Efficient quantum circuit for singular-value thresholding

Bojia Duan,* Jiabin Yuan,† Ying Liu, and Dan Li

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics,
No. 29 Jiangjun Avenue, 211106 Nanjing, China

A singular-value thresholding (SVT) operation is a fundamental core module of many mathematical models in computer vision and machine learning, particularly for many nuclear norm minimizing-based problems. A quantum SVT (QSVT) algorithm was proposed in Phys. Rev. A. **96**, 032301 (2017) to solve an image-classification problem. This algorithm runs in $O[\log_2 (pq)]$, an exponential speed improvement over the classical algorithm, which runs in $O[\text{poly}(pq)]$. In this paper, we design a scalable quantum circuit for a QSVT. The quantum circuit is designed with $O[\log_2(pq/\varepsilon)]$ qubits and $O[\text{poly}\log_2(1/\varepsilon)]$ quantum gates in terms of error $O(\varepsilon)$. We also show that a high-probability and high-fidelity output can be obtained in one iteration of the quantum circuit. The quantum circuit for a QSVT implies a tempting possibility for experimental realization on a quantum computer. Finally, we propose a small-scale quantum circuit for a QSVT. We numerically simulate and demonstrate the performance of this circuit, verifying its capability to solve the intended SVT.

I. INTRODUCTION

Quantum computing has been shown to perform significantly better than classical computing at certain computational tasks, especially in the emerging interdisciplinary field of quantum machine learning [1–4]. To show its superiority, remarkable quantum algorithms help a range of classical algorithms achieve a speedup increase [5–12]. Shor's algorithm for factoring and Grover's algorithm for search are typical algorithms that can achieve exponential and quadratic speed increase, respectively [5,6]. In 2009, Harrow, Hassidim, and Lloyd (HHL) proposed an algorithm for solving linear systems of equations [10]. This algorithm offers an exponential speed over its classical counterparts by calculating the expectation value of an operator associated with the solution of the linear equation under certain circumstances. Considering that a linear system is the center of various areas in science and engineering, the HHL algorithm guarantees widespread applications [13–15]. Inspired by the idea of the HHL algorithm, other fruitful quantum machine learning algorithms are proposed [16–20]. Moreover, experimentalists aim to implement the HHL algorithm on a quantum computer. Therefore, research on the numerical theoretical simulation and experimental realization of the algorithm has recently emerged [21–24].

In many mathematical models in computer vision and machine learning, there is a fundamental core module known as singular-value thresholding (SVT). In particular, the SVT method has been widely adopted to solve many nuclear norm minimizing (NNM) -based problems, such as matrix completion, matrix denoising, and robust principal component analysis (RPCA). Therefore, the SVT method can be applied to many applications, such as image extraction, image coloriza-

tion, image denoising, image inpainting, and motion capture date recovery. However, NNM usually requires the iterative application of singular value decomposition (SVD) for SVT, and the computational cost of SVD may be too expensive to handle data with high dimension, e.g., high-resolution images. To speed up the process of SVT, we proposed a quantum SVT (QSVT) algorithm that can execute the SVT operator exponentially faster than its classical counterparts. The QSVT algorithm is used as a subroutine to accelerate an image classifier support matrix machine (SMM) [25].

In this article, we conduct a more specific discussion regarding the algorithm, and we design a quantum circuit for a QSVT that can be applied on a universal quantum computer. And the scalable quantum circuit requires $O[\log_2(pq/\varepsilon)]$ qubits and $O[\text{poly}\log_2(1/\varepsilon)]$ quantum gates with error $O(\varepsilon)$. Reference [25] shows that the QSVT algorithm is based on the HHL algorithm, which consists of two core subroutines, namely phase estimation and controlled rotation. Phase estimation outputs the eigenvalues of the input matrix **A** and decomposes the input vector $|\psi_{A_0}\rangle$ in the eigenbasis of **A**. Innovating and implementing controlled rotation are the key of HHL-based algorithms. Herein, we add an important missing piece to the algorithm by developing the detailed circuit of the controlled rotation via the theoretical function in Ref. [25]. Specifically, we divide the controlled rotation into two unitary operations. The first unitary operation is $\mathbf{U}_{\sigma,\tau}$, which is used to compute the function of the eigenvalues of **A**. By implementing the Newton iteration method, these function values can be stored in the quantum basis states. The second unitary operation is $\mathbf{R}_y$, which is used to extract the values in the quantum basis state to the corresponding amplitudes of the basis states. Implementing $\mathbf{R}_y$ directly affects the probability of obtaining the final result and the fidelity of actual and ideal final states. To improve probability and fidelity, we introduce a parameter α in $\mathbf{R}_y$, which can be computed ahead of quantum circuit implementation. Moreover, we present an example of a

*deja@nuaa.edu.cn
†jbyuan@nuaa.edu.cn

small-scale circuit for the algorithm, and we execute the numerical simulation of the example. The result shows the capability of the quantum computer to solve the intended SVT, and the performance of the algorithm is discussed.

Our work has two major contributions. First, we design a quantum circuit for the QSVT algorithm, which makes it possible to implement the algorithm on a quantum computer. Second, by introducing the parameter α, which can be computed ahead of implementing the quantum circuit, in the circuit design of controlled rotation, high probability and high fidelity can be obtained. Our work based on the QSVT algorithm may also inspire the circuit design of other HHL-based algorithms.

The remainder of the paper is organized as follows: We give a brief overview of the QSVT algorithm in Sec. II. Section III puts forward the quantum circuit for the QSVT and analyzes the probability and fidelity. We propose an example in Sec. IV, and we present our conclusions in Sec. V.

II. REVIEW OF QSVT

In this section, we briefly review the QSVT problem and the key procedure of the QSVT algorithm. More detailed information can be found in Ref. [25].

A. QSVT problem

SVT is an algorithm based on the SVD of a matrix. Suppose the input of the SVT is a low-rank matrix $\mathbf{A}_0 \in \mathbb{R}^{p \times q}$ with singular-value decomposition $\mathbf{A}_0 = \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \mathbf{v}_k^T$, where $r \leqslant \min(p,q)$ is the rank of $\mathbf{A}_0$, and $\sigma_k (\sigma_1 > \cdots > \sigma_r > 0)$ are just the singular values of $\mathbf{A}_0$, with $\mathbf{u}_k$ and $\mathbf{v}_k$ being the left and right singular vectors. SVT solves the problem $\mathbf{S} = \mathcal{D}_\tau(\mathbf{A}_0) := \sum_{k=1}^{r} (\sigma_k - \tau)_+ \mathbf{u}_k \mathbf{v}_k^T$, where $(\sigma_k - \tau)_+ = \max(\sigma_k - \tau, 0)$ and $\tau \in (0, \sigma_1)$ [26]. The vectorizations of the matrices $\mathbf{A}_0$ and $\mathbf{S}$ are $\mathrm{vec}(\mathbf{A}_0^T) = \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \otimes \mathbf{v}_k$ and $\mathrm{vec}(\mathbf{S}^T) = \sum_{k=1}^{r} (\sigma_k - \tau)_+ \mathbf{u}_k \otimes \mathbf{v}_k$, which vary as the quantum states $|\psi_{\mathbf{A}_0}\rangle = 1/\sqrt{N_1} \sum_{k=1}^{r} \sigma_k |\mathbf{u}_k\rangle |\mathbf{v}_k\rangle$ and $|\psi_{\mathbf{S}}\rangle = 1/\sqrt{N_2} \sum_{k=1}^{r} (\sigma_k - \tau)_+ |\mathbf{u}_k\rangle |\mathbf{v}_k\rangle$, respectively, where $N_1 = \sum_{k=1}^{r} \sigma_k^2$ and $N_2 = \sum_{k=1}^{r} (\sigma_k - \tau)_+^2$. Therefore, the QSVT algorithm solves the transformed problem $\mathrm{vec}(\mathbf{S}^T) = \mathcal{D}_\tau[\mathrm{vec}(\mathbf{A}_0^T)]$ [25].

B. QSVT algorithm

Let $\mathbf{A} = \mathbf{A}_0 \mathbf{A}_0^\dagger$, therefore $\mathbf{A} = \sum_{k=1}^{r} \sigma_k^2 \mathbf{u}_k \mathbf{u}_k^T$. The QSVT algorithm is now shown as follows [25]:

Input. A quantum state $|\psi_{\mathbf{A}_0}\rangle$, a unitary $e^{i\mathbf{A}t_0}$, and a constant τ.

Output. A quantum state $|\psi_{\mathbf{S}}\rangle$.

