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, CA, USA, 2014.

[20] J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, "Efficient decomposition of quantum gates," *Phys. Rev. Lett.*, vol. 92, no. 17, 2004, Art. no. 177902, doi: 10.1103/PhysRevLett.92.177902.

[21] H. Mohammadbagherpoor, Y. H. Oh, A. Singh, X. Yu, and A. J. Rindos, "Experimental challenges of implementing quantum phase estimation algorithms on IBM quantum computer," 2019, *arXiv:1903.07605*.

Chen He (Member, IEEE) received the B.Eng. degree (*summa cum laude*) from McMaster University, Hamilton, ON, Canada, in 2007, and the M.A.Sc. and Ph.D. degrees from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2009 and 2014, respectively, all in electrical and computer engineering.

He was a Research Engineer with Blackberry Limited, Waterloo, ON, and a Postdoctoral Research Fellow with UBC. Since 2016, he has been a Full Professor with Northwest University, Xi'an, China. His research interests include wireless communications, signal processing, and quantum information science.

Dr. He is an Associate Editor for IEEE SIGNAL PROCESSING LETTERS.

IEEE Transactions on Quantum Engineering

Jiazhen Li received the bachelor's degree in applied mathematics and computing from the Taiyuan University of Technology, Taiyuan, China, in 2018, and the master's degree in computer science from Northwest University, Xi'an, China, in 2021.

Her research interests include quantum computing and machine learning.

Weiqi Liu received the Ph.D. degree in physics from Northwest University, Xi'an, China, in 2018.

She is currently a Lecturer with Northwest University. Her research interests include quantum private communication and quantum information.

Jinye Peng received the M.S. degree in computer science from Northwestern University, Xi'an, China, in 1996, and the Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, in 2002.

In 2006, he joined Northwest University as a Professor. He is currently the Dean of the School of Information Science and Technology, Northwest University. His research interests include statistical signal processing, image processing, and machine learning.

Z. Jane Wang (Fellow, IEEE) received the B.Sc. degree from Tsinghua University, Beijing, China, in 1996, and the M.Sc. and Ph.D. degrees from the University of Connecticut, Storrs, CT, USA, in 2000 and 2002, respectively, all in electrical engineering.

She has been a Research Associate with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA. Since 2004, she has been with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, where she is currently a Professor. Her research interests include statistical signal processing theory and applications, with focus on multimedia security and biomedical signal processing and modeling.

Dr. Wang received the Outstanding Engineering Doctoral Student Award at the University of Connecticut. She was a co-recipient of the *EURASIP Journal on Applied Signal Processing* Best Paper Award in 2004 and the IEEE Signal Processing Society Best Paper Award in 2005. She is the Chair and Founder of the IEEE Signal Processing Chapter at Vancouver. She is an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING.