Quantum Algorithms for Tensor Decompositions and Machine Learning

Wang Xiaoqiang

PhD

The Hong Kong Polytechnic University

2021

The Hong Kong Polytechnic University

Department of Applied Mathematics

Quantum Algorithms for Tensor Decompositions and Machine Learning

Wang Xiaoqiang

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

May 2021

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

WANG Xiaoqiang (Name of student)

Abstract

Tensor decompositions have been proved to be useful in a number of applications, such as data completion, recommendation systems, multi-partite quantum systems, etc. Some of these applications exploit low structural complexity of the data, expressed either as low rank for the matrices, or low tensor-rank under some tensor decompositions, such as higher-order singular value decomposition (HOSVD), tensor singular value decomposition (t-svd), etc. In this thesis, we focus on exploring several novel applications of the tensor t-svd decomposition.

The t-svd decomposition extends the familiar matrix svd strategy to tensors and perform matrix svd in the Fourier domain. However, the complexity of calculating full t-svd is extremely high especially for large scale datasets. Hence in our work, we present the first quantum t-svd algorithm for third-order tensors which achieves polynomial speedup compared with its classical counterpart, and then extend it to order p tensors. To our best knowledge, the efficiency of this algorithm beats any known classical t-svd algorithms in the literature.

Quantum machine learning investigates how quantum techniques can be used to speed up some classical machine learning problems. Based on the proposed quantum t-svd algorithm, we next extend Kerenidis and Prakash's matrix recommendation system algorithm to third-order tensors, and propose a quantum machine learning algorithm, context-aware recommendation systems algorithm, based on truncated t-svd factorization. In fact, our algorithm offers recommendations by just sampling

from an approximated preference matrix, which corresponds to measuring certain times a quantum state representing a user' approximated preference information, instead of reconstructing the entire tensor as its classical counterpart does. Therefore, the running time is polylogrithmic in the dimension of the preference tensor.

Inspired by the Monte-Carlo randomized algorithm for finding the low-rank matrix approximations proposed by Frieze, Kannan and Vempala, we present a classical Monte-Carlo low tubal-rank tensor approximation algorithm based on truncated t-svd. The main idea is approximating the original tensor by performing truncated matrix svd on every frontal slice of the small sampled tensor under Fourier domain. In terms of time complexity, our algorithm achieves a polynomial speedup compared with the classical truncated t-svd algorithm.

In the final part of our work, two schemes for the effective generation of large-size Schrödinger's cat states are proposed based on conditioned measurement, which provides a powerful resource for quantum information technology based on the superposition of coherent states. The schemes are based on the linear operation of Fock states and squeezed vacuum states. The simulation results shows that odd cat states with an amplitude of 2.001 with the fidelity of 0.99 could be obtained.

Acknowledgements

The privilege of the first position is for my supervisor, Dr. Guofeng Zhang. He is a serous scholar, a genuinely learned man, a man of strong principles. During these three years, he has taught me how to do research by his words and deeds. What impressed me most is his wise, rigorous thinking habits and diligent, positive working attitude. The scientific guidance he gave to me at each time of our meeting is a lifetime treasure that I will bring with me wherever I go. Then, I would like to thank my supervisor Dr. Heung Wing Joseph Lee for enrolling me as his student and he has been a constant support during my professional development.

Next, I would like to thank my supervisor during the master period, Prof. Qingling Zhang, and his wife, Dr. Yi Zhang, for their more than 10 years' care and guidance in my study, career and life. They care about my life in Hong Kong and give me many farsighted suggestions in the crucial turning pointing of my life. Also, I would like to give my gratitude to Prof. Youfa Han, Prof. Liqun Qi and Prof. Yanping Lin for helping me during my research period.

Furthermore, I will give my great thanks to my senior and research partner, Dr. Lejia Gu, for his support and help in both study and life. When I first entering this field, I was inadequate to accomplish the task I was facing. He helped me to explain the exercise sets designed by Dr. Guofeng Zhang in order to improve my understanding of Quantum Computing. He is always the first one to listen and criticize my preliminary idea, but also the first one to untangle my confusions and

the intricate problems. Next, I would like to thank my senior Dr. Zhiyuan Dong for his academic advice and kindly assistance. Dr. Guofeng Zhang, Dr. Zhiyuan Dong and Dr. Lejia Gu have been the greatest example of team spirit. Also, I want to thank my friends, Ms. Changyu Liu, Dr. Huihui Qin, Ms. Jie Deng, Ms. Ying Sun, Mr. Xiaozhou Wang, and Mr. Qinyi Zhang, for their academic assistant and help in my life. We spent a lot of happy time together during these three years.

Finally, I would like to thank my parants, Mr. Fude Wang and Mrs. Xiuqin Shen, and my mother-in-law, Mrs. Yulan Zheng, for their great and selfless love. I also want to express my deep gratitude to my husband, Prof. Yan Shi, who give me a steady flow of support and help in the achievement of this thesis. This one is dedicated to my daughter, Ms. Yahan Shi, a fairy with shining hair, a princess with sparking eyes.