Algorithm. $\mathbf{S} = \mathrm{QSVT}(\mathbf{A}_0, \tau)$. The procedure of the algorithm can be illustrated as a sequence of the unitary operations:

$$\mathbf{U}_{\mathrm{QSVT}} = (\mathbf{I}^a \otimes \mathbf{U}_{\mathrm{PE}}^\dagger)(\mathbf{U}_{\mathrm{cR}} \otimes \mathbf{I}^B)(\mathbf{I}^a \otimes \mathbf{U}_{\mathrm{PE}}), \quad (1)$$

where $\mathbf{U}_{\mathrm{PE}}$ and $\mathbf{U}_{\mathrm{cR}}$ are the unitary operations of "phase estimation" and "controlled rotation," which are shown in Eqs. (2) and (3), respectively, and $\mathbf{U}_{\mathrm{PE}}^\dagger$ represents the inverse of $\mathbf{U}_{\mathrm{PE}}$.

Equation (1) shows that the QSVT algorithm consists of two core subroutines, namely $\mathbf{U}_{\mathrm{PE}}$ and $\mathbf{U}_{\mathrm{cR}}$.

(i) The first core subroutine $\mathbf{U}_{\mathrm{PE}}$ is presented as follows:

$$\mathbf{U}_{\mathrm{PE}} = \mathbf{U}_{\mathrm{PE}}(\mathbf{A})$$
$$= (\mathbf{F}_{\mathbf{T}}^\dagger \otimes \mathbf{I}^B)\left(\sum_{\tau=0}^{T-1} |\tau\rangle\langle\tau|^C \otimes e^{i\mathbf{A}\tau t_0/T}\right)(\mathbf{H}^{\otimes t} \otimes \mathbf{I}^B),$$
$$(2)$$

where register C stores the estimated eigenvalues of a Hermite matrix $\mathbf{A}$, register B stores the input state $|\psi_{\mathbf{A}_0}\rangle$, $\mathbf{F}_{\mathbf{T}}^\dagger$ is the inverse quantum Fourier transform, and $\sum_{\tau=0}^{T-1} |\tau\rangle\langle\tau|^C \otimes e^{i\mathbf{A}\tau t_0/T}$ is the conditional Hamiltonian evolution [10]. The Hermite matrix $\mathbf{A}$ determines on which eigenspace the quantum algorithm is implemented. Note that $\mathbf{A} = \mathbf{A}_0 \mathbf{A}_0^\dagger$, where $\mathbf{u}_k$ are the eigenvectors of $\mathbf{A}$ and the corresponding eigenvalues are $\lambda_k = \sigma_k^2$. By taking a partial trace of $|\psi_{\mathbf{A}_0}\rangle\langle\psi_{\mathbf{A}_0}|$, the density matrix that represents $\mathbf{A}$ can be obtained [17]: $\mathrm{tr}_2(|\psi_{\mathbf{A}_0}\rangle\langle\psi_{\mathbf{A}_0}|) = 1/N_1 \sum_{k=1}^{r} \sigma_k^2 |\mathbf{u}_k\rangle\langle\mathbf{u}_k| = \mathbf{A}/\mathrm{tr}\mathbf{A}$.

(ii) The second core subroutine $\mathbf{U}_{\mathrm{cR}}$ aims to "extract" and then "reassign" the proportion of each eigenstate in the superposition $|\psi_{\mathbf{A}_0}\rangle = 1/\sqrt{N_1} \sum_{k=1}^{r} \sigma_k |\mathbf{u}_k\rangle |\mathbf{v}_k\rangle$. In particular, $\mathbf{U}_{\mathrm{cR}}$ helps change the probability amplitude of each basic state $|\mathbf{u}_k\rangle |\mathbf{v}_k\rangle$ from σ_k to $(\sigma_k - \tau)_+$ via a transformation $(\sigma_k - \tau)_+ = \sigma_k \times \frac{(\sqrt{\sigma_k^2} - \tau)_+}{\sqrt{\sigma_k^2}}$. Without loss of generality, $\mathbf{U}_{\mathrm{cR}}$ is defined as follows: if $\sqrt{z} > \tau$,

$$|0\rangle|z\rangle \rightarrow \left(\frac{\gamma(\sqrt{z} - \tau)}{\sqrt{z}}|1\rangle + \sqrt{1 - \frac{\gamma^2(\sqrt{z} - \tau)^2}{z}}|0\rangle\right)|z\rangle;$$
$$(3)$$

otherwise do nothing.

III. QSVT CIRCUIT

In this section, we further study the QSVT algorithm [25] based on the quantum circuit model. It enables the quantum computer to solve many NNM-based problems. First, we describe the overview model of the quantum circuit for the QSVT. Second, we investigate in depth the realization of controlled rotation via the quantum circuit, which involves the computation of $\mathbf{U}_{\sigma,\tau}$ and $\mathbf{R}_y$. In the stage of $\mathbf{U}_{\sigma,\tau}$, we introduce Newton's method and simplify the Newton iteration function in terms of an intermediate variable z_k. We also reduce the number of Newton iterations with the help of a magic number R. In the stage of $\mathbf{R}_y$, we introduce an adjustable parameter α, and we demonstrate that the value of α can be computed ahead of implementing the quantum circuit to ensure a high-probability and high-fidelity readout.

The overview of the circuit for solving the QSVT is shown in Fig. 1. A solution of the QSVT with error $O(\varepsilon)$ can then be obtained. We omit the ancilla $|\tau\rangle$ in the following register presentation because it remains the same during the procedure of the quantum circuit. The detailed quantum circuit is presented as follows.

(i) Prepare the quantum registers in the state

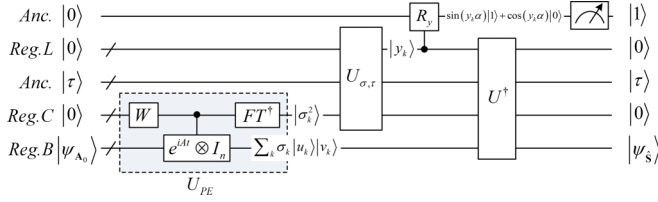$$|\psi_1\rangle = |0\rangle|0\rangle^L|0\rangle^C|\psi_{\mathbf{A}_0}\rangle^B, \quad (4)$$

FIG. 1. Overview of the quantum circuit for solving the transformed SVT. Wires with "/" represent the groups of qubits. The norms of the quantum states are omitted for convenience.

where the input matrix $\mathbf{A_0}$ has been prepared quantum mechanically as a quantum state $|\psi_{\mathbf{A_0}}\rangle$ stored in the quantum register B. The number of qubits in the register B is $b = O[\log_2(pq)]$.

(ii) Perform the unitary operation $\mathbf{U}_{\text{PE}}(\mathbf{A})$ on the state $|0\rangle^C |\psi_{\mathbf{A_0}}\rangle^B$. Recall $|\psi_{\mathbf{A_0}}\rangle = 1/\sqrt{N_1} \sum_{k=1}^{r} \sigma_k |\mathbf{u}_k\rangle |\mathbf{v}_k\rangle$, where $|\mathbf{u}_k\rangle$ are the eigenstates of $\mathbf{A}$, and σ_k^2 are the corresponding eigenvalues. As shown in Ref. [25], we have the state

$$|\psi_2\rangle = \frac{1}{\sqrt{N_1}} |0\rangle |0\rangle^L \sum_{k=1}^{r} \sigma_k |\sigma_k^2\rangle^C |\mathbf{u}_k\rangle |\mathbf{v}_k\rangle^B. \qquad (5)$$

Let the efficient condition number of $\mathbf{A}$ be κ, such that $\sigma_k^2 \in [1/\kappa, 1]$. The number of qubits in the register C is $n = O(\log_2 \kappa)$.

(iii) The unitary $\mathbf{U}_{\sigma,\tau}$ converts the eigenvalues $|\sigma_k^2\rangle$ stored in the register C to the intermediate result $|y_k\rangle$ stored in the register L, where $y_k = (1 - \tau/\sigma_k)_+ \in [0, 1)$. The state is

$$|\psi_3\rangle = \frac{1}{\sqrt{N_1}} |0\rangle \sum_{k=1}^{r} \sigma_k |y_k\rangle^L |\sigma_k^2\rangle^C |\mathbf{u}_k\rangle |\mathbf{v}_k\rangle^B. \qquad (6)$$

According to the ideas of Ref. [27], Newton iteration can be used to realize $\mathbf{U}_{\sigma,\tau}$ (see Sec. III A). The number of qubits in the register L and the ancilla $|\tau\rangle$ are both $d = O(\log_2 \kappa)$.