Contents

List of Figures

List of Tables

List of Algorithms

List of Notations

$\mathbb{R}, \mathbb{R}^+$	the set of real numbers, the set of positive real numbers
$\mathbb{C}$	the set of complex numbers
$a, b, \ldots$	scalar
$\boldsymbol{x}, \boldsymbol{y}, \ldots$	vector
$A, B, \ldots$	matrix or operator
$\mathcal{A}, \mathcal{B}, \ldots$	tensor
$\lvert\cdot\rangle$	ket, dirac notation of a column vector in quantum mechanics
$\langle\cdot\rvert$	bra, dirac notation of a row vector in quantum mechanics
$\mathcal{O}(f(n))$	big O notation, which indicates a running time with upper bound $cf(n)$ for a fixed $c \in \mathbb{R}^+$ and sufficiently large positive n
$\cdot$ or $\langle\cdot,\cdot\rangle$	inner product
$\odot$	Hadamard product
$\circledast$	circular convolution
$\otimes$	Kronecker product
$\mathcal{A} * \mathcal{B}$	t-product between tensors $\mathcal{A}$ and $\mathcal{B}$
$A^{(i)}$	the abbreviation for $\mathcal{A}(:,:,i)$, the i-th frontal slice of tensor $\mathcal{A}$

DFT($\boldsymbol{u}$)	the discrete Fourier transform (DFT) on $\boldsymbol{u}$, which is computed by the fast Fourier transform represented in Matlab notation fft($\boldsymbol{u}$)
ifft($\boldsymbol{u}$)	inverse fast Fourier transform
$\hat{\mathcal{A}}$	DFT along the third mode of $\mathcal{A}$, i.e., $\hat{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3)$

Chapter 1

Introduction

1.1 Background

Machine learning is a branch of artificial intelligence and is increasingly ubiquitous in various areas such as natural language processing, data mining, biological analysis, etc. However, a major deficiency of many machine learning algorithms is their high computational and storage cost when processing big data. On the other hand, quantum computer is considered as one of the most promising and emerging technologies, and its development has made great progress in recent years. Considering high demanding computational power of machine learning and the fast development of quantum technology, researchers are developing a new interdisciplinary research field, quantum machine learning.

Quantum machine learning explores the interaction between quantum computing and machine learning, by investigating how quantum techniques, e.g., superposition and entanglement, can be used to speed up some classical machine learning problems. Successful examples are quantum support vector machine (QSVM) [63], quantum principle component analysis (QPCA) [38], among others. In most cases, quantum computing is supposed to deal with quantum data, as commented in [3]. Hence, the classical data should be preprocessed into quantum data using some methods like QRAM [20, 33] so that quantum algorithms can proceed as desired. As quantum

features such as parallelism and entanglement can be used to accelerate some computational procedures which classical operations are generally regarded as inefficient, it is reasonable to assume that the performance of quantum computers outperforms classical computers on certain machine learning problems.

The development of quantum computer has made some progress recently. Some companies have produced actual prototype machines of universal quantum computers based on the quantum circuit model, which make experiments can be conducted with quantum computational operations on a small number of qubits via cloud platforms. However, the general large-scale quantum computer is still being developed.

Tensor refers to a multi-dimensional array of numbers, thus it can represent more complex structures of higher-order data. Applications involving tensors include image deblurring, video recovery, denoising, data completion, multi-partite quantum systems, networks and machine learning [34, 92, 93, 14, 95, 35, 58, 57, 67, 66, 25, 89, 59, 60, 64, 91, 87, 26, 43], due to the flexibility of tensors in representing data. Some of these applications make use of various tensor decompositions including CANDECOMP/PARAFAC (CP) [9], TUCKER [84], higher-order singular value decomposition (HOSVD) [11, 89, 21], tensor-train decomposition (TT) [53], and tensor singular value decomposition (t-svd) [34, 93, 43].

Plenty of research has been carried out on t-svd in the last decade. The concept of t-svd was first proposed by Kilmer and Martin [34] for third-order tensors. Later, Martin et al. [45] extended it to higher-order tensors. The t-svd algorithm is superior to TUCKER and CP decompositions in the sense that it extends the familiar matrix svd strategy to tensors, thus avoiding the loss of information inherent in flattening tensors used in TUCKER and CP decompositions. Compared with HOSVD, t-svd also has optimality properties similar to the truncated svd for matrices, hence t-svd is shown to have better performance than HOSVD when applied to facial recognition [22], tensor completion [92, 75]. Another advantage of t-svd is that it can be obtained

by computing matrix svd in the Fourier domain; the similar idea allows other matrix factorization techniques like QR decomposition to be extended to tensors easily.

However, the complexity of calculating full t-svd for third-order N dimensional tensors is $\mathcal{O}(N^4)$, which is extremely high for large scale datasets. Hence many works have devoted to low-rank approximated t-svd representation which gives up the optimality property and has comparatively low comlexity. In [90], Zhang et al. propose a randomized t-svd method which can produce a factorization with similar66 properties to the t-svd, and the computational complexity is reduced to $\mathcal{O}(kN^3 + N^3 \log N)$, where k is the truncated term.