(iv) To realize $\mathbf{R}$: $|0\rangle |y_k\rangle \mapsto (y_k|1\rangle + \sqrt{1 - y_k^2}|0\rangle)|y_k\rangle$, i.e., to "extract" the value of y_k in register L to the amplitude of the ancilla qubit, we introduce a unitary $\mathbf{R}_y$ with parameter α (see Sec. III B) to approximate $\mathbf{R}$:

$$\mathbf{R}_y|0\rangle |y_k\rangle = [\sin(y_k\alpha)|1\rangle + \cos(y_k\alpha)|0\rangle]|y_k\rangle, \quad 0 < y_k < 1. \qquad (7)$$

Subsequently, $\mathbf{R}_y$ is applied to the ancilla qubit on the top of the circuit and controlled by the register L. We obtain the state

$$|\psi_4\rangle = \frac{1}{\sqrt{N_1}} \sum_{k=1}^{r} \sigma_k [\sin(y_k\alpha)|1\rangle \\ + \cos(y_k\alpha)|0\rangle]|y_k\rangle^L |\sigma_k^2\rangle^C |\mathbf{u}_k\rangle |\mathbf{v}_k\rangle^B. \qquad (8)$$

(v) Uncompute the registers L, C, and B, and remove the register L and C, to obtain

$$|\psi_5\rangle = \frac{1}{\sqrt{N_1}} \sum_{k=1}^{r} \sigma_k [\sin(y_k\alpha)|1\rangle + \cos(y_k\alpha)|0\rangle]|\mathbf{u}_k\rangle |\mathbf{v}_k\rangle^B. \qquad (9)$$
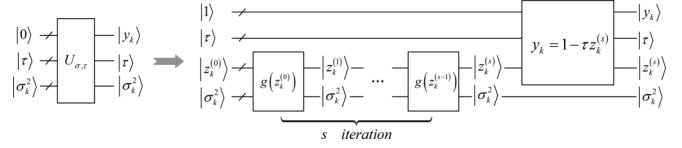

FIG. 2. A quantum circuit of the unitary $\mathbf{U}_{\sigma,\tau}$.

(vi) Measure the top ancilla bit. If the result returns to 1, then the register B of the system collapses to the final state,

$$|\psi_{\hat{\mathbf{S}}}\rangle = \frac{1}{\sqrt{N_\alpha}} \sum_{k=1}^{r} \sigma_k \sin(y_k\alpha)|\mathbf{u}_k\rangle |\mathbf{v}_k\rangle^B, \qquad (10)$$

where $N_\alpha = \sum_{k=1}^{r} \sigma_k^2 \sin^2(y_k\alpha)$.

In summary, the procedure of the quantum circuit can be illustrated as follows:

$$\tilde{\mathbf{U}}_{\text{QSVT}} = (\mathbf{I}^{aLa} \otimes \mathbf{U}_{\text{PE}}^\dagger)(\mathbf{I}^a \otimes \mathbf{U}_{\sigma,\tau}^\dagger \otimes \mathbf{I}^B)(\mathbf{R}_y \otimes \mathbf{I}^{aCB})$$
$$\times (\mathbf{I}^a \otimes \mathbf{U}_{\sigma,\tau} \otimes \mathbf{I}^B)(\mathbf{I}^{aLa} \otimes \mathbf{U}_{\text{PE}}). \qquad (11)$$

A. Computation of $\mathbf{U}_{\sigma,\tau}$

We now deal with the detailed quantum circuit of the unitary $\mathbf{U}_{\sigma,\tau}$, that is, we deal with the function of the eigenvalues σ_k^2 of $\mathbf{A}$: $y_k = (1 - \tau/\sqrt{\sigma_k^2})_+ \in [0, 1)$. Here Newton iteration is introduced to compute $y_k = 1 - \tau/\sqrt{\sigma_k^2}$ for $\sigma_k > \tau$. To simplify the quantum circuit design of $\mathbf{U}_{\sigma,\tau}$, an intermediate variable z_k is introduced such that $z_k = z(\sigma_k^2) = 1/\sqrt{\sigma_k^2}$ and $y_k = y(z_k) = 1 - \tau z_k$. Therefore, Newton iteration can be just used to compute the simplified function $z_k = z(\sigma_k^2) = 1/\sqrt{\sigma_k^2}$. From above, a quantum circuit of the unitary $\mathbf{U}_{\sigma,\tau}$ can be designed as shown in Fig. 2.

Specifically, the detailed quantum circuit design of $\mathbf{U}_{\sigma,\tau}$ can be divided into two parts. The first part is the rightmost unitary $y_k = 1 - \tau z_k$ as shown in Fig. 2, and the second part is the leftmost Newton iterations.

Obviously, the first part can be simply realized via the quantum circuits for addition and multiplication, which have been studied in Refs. [28,29]. The corresponding quantum circuit is shown in Fig. 3. The number of qubits is $O(d)$ and the number of quantum operations for implementing addition and multiplication is $O(d^2)$.

Let us turn to the second part. A method for solving $z(\sigma_k^2) = 1/\sqrt{\sigma_k^2}$ has been presented in Ref. [30]. It provides the quantum circuit for the initial state $|z_k^{(0)}\rangle = |2^{\lfloor \frac{w-1}{2} \rfloor}\rangle$, where $w \in N$ and $2^{1-w} > x > 2^{-w}$, and it presents the idea for
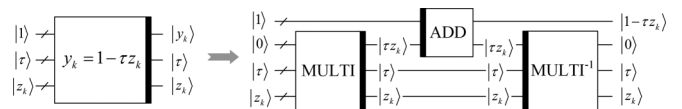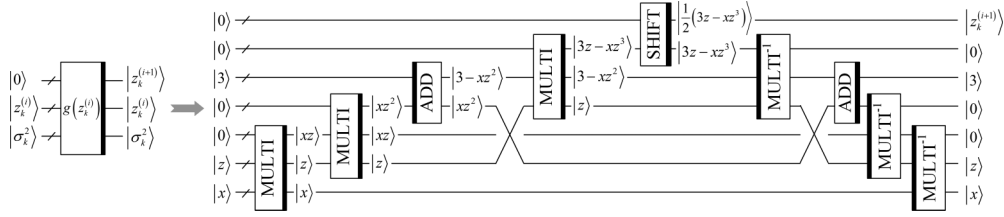

FIG. 3. Quantum circuit for computing $y_k = 1 - \tau z_k$. The black strip on the right side of the rectangle represents a unitary operation, and the black strip on the left represents the inverse of the corresponding unitary operation.

FIG. 4. A quantum circuit of one Newton iteration for computing $g(z_k^{(i)})$.

solving the iteration function in terms of an abstraction of an elementary module. Here we perform additional research on the Newton method. Concretely, we give the detailed quantum circuit of the Newton iteration in terms of basic elementary gates, and we introduce a different initial state that helps to reduce the iteration steps.

Applying the Newton method to the function $f(z_k) = 1/z_k^2 - \sigma_k^2$, we can obtain the iteration function

$$
\begin{aligned}
z_k^{(i+1)} = g(z_k^{(i)}) &= z_k^{(i)} - \frac{f(z_k^{(i)})}{f'(z_k^{(i)})} = z_k^{(i)} - \frac{(z_k^{(i)})^{-2} - \sigma_k^2}{-2(z_k^{(i)})^{-3}} \\
&= \frac{1}{2}\left[3z_k^{(i)} - \sigma_k^2(z_k^{(i)})^3\right],
\end{aligned}
\tag{12}
$$

where $i = 1, 2, \ldots, s$.

The detailed quantum circuit of $g(z_k^{(i)})$ is presented in Fig. 4. Inevitably, four extra quantum registers are needed for the inverse of the unitary operators, and one more ancilla register is needed for storing the output $z_k^{(i+1)}$ in each iteration $g(z_k^{(i)})$. As the circuit of the iteration is composed of basic operations (i.e., addition, multiplication, and shift operations), each iterative step requires $O(n+d)$ qubits and $O[\text{poly}(n+d)]$ quantum operations, where the degree of the polynomial is no more than 3 according to Refs. [28,29].

With indispensable ancilla registers, a quantum circuit for computing $z_k^{(s)}$ after s Newton iterations is designed as in Fig. 5. Therefore, the number of qubits for implementing the Newton iteration is $O(n+sd)$, and the number of quantum operations is $O[s \cdot \text{ploy}(n+d)]$.

As in Fig. 5, after each iteration, at least one extra register is needed for storing the intermediate output $z_k^{(i)}$. Therefore, after s iterations, $s-1$ garbage registers are produced.

To reduce the number of iterations, we choose a magic number R in Ref. [31] rather than the initial state in Ref. [30]. The magic number R produces the first approximation of the initial state by $X = R - (X \gg 1)$, which follows the

IEEE 754 floating-point format [32]. It helps Newton's method run only one or two iterations and output a more precise approximation (see Appendix A). Then the number of qubits for implementing the Newton iteration can be reduced to $O(n+d)$, and the number of quantum operations can be reduced to $O[\text{poly}(n+d)]$. The corresponding quantum circuit is shown in Fig. 6.

Now we turn to the error analysis of the quantum circuit of the unitary $\mathbf{U}_{\sigma,\tau}$. It consists of two parts. The first is the error e_s caused by Newton's iteration, the second is the roundoff error $\hat{e}_s$ caused by truncating the result of each iteration to d qubits of accuracy before passing it to the next iteration.

According to Eq. (12), we have

$$
\begin{aligned}
g(z_k^{(i)}) - \frac{1}{\sqrt{\sigma_k^2}} &= \frac{1}{2}\left[3z_k^{(i)} - \sigma_k^2(z_k^{(i)})^3\right] - \frac{1}{\sqrt{\sigma_k^2}} \\
&= -\frac{1}{2}\left(z_k^{(i)} - \frac{1}{\sqrt{\sigma_k^2}}\right)^2 \sqrt{\sigma_k^2}(z_k^{(i)}\sqrt{\sigma_k^2} + 2).
\end{aligned}
\tag{13}
$$

The last quantity is nonpositive for $z_k^{(i)} \geqslant 0$. Similar to the equation in Ref. [30], the error e_s satisfies

$$
\begin{aligned}
e_s &:= \left|z_k^{(s)} - \frac{1}{\sqrt{\sigma_k^2}}\right| = \frac{1}{2}e_{s-1}^2\sqrt{\sigma_k^2}(z_k^{(s-1)}\sqrt{\sigma_k^2} + 2) \\
&\leqslant \frac{3}{2}\sqrt{\sigma_k^2}e_{s-1}^2 \leqslant \frac{2}{3\sqrt{\sigma_k^2}}\left(\frac{3}{2}\sqrt{\sigma_k^2}e_0\right)^{2^s}.
\end{aligned}
\tag{14}
$$

The initial error e_0 satisfies $\sqrt{\sigma_k^2}e_0 \leqslant 1/8$ for the initial state derived from the magic number R [31], which is better than
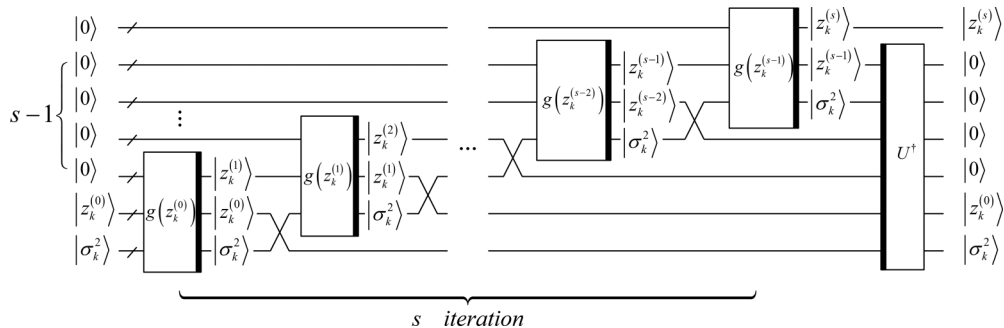
FIG. 5. A quantum circuit for computing $z_k^{(s)}$ with s iterations.
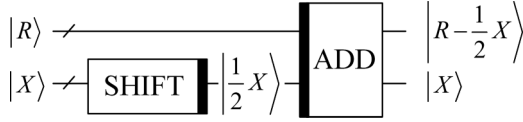
FIG. 6. A quantum circuit for computing the initial state $z_k^{(0)}$. X is presented as initial $z_k^{(0)}$, which follows the IEEE 754 floating-point format.

$\sqrt{\sigma_k^2} e_0 \leqslant 1/2$ for the traditional initial state $|z_k^{(0)}\rangle = |2^{\lfloor \frac{w-1}{2} \rfloor}\rangle$ [30].

As shown in Ref. [30], the truncation error $\hat{e}_s$ satisfies

$$\hat{e}_s := |\hat{z}^{(s)} - z^{(s)}| \leqslant 2^{1-d} \left(\frac{3}{2}\right)^s. \tag{15}$$

To sum up, with the initial state in Ref. [30], the number of the iteration steps is $s = O(\log_2 d)$. Then the error caused by the unitary $\mathbf{U}_{\sigma,\tau}$ is

$$\left| z_k^{(s)} - \frac{1}{\sqrt{\sigma_k^2}} \right| \leqslant e_s + \hat{e}_s \leqslant \frac{2}{3\sqrt{\sigma_k^2}} \left(\frac{3}{4}\right)^{2^s} + 2^{1-d} \left(\frac{3}{2}\right)^s$$

$$\leqslant \frac{2}{3\sqrt{\sigma_k^2}} \left(\frac{3}{4}\right)^d + 2^{1-d} 2^{\log_2 d + 1}$$

$$\leqslant \frac{2\sqrt{\kappa}}{3} \left(\frac{3}{4}\right)^d + 2^{2-d} d. \tag{16}$$

And with the initial state in Ref. [31], the number of iteration steps is $s = 1$ or 2. Then the errors caused by the unitary $\mathbf{U}_{\sigma,\tau}$ are

$$\left| z_k^{(1)} - \frac{1}{\sqrt{\sigma_k^2}} \right| \leqslant e_1 + \hat{e}_1 \leqslant \frac{2}{3\sqrt{\sigma_k^2}} \left(\frac{3}{16}\right)^2 + 2^{1-d} \left(\frac{3}{2}\right)$$

$$\leqslant \frac{3\sqrt{\kappa}}{2^7} + \frac{3}{2^d} \tag{17}$$

and

$$\left| z_k^{(2)} - \frac{1}{\sqrt{\sigma_k^2}} \right| \leqslant e_2 + \hat{e}_2 \leqslant \frac{2}{3\sqrt{\sigma_k^2}} \left(\frac{3}{16}\right)^4 + 2^{1-d} \left(\frac{3}{2}\right)^2$$

$$\leqslant \frac{27\sqrt{\kappa}}{2^{15}} + \frac{9}{2^{d+1}}, \tag{18}$$

respectively.

B. Computation of $\mathbf{R}_y$

A sequence of rotations about the y axis can be used to implement the unitary $\mathbf{R}_y$ [27], where $\mathbf{R}_y(2\theta)|0\rangle = \sin\theta|1\rangle + \cos\theta|0\rangle$. Here we introduce a parameter α, which can be used to improve the probability and accuracy of obtaining the final state (see Sec. III D). Assume that we have obtained a d-qubit state $|y_k\rangle \triangleq |\theta\rangle$. Consider the binary representation of θ: $\theta = 0.\theta_1 \cdots \theta_d = \sum_{j=1}^d \theta_j 2^{-j}, (\theta_j \in \{0,1\})$. We have

$$\mathbf{R}_y(2\alpha\theta) = e^{-i\alpha\theta Y} = \prod_{j=1}^d e^{-iY\alpha\theta_j/2^j} = \prod_{j=1}^d \mathbf{R}_y^{\theta_j}(2^{1-j}\alpha)$$

$$= R_y^{\theta_1}(\alpha) R_y^{\theta_2}(\alpha/2) \cdots R_y^{\theta_d}(\alpha/2^{d-1}), \tag{19}$$

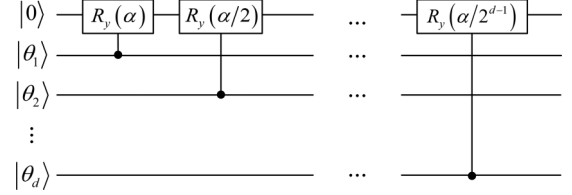FIG. 7. Quantum circuit of the unitary $\mathbf{R}_y$.

where Y is the Pauli Y operator. The quantum circuit of $\mathbf{R}_y$ is shown in Fig. 7.

C. Complexity analysis

We now analyze the space and time resources required for the whole quantum circuit. The numbers of qubits in register B, C, and L are $b = O[\log_2(pq)]$, $n = O(\log_2 \kappa)$, and $d = O(\log_2 \kappa)$, respectively. Ancilla qubits in the computation of $\mathbf{U}_{\sigma,\tau}$ take up the most space in the quantum circuit, and the number of ancilla qubits is $O(n + d)$. Typically, the condition number κ is taken as $\kappa = O(1/\varepsilon)$. Therefore, the total number of qubits required in the quantum circuit is $O[\log_2(pq/\varepsilon)]$.

Turning to the time cost, the phase estimation requires $O(n^2)$ operations and one call to the controlled-$\mathbf{A}$ black box [1]. The number of quantum gates in the computation of $\mathbf{U}_{\sigma,\tau}$ is $O[\text{poly}(n + d)]$, and the number of gates in the computation of $\mathbf{R}_y$ is $O(d)$. Therefore, the total number of gates required in the quantum circuit is $O[\text{poly} \log_2(1/\varepsilon)]$.