In this dissertation, we present the first quantum t-svd algorithm for third-order tensors which achieves polynomial speedup compared with its classical counterpart, and present quantum algorithms for low rank approximation based on this quantum t-svd algorithm. We next present a quantum algorithm for solving a machine learning problem, namely, the context-aware recommendation systems based on that quantum tensor approximation problem. Also, we present a classical low rank tensor approximation based on t-svd and Monte-Carlo methods. Finally, we propose two schemes for the effective generation of large-size Schrödinger's cat states.

1.2 Organization of the Thesis

The rest of this thesis is organized as follows.

In Chapter 2, we introduce some quantum computing preliminaries, together with some relevant definitions of tensors. The classical (non-quantum) t-svd algorithm is the theoretical basis of our quantum algorithms, so we will also introduce its relevant results in this chapter.

In Chapter 3, we present a quantum t-svd algorithm for third-order tensors and then extend it to order-p tensors. We prove that our quantum t-svd algorithm for a

third-order N dimensional tensor runs in time $\mathcal{O}\left(N\text{polylog}(N)\right)$ which is polynomial faster than its classical counterpart.

In Chapter 4, we present a quantum algorithm for recommendation systems which incorporates the contextual information of users to the personalized recommendation. The preference information of users is encoded in a third-order tensor of dimension N which can be approximated by the truncated tensor singular value decomposition (t-svd) of the subsample tensor. Unlike the classical algorithm that reconstructs the approximated preference tensor using truncated t-svd, our quantum algorithm obtains the recommended product under certain context by measuring the output quantum state corresponding to an approximation of a user's dynamic preferences. The algorithm achieves the time complexity $\mathcal{O}(\sqrt{k}N\text{polylog}(N))$, compared to the classical counterpart with complexity $\mathcal{O}(kN^3)$, where k is the truncated tubal-rank.

In Chapter 5, we present a Monte-Carlo low tubal-rank tensor approximation algorithm based on t-svd. Our algorithm is inspired by the Monte-Carlo randomized algorithm for finding the low rank matrix approximations proposed by Frieze, Kannan and Vempala [17]. We first sample the slices of the original tensor by length-squared sampling and then approximate the original tensor in Fourier domain using truncated t-svd method. The complexity of our algorithm is only $\mathcal{O}(k^5N/\epsilon^4 + N^3\text{log}N)$ for finding a description of the approximation tensor. On the other hand, if we need to calculate the approximated tensor definitely, the complexity of our algorithm is $\mathcal{O}(k^5N/\epsilon^4 + kN^3 + N^3\text{log}N)$, which is advantageous in the case of $k << N$ when compared with the classical t-svd whose complexity is $\mathcal{O}(N^4 + N^3\text{log}N)$.

In Chapter 6, we propose two schemes for the effective generation of large-size Schrödinger's cat states based on conditioned measurement, which provides a powerful resource for quantum information technology based on the superposition of coherent states. The schemes are based on the linear operation of Fock states and squeezed vacuum states, and the simulation results shows that odd cat states with

an amplitude of 2.001 with the fidelity of 0.99 could be obtained.

Chapter 2

Preliminaries

In Section 2.1, we introduce some basic concepts in quantum computing including quantum Fourier transform (QFT) and the quantum singular value estimation algorithm (QSVE) [33]. In Section 2.2, we introduce some concepts of tensor. Also, we review the concept of t-product and relevant definitions used in the classical t-svd algorithm in Section 2.2.1.

2.1 Quantum Computing

Quantum bit (qubit) is a fundamental concept of quantum computation, just as the classical computation and information is built on bits.

Definition 2.1. ***qubit***

Two possible states for a qubit are the states $|0\rangle$ *and* $|1\rangle$*. A qubit can be in a state which is the linear combinations of* $|0\rangle$ *and* $|1\rangle$*:*

$$|\phi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,$$

where $\alpha, \beta \in \mathbb{C}$ *and* $|\alpha|^2 + |\beta|^2 = 1$*.*

In other words, the state of a qubit can be regarded as a vector in a two-dimensional complex vector space formed with an orthonormal basis

$$|0\rangle \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.1}$$

We shall denote by $\langle\phi|$ the conjugate-transpose of $|\phi\rangle$. The joint state of two qubits $|\phi\rangle, |\psi\rangle$ is given by $|\phi\rangle\otimes|\psi\rangle$. Accordingly, an n -qubit quantum state can be denoted as

$$|\phi\rangle = \sum_{i_1,\ldots,i_n} \alpha_{i_1\ldots i_n} |i_1 \ldots i_n\rangle$$

where $i_k \in \{0,1\}, \sum_{i_1\ldots i_n} |\alpha_{i_1\ldots i_n}|^2 = 1$ and $|i_1 \ldots i_n\rangle \equiv |i_1\rangle \otimes \cdots \otimes |i_n\rangle$. The basis $\{|i_1 \ldots i_n\rangle \mid i_k \in \{0,1\}\}$ is called the computational basis in a 2^n-dimensional complex vector space.