In summary, the number of qubits required by the circuit is $O[\log_2(pq/\varepsilon)]$ and the number of quantum operations used by the circuit is a low degree polynomial in $\log_2(1/\varepsilon)$.

D. Probability and fidelity analysis

We now analyze the probability of obtaining the final result and the fidelity of the ideal and actual outputs. Equations (1) and (11) show that the quantum circuit $\tilde{\mathbf{U}}_{\text{QSVT}}$ is a unitary approximation of the quantum algorithm $\mathbf{U}_{\text{QSVT}}$. Without loss of generality, we assume that there is no error in any step of $\mathbf{U}_{\text{QSVT}}$ and $\tilde{\mathbf{U}}_{\text{QSVT}}$. Therefore, the unitary $\mathbf{R}_y$ in $\tilde{\mathbf{U}}_{\text{QSVT}}$ dominates the probability and the fidelity in the quantum circuit.

The probability of obtaining the final result $|\psi_{\hat{\mathbf{S}}}\rangle$ in Eq. (10) can be calculated via Eq. (9):

$$P(\alpha) = \frac{N_\alpha}{N_1} = \frac{\sum_{k=1}^r \sigma_k^2 \sin^2(y_k \alpha)}{\sum_{k=1}^r \sigma_k^2}. \tag{20}$$

The fidelity of the actual and ideal final results can be calculated via the inner product of the output $|\psi_{\hat{\mathbf{S}}}\rangle$ and the theoretical output $|\psi_{\mathbf{S}}\rangle = 1/\sqrt{N_2} \sum_{k=1}^r (\sigma_k - \tau)_+ |\mathbf{u}_k\rangle |\mathbf{v}_k\rangle = 1/\sqrt{N_2} \sum_{k=1}^r \sigma_k y_k |\mathbf{u}_k\rangle |\mathbf{v}_k\rangle$:

$$F(\alpha) = \langle \psi_{\hat{\mathbf{S}}} | \psi_{\mathbf{S}} \rangle = \frac{1}{\sqrt{N_2 N_\alpha}} \sum_{k=1}^r \sigma_k^2 y_k \sin(y_k \alpha)$$

$$= \frac{\sum_{k=1}^r \sigma_k^2 y_k \sin(y_k \alpha)}{\sqrt{\sum_{k=1}^r \sigma_k^2 y_k^2 \times \sum_{k=1}^r \sigma_k^2 \sin^2(y_k \alpha)}}. \tag{21}$$

Typically, α can be obtained by the method in Ref. [33] such that $y_k \alpha \approx \sin^{-1}(y_k)$. Therefore, the probability of obtaining
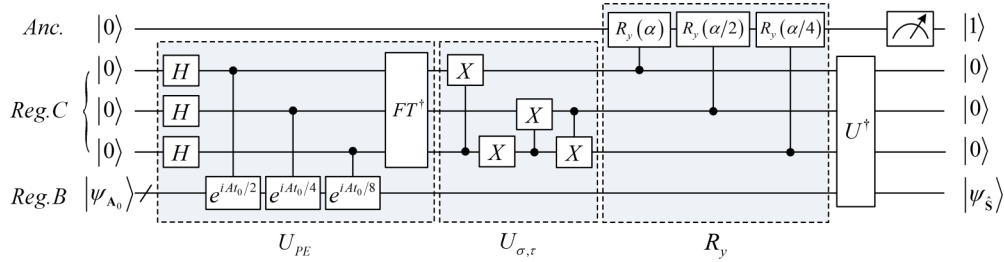
FIG. 8. Example quantum circuit for solving the SVT. $U^{\dagger}$ represents the inverse of all the operations before of $\mathbf{R}_y$. Each simplified unitary $e^{i\mathbf{A}t_0/2^j}$ ($j = 1,2,3$) in the figure represents the operation $e^{i\mathbf{A}t_0/2^j} \otimes \mathbf{I}_n$.

the final result is

$$P_1 \approx \frac{\sum_{k=1}^{r} \sigma_k^2 y_k^2}{\sum_{k=1}^{r} \sigma_k^2}. \tag{22}$$

And the fidelity of the output approximates to 1. But the method in Ref. [33] involves quantum circuits for exponentiation computations and the results in excessive time and space consumptions.

Here, we introduce a method to compute α ahead of implementing the QSVT quantum circuit to avoid extra circuits involved in Ref. [33]. To ensure high probability and high fidelity, we introduce function G of α:

$$\begin{aligned} G(\alpha) &= \sqrt{P(\alpha)} \times F(\alpha) \\ &= \sqrt{\frac{N_\alpha}{N_1}} \times \frac{1}{\sqrt{N_2 N_\alpha}} \sum_{k=1}^{r} \sigma_k^2 y_k \sin(y_k \alpha) \\ &= \frac{\sum_{k=1}^{r} \sigma_k^2 y_k \sin(y_k \alpha)}{\sqrt{\sum_{k=1}^{r} \sigma_k^2 \times \sum_{k=1}^{r} \sigma_k^2 y_k^2}}, \end{aligned} \tag{23}$$

and to find α that maximizes $G(\alpha)$, i.e., to solve the problem $\arg\max_\alpha G(\alpha)$. This problem is transformed to solve the following equation:

$$G'(\alpha) = \frac{\sum_{k=1}^{r} \sigma_k^2 y_k^2 \cos(y_k \alpha)}{\sqrt{\sum_{k=1}^{r} \sigma_k^2 \times \sum_{k=1}^{r} \sigma_k^2 y_k^2}} = 0. \tag{24}$$

A series of methods, such as gradient descent, Newton's method, and evolutionary algorithms, can be used to solve Eq. (24). Given that the problem to be solved is not convex, these iterative algorithms can only output a locally optimal solution. Taylor's series can also be used to solve this problem and obtain an approximate solution.

Instead of using the aforementioned methods, we select an "intuitive" method to compute an approximate solution $\tilde{\alpha}$ for this problem. Recalling that $y_k = (1 - \tau/\sigma_k)_+$, we have $y_1 > y_2 \geqslant \cdots \geqslant y_r$ as $\sigma_1 > \sigma_2 > \cdots > \sigma_r > 0$. The period of $\cos(y_k \alpha)$ is $2\pi/y_k$, thereby satisfying $2\pi/y_1 < 2\pi/y_2 \leqslant \cdots \leqslant 2\pi/y_r$. We now consider the case in which the value α satisfies $y_1 \alpha \in [0, \pi/2]$, therefore $0 \leqslant \cos(y_1\alpha) < \cos(y_2\alpha) \leqslant \cdots \leqslant \cos(y_r\alpha)$.

Equation (24) shows that $\sum_{k=1}^{r} \sigma_k^2 y_k^2 \cos(y_k\alpha) = 0$. Therefore,

$$\begin{aligned} \sigma_1^2 y_1^2 \cos(y_1\alpha) &= -\sigma_2^2 y_2^2 \cos(y_2\alpha) - \cdots \\ &\quad - \sigma_r^2 y_r^2 \cos(y_r\alpha) \leqslant 0, \end{aligned} \tag{25}$$

and considering $\cos(y_1\alpha) \in [0,1]$, we obtain the approximate solution $\tilde{\alpha} = \frac{\pi}{2y_1} = \frac{\pi}{2(1-\tau/\sigma_1)}$. Note that $\tilde{\alpha}$ is close to being optimal when $\sum_{k=1}^{r} \sigma_k^2 y_k^2 \cos(y_k\alpha)$ is dominated by $\sigma_1^2 y_1^2 \cos(y_1\alpha)$, because in this case the derivative is approximately zero. That is, $\tilde{\alpha}$ is most effective when σ_1 dominates most of the singular values.

Although the approximate solution is not the optimal solution, it has two advantages. First, it has a simpler expression and can be computed more efficiently compared with iterative algorithms. Second, this "intuitive" solution is only based on the maximized singular value of $\mathbf{A}_0$, i.e., prior knowledge of $\mathbf{A}_0$ is only the maximum singular value σ_1 instead of all singular values of $\mathbf{A}_0$ compared with Taylor's series method (see Appendix B). In Appendix C, we perform two experiments to analyze the probability and fidelity. The first experiment shows that $\tilde{\alpha}$ is likely to be a good solution to ensure high probability and fidelity. The second experiment shows how the input matrix and the hyperparameter τ impact the probability and fidelity readout.

IV. EXAMPLE

In this section, we design and implement a numerical simulation experiment of a small-scale QSVT circuit and analyze the results. The purpose for this example is to illustrate the algorithm and for potential experimental implementation using currently available resources.