Unitary operators and measurements are two operations which can perform on a quantum state.

Definition 2.2. *The unitary operator U on n qubits is a matrix of dimension $2^n \times 2^n$ satisfying $UU^\dagger = U^\dagger U = I$.*

Definition 2.3. *Quantum measurements are described by a collection $\{M_m\}$ of measurement operators. The operators satisfy the completeness condition: $\sum_m M_m^\dagger M_m = \mathcal{I}$. If the state of the quantum system is $|\psi\rangle$ before the measurement, then the probability that result m occurs is*

$$p(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle,$$

and the state of the system immediately after measurement is

$$\frac{M_m |\psi\rangle}{\sqrt{p(m)}}.$$

2.1.1 Quantum Fourier transform

The quantum Fourier transform (QFT) under an orthonormal basis $|x\rangle \in \{|0\rangle, \ldots, |N-1\rangle\}$ is defined as the linear operator with the following action on the basis vectors:

$$\text{QFT} : |x\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{x\cdot k} |k\rangle,$$

where $\omega = e^{\frac{2\pi \mathrm{i}}{N}}$. The inverse QFT is then defined as

$$\mathrm{QFT}^{\dagger} : |k\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \omega^{-k \cdot x} |x\rangle.$$

The circuit of QFT, shown in FIG. 2.1, is composed of a total number of $\mathcal{O}\left(\lceil \log N \rceil^2\right)$ H gates, CNOT gates, and controlled phase gate R_m (see [52, Section 5.1]), where

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad R_m = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi \mathrm{i}}{2^m}} \end{pmatrix}$$

with $m = 2, \cdots, n$, $n = \lceil \log N \rceil$.

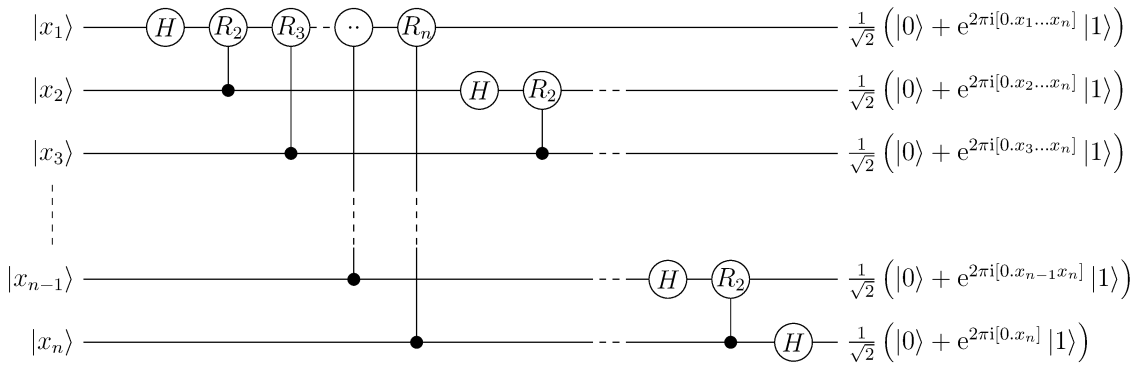


Figure 2.1: The circuit of QFT.

2.1.2 Quantum singular value estimation

In [33], Kerenidis and Prakash propose a quantum singular value estimation (QSVE) algorithm. They assume that the input data is stored in a classical binary tree data structure, as stated in the following lemma, such that the QSVE algorithm with access to this data structure can efficiently create superpositions of rows of the subsample matrix.

Lemma 2.1. *[33, Theorem 5.1] Consider a matrix $A \in \mathbb{R}^{N_1 \times N_2}$ with τ nonzero entries. Let A_i be its i-th row, and $\boldsymbol{s}_A = \frac{1}{||A||_F} \left[||A_0||_2, ||A_1||_2, \cdots, ||A_{N_1-1}||_2\right]^T$. There*

exists a data structure storing the matrix A in $\mathcal{O}(\tau \log^2(N_1 N_2))$ space such that a quantum algorithm having access to this data structure can perform the mapping $U_P : |i\rangle|0\rangle \rightarrow |i\rangle|A_i\rangle$, for $i = 0, \cdots, N_1 - 1$ and $U_Q : |0\rangle|j\rangle \rightarrow |\boldsymbol{s}_A\rangle|j\rangle$, for $j = 0, \cdots, N_2 - 1$ in time $\text{polylog}(N_1 N_2)$.

The following lemma summarizes the main idea of the QSVE algorithm and its detailed description can be found in [33].

Lemma 2.2. *[33, Theorem 5.2] Let $A \in \mathbb{R}^{N_1 \times N_2}$ and $\boldsymbol{x} \in \mathbb{R}^{N_2}$ be stored in the data structure as mentioned in Lemma 2.1. Let the singular value decomposition of A be $A = \sum_{l=0}^{r-1} \sigma_l |u_l\rangle\langle v_l|$, where $r = \min(N_1, N_2)$. The input state $|x\rangle$ can be represented in the eigenstates of A, i.e. $|x\rangle = \sum_{l=0}^{N_2-1} \beta_l |v_l\rangle$. Let $\epsilon > 0$ be the precision parameter. Then there is a quantum algorithm, denoted as U_{SVE}, that runs in time $\mathcal{O}(\text{polylog}(N_1 N_2)/\epsilon)$ and achieves*

$$U_{\text{SVE}}\left(|x\rangle|0\rangle\right) = \sum_{l=0}^{N_2-1} \beta_l |v_l\rangle|\overline{\sigma}_l\rangle$$

with probability at least $1 - 1/\text{poly}(N_2)$, where $\overline{\sigma}_l$ is the estimated value of σ_l satisfying $|\overline{\sigma}_l - \sigma_l| \leqslant \epsilon ||A||_F$ for all l.

Remark 2.1. *With regard to the matrix A stated in Lemma 2.2, we can also choose the input state as $|A\rangle = \frac{1}{||A||_F} \sum_{l=0}^{r-1} \sigma_l |u_l\rangle|v_l\rangle$, corresponding to the vectorized form of the normalized matrix $\frac{A}{||A||_F}$ represented in the svd form. This representation of the input state is adopted in Section 3.3. Note that we are able to express the state $|A\rangle$ in the above form even if the singular pairs of A are not known. According to Lemma 2.2, we can obtain $\overline{\sigma}_l$, an estimate of σ_l, stored in the third register superposed with the singular vectors $\{|u_l\rangle, |v_l\rangle\}$ after performing U_{SVE}, i.e., the output state is $\frac{1}{||A||_F} \sum_{l=0}^{r-1} \sigma_l |u_l\rangle|v_l\rangle|\overline{\sigma}_l\rangle$, where $|\overline{\sigma}_l - \sigma_l| \leqslant \epsilon ||A||_F$ for all $l = 0, \cdots, r-1$.*

2.2 Tensor background and notation

A tensor $\mathcal{A} = (a_{i_1 i_2 \cdots i_p}) \in \mathbb{C}^{N_1 \times N_2 \times \cdots \times N_p}$ is a multidimensional array of data, where p is the order and $(N_1, \cdots, N_p)$ is the dimension. The order of a tensor is the number of modes. For instance, $\mathcal{A} \in \mathbb{C}^{N_1 \times N_2 \times N_3}$ is a third-order tensor of complex values with dimension N_i for mode i, $i = 1, 2, 3$, respectively. In this sense, a matrix A can be considered as a second-order tensor, and a vector $\boldsymbol{x}$ is a tensor of order 1. For a third-order tensor, we use terms frontal slice $\mathcal{A}(:, :, i)$, horizontal slice $\mathcal{A}(i, :, :)$ and lateral slice $\mathcal{A}(:, i, :)$ (see FIG.1). By fixing all indices but the last one, the result is a tube of size $1 \times 1 \times N_3$, which is actually a vector. For example, $\mathcal{A}(i, j, :)$ is the (i, j)-th tube of $\mathcal{A}$.

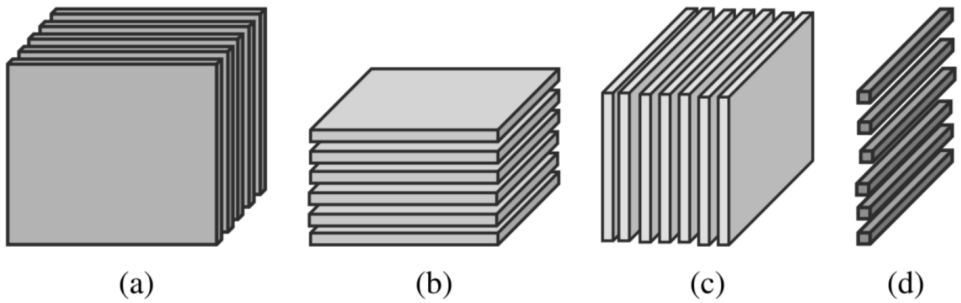


Figure 2.2: (a) frontal slices, (b) horizontal slices, (c) lateral slices of a third-order tensor. (d) a lateral slice as a vector of tubes.

2.2.1 t-product

Definition 2.4. *[34]* ***circulant matrix***

Given a vector $\boldsymbol{u} \in \mathbb{R}^N$ *and a tensor* $\mathcal{B} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ *with frontal slices* $B^{(l)}$, $l = 0, \cdots, N_3 - 1$, *the matrices* $\text{circ}(\boldsymbol{u})$ *and* $\text{circ}(\mathcal{B})$ *are defined as*

$$\text{circ}(\boldsymbol{u}) \triangleq \begin{bmatrix} u_0 & u_{N-1} & \cdots & u_1 \\ u_1 & u_0 & \cdots & u_2 \\ \vdots & \vdots & \ddots & \vdots \\ u_{N-1} & u_{N-2} & \cdots & u_0 \end{bmatrix}, \text{circ}(\mathcal{B}) \triangleq \begin{bmatrix} B^{(0)} & B^{(N_3-1)} & \cdots & B^{(1)} \\ B^{(1)} & B^{(0)} & \cdots & B^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ B^{(N_3-1)} & B^{(N_3-2)} & \cdots & B^{(0)} \end{bmatrix}.$$