We demonstrate a proof-of-principle experiment of the QSVT algorithm shown in Fig. 8. This simple quantum circuit solves a meaningful instance of the problem, that is, to perform the SVT on a 2×3 dimension matrix $\mathbf{A}_0$. For the numerical example, we select different inputs of $\mathbf{A}_0$ all satisfying that the singular values of $\mathbf{A}_0$ are $\sigma_1 = 2, \sigma_2 = 1$. Matrix $\mathbf{A}_0$ is selected such that the eigenvalues of $\mathbf{A}$ are 4 and 1, which can be exactly encoded with three qubits in register C. Without loss of generality, let $\tau = 1/2$. This allows us to optimize the subroutine $\mathbf{U}_{\sigma,\tau}$ (in Fig. 1) without involving register L and ancilla qubits $|\tau\rangle$. The input state of register B is a normalized quantum state $|\psi_{\mathbf{A}_0}\rangle$.

The initial quantum system is $|0000b\rangle$. Phase estimation generates the states that encode the eigenvalues of $\mathbf{A}$ in register C, and subsequently the system is in the following superposition: $\frac{1}{\sqrt{5}}(2|0100\rangle|\mathbf{u}_1\rangle|\mathbf{v}_1\rangle + |0001\rangle|\mathbf{u}_2\rangle|\mathbf{v}_2\rangle)$. The mapping of the operator $\mathbf{U}_{\sigma,\tau}$ is $|100\rangle^C \mapsto |110\rangle^C$ and $|001\rangle^C \mapsto |100\rangle^C$, where the outputs $|110\rangle$ and $|100\rangle$ can be interpreted as the encodings $2^3(1 - \tau/\sigma_1) = 6$ and $2^3(1 - \tau/\sigma_2) = 4$, respectively. After the $\mathbf{U}_{\sigma,\tau}$ operator, the

system becomes $\frac{1}{\sqrt{5}}(2|0110\rangle|\mathbf{u}_1\rangle|\mathbf{v}_1\rangle + |0100\rangle|\mathbf{u}_2\rangle|\mathbf{v}_2\rangle)$. Then we use $|2^3(1 - \tau/\sigma_k)\rangle$ in register C as the control register to execute a sequence of Pauli Y rotations $\mathbf{R}_y$ on the ancilla qubit $|0\rangle$ with $\alpha = \frac{\pi}{2/(1-\tau/\sigma_1)} = 2.0944$. Taking the inverse of all the operations before $\mathbf{R}_y$, and measuring the ancilla qubit to be $|1\rangle$, the system becomes

$$\frac{1}{\sqrt{N_\alpha}}|1000\rangle(1.9999|u_1\rangle|v_1\rangle + 0.8660|u_2\rangle|v_2\rangle), \quad (26)$$

where $N_\alpha = 4.7495$.

The theoretical result is as follows:

$$\frac{1}{\sqrt{N_2}}\sum_{k=1}^{2}(\sigma_k - \tau)|1000\rangle|\mathbf{u}_k\rangle|\mathbf{v}_k\rangle$$

$$= \frac{1}{\sqrt{10}}|1000\rangle(3|\mathbf{u}_1\rangle|\mathbf{v}_1\rangle + |\mathbf{u}_2\rangle|\mathbf{v}_2\rangle). \quad (27)$$

We then compute the probability and the fidelity according to Eqs. (20) and (21). The probability of measuring the ancilla qubit to be 1 is $P(\alpha) = 0.9499$, and the fidelity is $F(\alpha) = 0.9962$.

Our results may motivate experimentalists to verify this result by implementing the quantum circuit with the ability to address six or more qubits and execute basic quantum gates on their setups.

V. CONCLUSIONS

Nowadays, the quantum circuit model is the most popular and developed model for universal quantum computation. We investigated further the QSVT algorithm that we proposed in Ref. [25] by providing the possibility to implement the algorithm on a quantum computer via the circuit model. The scalable quantum circuit is presented, in which the key subroutine of the controlled rotation is designed by introducing Newton's method and an adjustable parameter α. We simplified the Newton iteration function in terms of an intermediate variable, and we reduced the number of Newton iterations with the help of a magic initial state. Then we analyzed the space-time complexity of the designed quantum circuit, which shows that the number of qubits and gates required in the quantum circuit is $O[\log_2(pq/\varepsilon)]$ and $O[\text{poly}\log_2(1/\varepsilon)]$, respectively. Moreover, we provided two methods to compute the value of α to ensure a high-probability and high-fidelity readout, and we conducted numerical experiments. The numerical results show that under different inputs, our method can output high probability and high fidelity in one iteration of the quantum circuit. Furthermore, we present a small-scale circuit as an example to verify the algorithm. We hope that our research motivates experimentalists to conduct new investigations in quantum computation.

APPENDIX A: INITIAL STATE OF THE NEWTON'S METHOD

For the single-precision floating-point format, the magic number is chosen as a hexadecimal constant 0x5f375a86, and for the double-precision floating-point format, the magic number can be chosen as a hexadecimal constant 0x5fe6ec85e7de30da. As mentioned in Sec. III A, both magic numbers follow the IEEE 754 floating-point format. The test in Ref. [31] shows that the relative error can be around 0.00175 after 1 Newton iteration, and reduce to around $4.65437e - 004$ after 2 Newton iterations. Therefore, our quantum circuit only needs one or two iterations to get the high-precision approximation of z_k with the help of the magic initial state.

APPENDIX B: TAYLOR'S SERIES METHOD

Using Taylor's series method to solve Eq. (24), we have

$$G'(\alpha) = 0$$

$$\Rightarrow \sum_{k=1}^{r}\sigma_k^2\left(y_k^2 - \frac{y_k^4\alpha^2}{2!} + \frac{y_k^6\alpha^4}{4!} - \cdots\right.$$

$$\left. + (-1)^n\frac{y_k^{2n+2}\alpha^{2n}}{(2n)!} + o\left(y_k^{2n+3}\alpha^{2n+1}\right)\right) = 0. \quad (B1)$$

If the second-order approximation is selected, then

$$\sum_{k=1}^{r}\sigma_k^2\left(y_k^2 - \frac{y_k^4\alpha^2}{2} + o\left(y_k^5\alpha^3\right)\right) = 0$$

$$\Rightarrow \sum_{k=1}^{r}\sigma_k^2\left(y_k^2 - \frac{y_k^4\alpha^2}{2}\right) \approx 0$$

$$\Rightarrow \alpha \approx \sqrt{\frac{2\sum_{k=1}^{r}\sigma_k^2 y_k^2}{\sum_{k=1}^{r}\sigma_k^2 y_k^4}}. \quad (B2)$$

If the fourth-order approximation is selected, then

$$\sum_{k=1}^{r}\sigma_k^2\left(y_k^2 - \frac{y_k^4\alpha^2}{2} + \frac{y_k^6\alpha^4}{4!} + o\left(y_k^7\alpha^5\right)\right) = 0$$

$$\Rightarrow \sum_{k=1}^{r}\sigma_k^2\left(y_k^2 - \frac{y_k^4\alpha^2}{2} + \frac{y_k^6\alpha^4}{4!}\right) \approx 0$$

$$\Rightarrow \alpha \approx \sqrt{\frac{b - \sqrt{b^2 - 4ac}}{2a}}, \quad (B3)$$

where $a = \frac{1}{24}\sum_{k=1}^{r}\sigma_k^2 y_k^6$, $b = \frac{1}{2}\sum_{k=1}^{r}\sigma_k^2 y_k^4$, $c = \sum_{k=1}^{r}\sigma_k^2 y_k^2$.
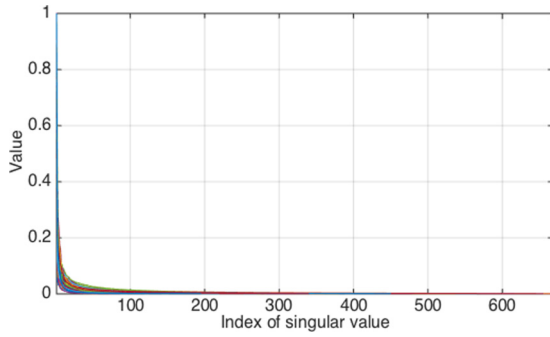
FIG. 9. Singular-value distributions of the 120 matrices.