Definition 2.5. ***circular convolution***

Let $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^N$. The circular convolution between $\boldsymbol{u}$ and $\boldsymbol{v}$ produces a vector $\boldsymbol{x}$ of the same size, defined as

$$\boldsymbol{x} \equiv \boldsymbol{u} \circledast \boldsymbol{v} \triangleq \text{circ}(\boldsymbol{u})\boldsymbol{v}.$$

As a circulant matrix can be diagonalized by means of the discrete Fourier transform (DFT), from Definition 2.5, we have $\text{DFT}(\boldsymbol{x}) = \text{diag}(\text{DFT}(\boldsymbol{u}))\text{DFT}(\boldsymbol{v})$, where $\text{diag}(\boldsymbol{u})$ returns a square diagonal matrix with elements of the vector $\boldsymbol{u}$ on the main diagonal. As a result, the circular convolution between two vectors in Definition 2.5 is better understood in the Fourier domain, which is given by the following result.

Theorem 2.1. ***Cyclic Convolution Theorem*** *[81]*

Given $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^N$, let $\boldsymbol{x} = \boldsymbol{u} \circledast \boldsymbol{v}$ as defined in Definition 2.5. We have

$$\text{DFT}(\boldsymbol{x}) = \text{DFT}(\boldsymbol{u}) \odot \text{DFT}(\boldsymbol{v}), \tag{2.2}$$

where $\odot$ denotes the element-wise product.

If a tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is considered as an $N_1 \times N_2$ matrix whose (i, j)-th entry is a tube of dimension N_3, i.e., $\mathcal{A}(i, j, :)$, then based on Definition 2.5, the t-product between tensors is defined as follows.

Definition 2.6. ***t-product*** *[34]*

Let $\mathcal{M} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{N} \in \mathbb{R}^{N_2 \times N_4 \times N_3}$. The t-product of $\mathcal{M}$ and $\mathcal{N}$, i.e. $\mathcal{A} \triangleq \mathcal{M} * \mathcal{N}$, is an $N_1 \times N_4 \times N_3$ tensor (i, j)-th tube is

$$\mathcal{A}(i, j, :) = \sum_{k=0}^{N_2-1} \mathcal{M}(i, k, :) \circledast \mathcal{N}(k, j, :) \tag{2.3}$$

for all $i = 1, \ldots, N_1$ and $j = 1, \ldots, N_2$.

Similar as the circular convolution in Definition 2.5, the t-product in Definition 2.6 can be better interpreted in the Fourier domain. Specifically, let $\hat{\mathcal{A}}$ be the tensor whose (i,j)-th tube is $\mathrm{DFT}(\mathcal{A}(i,j,:))$. Then by Theorem 2.1 and Definition 2.6, we have

$$\hat{\mathcal{A}}(i,j,:) = \sum_{k=0}^{N_2-1} \hat{\mathcal{M}}(i,k,:) \odot \hat{\mathcal{N}}(k,j,:), \tag{2.4}$$

which is the Fourier counterpart of Equation (2.3). Interestingly, for a fixed index l in the third mode, Equation (2.4) is equivalent to $\hat{A}^{(l)} = \hat{M}^{(l)}\hat{N}^{(l)}$, which is the conventional matrix product. This nice equivalence relations between t-product and matrix multiplication (in the Fourier domain) are summarized in the following theorem.

Theorem 2.2. *[34] For tensors $\mathcal{M} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{N} \in \mathbb{R}^{N_2 \times N_4 \times N_3}$, the equivalence relation*

$$\mathcal{A} = \mathcal{M} * \mathcal{N} \quad \Longleftrightarrow \quad \hat{A}^{(l)} = \hat{M}^{(l)}\hat{N}^{(l)} \tag{2.5}$$

holds for $l = 0, \cdots, N_3 - 1$, where $\hat{A}^{(l)}$ is the l-th frontal slice of $\hat{\mathcal{A}}$.

In what follows, we list some definitions used in the classical t-svd algorithm.

Definition 2.7. ***tensor transpose*** *[34]*

The transpose of a tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, denoted by $\mathcal{A}^T$, is obtained by transposing all the frontal slices and then reversing the order of the transposed frontal slices 1 through $N_3 - 1$.

The tensor transpose defined in Definition 2.7 has the same property as the matrix transpose, i.e., $(\mathcal{A} * \mathcal{B})^T = \mathcal{B}^T * \mathcal{A}^T$.

Definition 2.8. ***orthogonal tensor*** *[34]*

A tensor $\mathcal{U} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is an orthogonal tensor if it satisfies $\mathcal{U}^T * \mathcal{U} = \mathcal{U} * \mathcal{U}^T = \mathcal{I}$, where $\mathcal{I} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ is an identity tensor, in other words, its first frontal slice $I^{(0)}$ is an $N_1 \times N_1$ identity matrix and all the other frontal slices are zero matrices.

Finally, we given the definition of tensor Frobenius norm as it is quite useful for tensor approximation.

Definition 2.9. ***tensor Frobenius norm*** *[34]*

The Frobenius norm of a third-order tensor $\mathcal{A} = (a_{ijk})$ *is defined as* $||\mathcal{A}||_F = \sqrt{\sum_{i,j,k} |a_{ijk}|^2}$.

Definition 2.10. ***identity tensor*** *[34]*

The identity tensor $\mathcal{I} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ *is a tensor whose first frontal slice* $I^{(0)}$ *is an* $N_1 \times N_1$ *identity matrix and all the other frontal slices are zero matrices.*

Similar to orthogonal matrices, the orthogonality defined in Definition 2.8 preserves the Frobenius norm of a tensor, i.e., given an orthogonal tensor $\mathcal{Q}$, we have $||\mathcal{Q} * \mathcal{A}||_F = ||\mathcal{A}||_F$. Moreover, when the tensor is second-order, Definition 2.8 coincides with the definition of orthogonal matrices. Finally, note that the frontal slices of an orthogonal tensor are not necessarily orthogonal.

Definition 2.11. ***multi-rank*** *[34]*

The multi-rank of a tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ *is a vector in* $\mathbb{R}^{N_3}$ *whose* i*-th entry equals to the rank of* $\hat{\mathcal{A}}(:,:,i)$.

Definition 2.12. ***tubal-rank*** *[34]*

The tubal-rank of a tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ *is the number of nonzero tubes* $\mathcal{S}(i,i,:)$, $i = 0, \cdots, \min(N_1, N_2) - 1$, *in t-svd factorization.*

$$\text{circ}(\boldsymbol{u}) \triangleq \begin{bmatrix} u_0 & u_{N-1} & \cdots & u_1 \\ u_1 & u_0 & \cdots & u_2 \\ \vdots & \vdots & \ddots & \vdots \\ u_{N-1} & u_{N-2} & \cdots & u_0 \end{bmatrix}, \text{circ}(\mathcal{B}) \triangleq \begin{bmatrix} B^{(0)} & B^{(N_3-1)} & \ldots & B^{(1)} \\ B^{(1)} & B^{(0)} & \ldots & B^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ B^{(N_3-1)} & B^{(N_3-2)} & \ldots & B^{(0)} \end{bmatrix}.$$

Chapter 3

Quantum Tensor Singular Value Decomposition

3.1 Introduction

As introduced in Chapter 2, due to the advantage of the classical t-svd algorithm and its high cost, we present a quantum version of t-svd for third-order tensors which reduces the complexity to $\mathcal{O}(N\text{polylog}(N))$. To our best knowledge, the efficiency of this algorithm beats any known classical t-svd algorithms in the literature. In Section 3.4, we extend the quantum t-svd algorithm to order-p tensors.

An important step in a classical t-svd algorithm is to perform discrete Fourier transform (DFT) along the third mode of a tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, obtaining $\hat{\mathcal{A}}$ with computational complexity $\mathcal{O}(N_3 \log N_3)$ for each tube $\mathcal{A}(i, j, :)$, $i = 0, \cdots, N_1 - 1$; $j = 0, \cdots, N_2 - 1$. Thus, the complexity of performing the DFT on all tubes of the tensor $\mathcal{A}$ is $\mathcal{O}(N_1 N_2 N_3 \log N_3)$. In the quantum t-svd algorithm to be proposed, this procedure is accelerated by the quantum Fourier transform (QFT) [52] whose complexity is only $\mathcal{O}((\log N_3)^2)$. Moreover, due to quantum superposition, the QFT can be simultaneously performed on the third register of the state $|\mathcal{A}\rangle$, which is equivalent to performing the DFT for all tubes of $\mathcal{A}$ parallelly, so the total complexity of this step is still $\mathcal{O}((\log N_3)^2)$.

After performing the QFT, in order to further accelerate the second step in the classical t-svd algorithm which performs the matrix svd for every frontal slice of $\hat{\mathcal{A}}$, we apply a modified quantum singular value estimation (QSVE) algorithm originally proposed in [33] to the frontal slices $\hat{\mathcal{A}}(:,:,i)$ parallelly with complexity at most $\mathcal{O}(N\text{polylog}(N))$ for N dimensional tensors. Traditionally, the quantum singular value decomposition of non-sparse low-rank matrices involves exponentiating matrices and outputs the superposition state of singular values and their associated singular vectors. However, this Hamiltonian simulation method requires that the matrix to be exponentiated be low-rank, which is difficult to be satisfied in general. In our algorithm, we use the modified QSVE algorithm, where the matrix is unnecessarily low-rank, sparse or Hermitian.