APPENDIX C: NUMERICAL SIMULATIONS

1. Initialization

Recall that the applications of the SVT are mainly based on videos or pictures. Thus, we select 120 different inputs of matrix $\mathbf{A}_0^{(i)}(i = 1, 2, \ldots, 120)$, which are derived from random pictures. These matrices have been preprocessed in terms of the condition number and the ℓ_2 normalization. The efficient condition number κ of the matrix $\mathbf{A}^{(i)} = \mathbf{A}_0^{(i)} \mathbf{A}_0^{(i)\dagger}$ is set to 10^6, therefore only the singular values in the range of $[1/1000, 1]$ are taken into account. The ℓ_2 normalization $\|\mathbf{A}_0^{(i)}\|_2$ is set to 1, therefore the largest singular value $\sigma_1^{(i)}$ of each matrix is equal to 1. (Alternatively, the Frobenius norm can also be used to normalize these matrices.) The dimensionality of these matrices ranges from 200×300 to 2621×3995, and the rank of these matrices ranges from 55 to 668. The singular value distributions of the 120 matrices $\mathbf{A}_0^{(i)}$ are shown in Fig. 9. Typically, the largest singular value dominates most of the singular values, and only a few principal singular values of the matrix $\mathbf{A}_0^{(i)}$ are large, while the rest of the singular values are very small. In the following experiments, all the singular values of the 120 matrices are computed, and the probability and fidelity are then derived by Eqs. (20) and (21), respectively.

a. Experiment (1)

Figure 10 shows that the x axis represents the 120 inputs. Figures 10(a) and 10(b) show the probability and the fidelity in terms of different α, which are obtained by Taylor's method (shown by the blue dashed line) and our "intuitive" method (in red, "+"), respectively. The numerical results show that our "intuitive" method works as well as Taylor's method. The probability is almost the same in terms of $\tilde{\alpha}$ obtained by our method and Taylor's method. Moreover, fidelity performs slightly better when our method is used instead of Taylor's method. Both methods can help output the high probability and high fidelity in the context of the 120 different random inputs.

b. Experiment (2)

We now analyze which properties of the input matrix $\mathbf{A}_0^{(i)}$ and hyperparameter τ impact the probability and fidelity. Specifically, we select the hyperparameter τ from {0.001, 0.002, 0.005, 0.007, 0.009, 0.01, 0.02, 0.05, 0.07, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5}. Figure 11 shows how the probability and the fidelity changes with the rank of the input matrices. Here we pick up the pictures with significant features to show up the trends. The black dots represent the probability or fidelity in Figs. 11(a) and 11(b) respectively, and the blue solid lines represents the 3-order polynomial fit function generated via cftool in Matlab.

Figure 11 shows that the rank r of the input matrices would impact the probability and fidelity of the readout, but the effect is not significant. When τ is small, the probability and fidelity decrease while the rank of input matrices increases. As τ become large, the relationship of the probability and fidelity and the rank become unapparent. We can also see that the probability is more susceptible to rank than the fidelity. The results are reasonable. According to Eqs. (20) and (21), which derive the probability and the fidelity, the rank of the input matrix would affect the total number of summed items in the numerator and denominator at the same time. But a large amount of the items are very small since many singular values of the input matrix are small. Therefore, the probability and
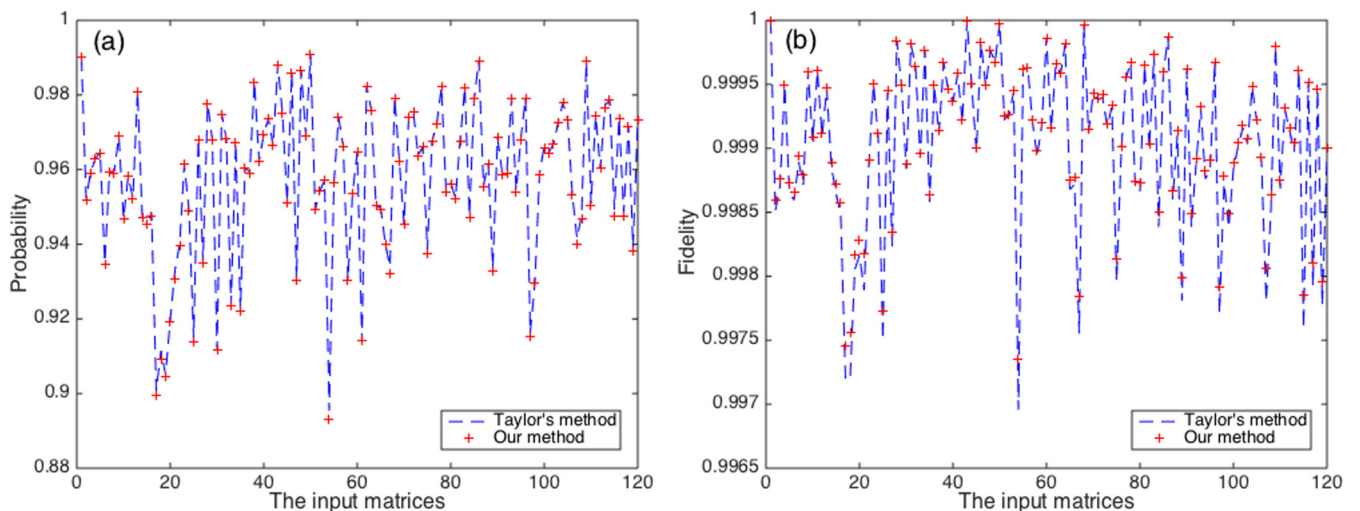

FIG. 10. Probability and fidelity based on the solutions obtained by Taylor's method and our method. (a) Probability. (b) Fidelity.
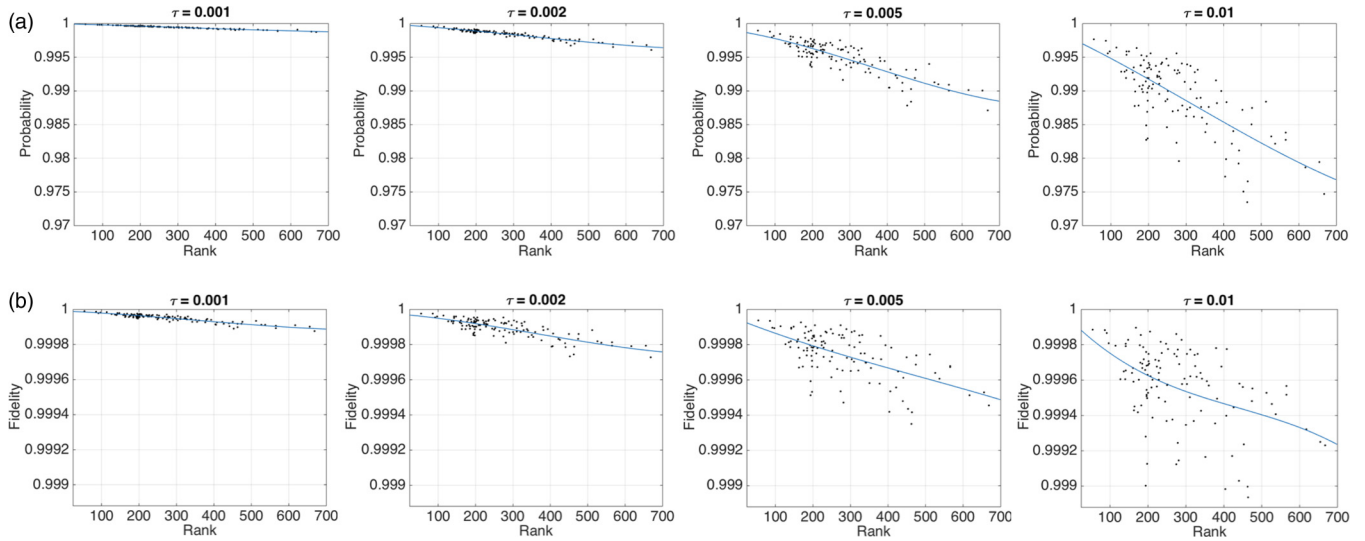
FIG. 11. Probability and fidelity with the rank of input matrices. (a) Probability. (b) Fidelity.

fidelity may not be significantly affected by the rank of the matrix.

Figure 12 shows how the probability and the fidelity change with the hyperparameter τ. Each label τ in the x axis consists
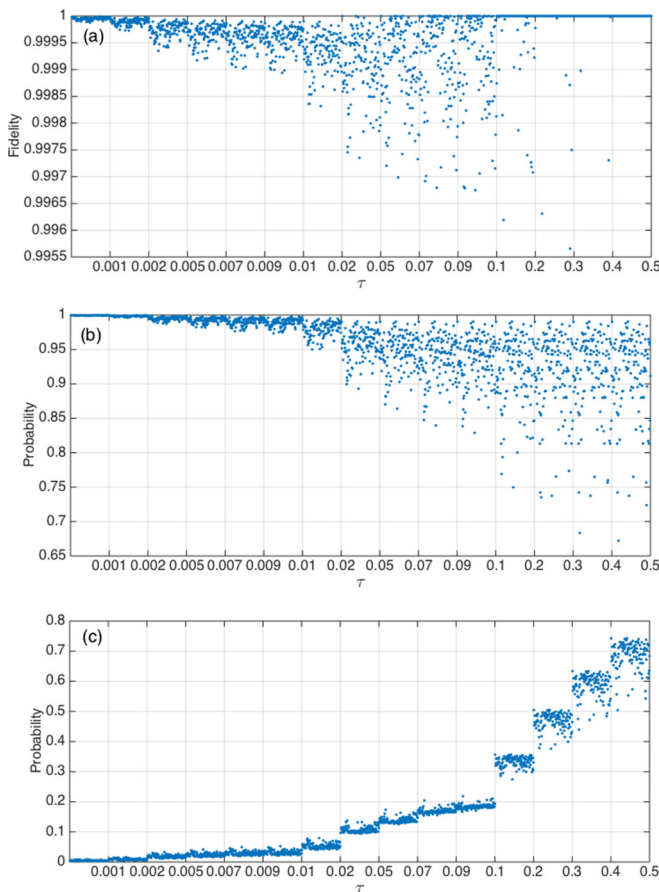

FIG. 12. (a) Fidelity with different input matrices and different τ. (b) Probability with different input matrices and different τ. (c) The difference of the probabilities in terms of our method and the normal method with different input matrices and different τ.

of 120 inputs (i.e., the dimensional of the x axis amounts to 15×120). The blue dots in Figs. 12(a) and 12(b) represent the fidelity and the probability derived by Eqs. (21) and (20), respectively. The blue dots in Fig. 12(c) represent the difference between the probabilities derived by our method and the method in Ref. [33], i.e., $\Delta P = P(\alpha) - P_1$.

As shown in Fig. 12(a), the fidelity is higher than 99.95% when $\tau \leqslant 0.002$. Then the fidelity declines and falls to its lowest point, at which point it begins to increase, reaching its highest point at 1 [when $\sigma_2 \leqslant \tau < \sigma_1$, the fidelity is $F(\alpha) = \sigma_1^2 y_1 / \sqrt{\sigma_1^2 y_1^2 \times \sigma_1^2} = 1$ according to Eq. (21)]. The lowest point among the 120 different matrices is just over 99.55%, which shows that our method can derive high fidelity even at the worst case.

As shown in Fig. 12(b), the probability decreases steadily while τ rises. This trend can be proved by Theorem 1 below. When $\tau \leqslant 0.02$, the probability can be reached to a point over 95%, and the lowest probability is over 65% among the 120 different matrices when τ is large. Comparing to Fig. 12(c), the difference between the probabilities derived by our method and the method in Ref. [33] increases steadily while τ rises. When $\tau \leqslant 0.02$, the difference is less than 10%. At the point $\tau = 0.1$, the difference begins to increase rapidly. This indicates that when τ is large, our methods can derive much higher probability than the normal method.

The probability of obtaining the final result would affect the iteration of the quantum circuit. Theorem 1 gives a lower bound of the probability.

Theorem 1. For a given input matrix $\mathbf{A}_0$, the probability of obtaining the final result is a nonincreasing function of the hyperparameter τ, and the lower bound of the probability is

$$P_{\min} = \frac{\sigma_1}{\sum_{k=1}^{r} \sigma_k^2}. \tag{C1}$$

Apparently, the iteration of the quantum circuit is $O(\sum_{k=1}^{r} \sigma_k^2 / \sigma_1)$.

Proof. According to Eq. (20) and the value of $\tilde{\alpha} = \frac{\pi}{2(1-\tau/\sigma_1)}$, the probability becomes

$$
\begin{aligned}
P &= \frac{\sum_{k=1}^{r} \sigma_k^2 \sin^2(y_k \alpha)}{\sum_{k=1}^{r} \sigma_k^2} \\
&= \frac{\sum_{k=1}^{r} \sigma_k^2 \sin^2\left[\frac{\pi \sigma_1}{2\sigma_k} \frac{(\sigma_k - \tau)_+}{\sigma_1 - \tau}\right]}{\sum_{k=1}^{r} \sigma_k^2} \triangleq P(\tau).
\end{aligned} \tag{C2}
$$

Obviously, as $\sigma_k \leqslant \sigma_1$ and $\tau > 0$, the function $f(\tau) = \frac{\pi \sigma_1}{2\sigma_k} \frac{(\sigma_k - \tau)_+}{\sigma_1 - \tau}$ is a nonincreasing function of the hyperparameter τ, and as $y_1 \alpha \in [0, \pi/2]$, which has been assumed in Sec. III D, the function $g(\tau) = \sin^2(f(\tau))$ is an increasing function of $f(\tau)$. Therefore, $P(\tau)$ is a nonincreasing function of τ. Specifically, when τ satisfies $\sigma_2 < \tau < \sigma_1$, $P(\tau)$ gets the lower bound shown as in Eq. (C1).

---

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, 2010), pp. 1–59.

[2] G. Sentís, J. Calsamiglia, R. M. Tapia, and E. Bagan, Sci. Rep. **2**, 708 (2012).

[3] M. Schuld, I. Sinayskiy, and F. Petruccione, Contemp. Phys. **56**, 172 (2014).

[4] P. Wittek, *Quantum Machine Learning what Quantum Computing Means to Data Mining* (Elsevier, Amsterdam, 2014).

[5] P. Shor, in *Symposium on Foundations of Computer Science* (IEEE, Piscataway, NJ, 1994), pp. 124–134.

[6] L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997).

[7] C. Durr and P. Hoyer, arXiv:quant-ph/9607014v2.

[8] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, *Quantum Amplitude Amplification and Estimation*, Contemporary Mathematics Series, Millenium Vol. 305 (AMS, New York, 2002).

[9] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **100**, 160501 (2008).

[10] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[11] Y. Liu, J. Yuan, B. Duan, and D. Li, arXiv:1709.05015v3.

[12] D. Li, M. McGettrick, F. Gao, J. Xu, and Q. Y. Wen, Phys. Rev. A **93**, 042323 (2016).

[13] S. Lloyd, M. Mohseni, and P. Rebentrost, Nat. Phys. **10**, 631 (2014).

[14] P. Rebentrost, A. Steffens, and S. Lloyd, Phys. Rev. A **97**, 012327 (2018).

[15] P. Rebentrost, M. Mohseni, and S. Lloyd, Phys. Rev. Lett. **113**, 130503 (2014).

[16] Y. Liu and S. Zhang, Theor. Comput. Sci. **657**, 38 (2017).

[17] M. Schuld, I. Sinayskiy, and F. Petruccione, Phys. Rev. A **94**, 022342 (2016).

[18] J. Adcock, E. Allen, M. Day, S. Frick, J. Hinchliff, M. Johnson, S. Morleyshort, S. Pallister, A. Price, and S. Stanisic, arXiv:1512.02900v1.

[19] B. Qi, Z. Hou, L. Li, D. Dong, G. Xiang, and G. Guo, Sci. Rep. **3**, 3496 (2013).

[20] C.-H. Yu, F. Gao, Q.-L. Wang, and Q.-Y. Wen, Phys. Rev. A **94**, 042311 (2016).

[21] Y. Cao, A. Daskin, S. Frankel, and S. Kais, Mol. Phys. **110**, 1675 (2012).

[22] J. Pan, Y. Cao, X. Yao, Z. Li, C. Ju, H. Chen, X. Peng, S. Kais, and J. Du, Phys. Rev. A **89**, 022313 (2014).

[23] X. D. Cai, C. Weedbrook, Z. E. Su, M. C. Chen, M. Gu, M. J. Zhu, L. Li, N. L. Liu, C. Y. Lu, and J. W. Pan, Phys. Rev. Lett. **110**, 230501 (2013).

[24] S. Barz, I. Kassal, M. Ringbauer, Y. O. Lipp, B. Dakić, A. Aspuru-Guzik, and P. Walther, Sci. Rep. **4**, 6115 (2014).

[25] B. Duan, J. Yuan, Y. Liu, and D. Li, Phys. Rev. A **96**, 032301 (2017).

[26] J. F. Cai, E. J. Candes, and Z. Shen, SIAM J. Optim. **20**, 1956 (2008).

[27] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais, New J. Phys. **15**, 013021 (2012).

[28] V. V. Vedral, A. Barenco, and A. Ekert, Phys. Rev. A **54**, 147 (1996).

[29] A. Parent, M. Roetteler, and M. Mosca, arXiv:1706.03419v1.

[30] M. K. Bhaskar, S. Hadfield, A. Papageorgiou, and I. Petras, Quantum Comput. & Inf. **16**, 197 (2016).

[31] C. S. Evelyn, *Fast Inverse Square Root* (Fec Publishing, 2012).

[32] IEEE Std. 754-2008, 1 (2008).

[33] I. Cong and L. Duan, New J. Phys. **18**, 073011 (2016).