It should be notice that the original QSVE algorithm proposed in [33] has to be carefully modified to become a useful subroutine in our quantum tensor-svd algorithm. Specifically, the original QSVE, stated in Lemma 2.2, requires the matrix A be stored in the classical binary tree structure, then the singular values of A can be estimated efficiently. Given a tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, in Algorithm 3, QSVE is performed on the matrices $\hat{A}^{(m)}$ which are frontal slices of $\hat{\mathcal{A}}$, $m = 0, \cdots, N_3 - 1$. The difficulty lies in the fact that we cannot require all $\hat{A}^{(m)}$ be stored in the data structure since they are obtained after QFT. It is more reasonable to assume that the frontal slices of the original tensor $\mathcal{A}$ is stored in the binary tree structure. Therefore, the main obstacle we need to overcome is how to estimate the singular values of $\hat{A}^{(m)}$ on the condition that every frontal slice of the original tensor $A^{(k)}$ is stored in the data structure. This problem is solved by Theorem 3.2, and whose proof presents a detailed illustration of this process.

The rest of this chapter is organized as follows. A standard classical t-svd algorithm and several related concepts are introduced in Section 3.2; Section 3.3 presents our main algorithm, quantum t-svd, and its complexity analysis. We extend the

quantum t-svd algorithm to order-p tensors in Section 3.4. In Section 3.5, we conclude the chapter.

3.2 Preliminaries

In this subsection, we present the classical t-svd as well as its pseudocode, Algorithm 1. Simply speaking, the t-svd of a tensor can be interpreted as the usual matrix svd in the Fourier domain, as can be seen in Algorithm 1.

Theorem 3.1. ***tensor singular value decomposition (t-svd)*** *[34]*
For $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, its t-svd is given by $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$, where $\mathcal{U} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ and $\mathcal{V} \in \mathbb{R}^{N_2 \times N_2 \times N_3}$ are orthogonal tensors, and every frontal slice of $\mathcal{S} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is a diagonal matrix, and the entries of $\mathcal{S}$ are called the singular values of $\mathcal{A}$.

The t-svd algorithm [34] utilizing the fast Fourier transform (fft) is presented as followed.

Algorithm 1 t-svd for third-order tensors [34]

Input: $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$
Output: $\mathcal{U} \in \mathbb{R}^{N_1 \times N_1 \times N_3}, \mathcal{S} \in \mathbb{R}^{N_1 \times N_2 \times N_3}, \mathcal{V} \in \mathbb{R}^{N_2 \times N_2 \times N_3}$

$\hat{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3)$;
for $i = 1, \cdots, N_3$ **do**
  $[U, S, V] = \text{svd}(\hat{\mathcal{A}}(:, :, i))$;
  $\hat{\mathcal{U}}(:, :, i) = U; \hat{\mathcal{S}}(:, :, i) = S; \hat{\mathcal{V}}(:, :, i) = V$;
end for
$\mathcal{U} = \text{ifft}(\hat{\mathcal{U}}, [], 3); \mathcal{S} = \text{ifft}(\hat{\mathcal{S}}, [], 3); \mathcal{V} = \text{ifft}(\hat{\mathcal{V}}, [], 3)$;

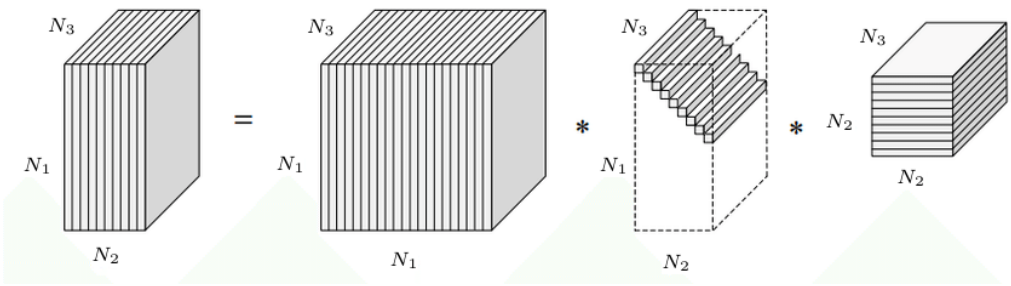


Figure 3.1: The t-svd of $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$.

By this means, we can obtain the tensor decomposition for a tensor $\mathcal{A}$ using t-product by performing matrix factorization strategies on $\hat{A}^{(l)}$. For example, the tensor QR decomposition $\mathcal{A} = \mathcal{Q} * \mathcal{R}$ is performed by the matrix QR decomposition on each frontal slice of the tensor $\hat{\mathcal{A}}$, i.e., $\hat{A}^{(l)} = \hat{Q}^{(l)}\hat{R}^{(l)}$, for $l = 0, \cdots, N_3 - 1$, where $\hat{Q}^{(l)}$ is an orthogonal matrix and $\hat{R}^{(l)}$ is an upper triangular matrix [22]. If we compute the matrix svd on $\hat{A}^{(l)}$, i.e., $\hat{A}^{(l)} = \hat{U}^{(l)}\hat{S}^{(l)}\hat{V}^{(l)\dagger}$, the t-svd of tensor $\mathcal{A}$ is obtained; see Algorithm 1. Before presenting this algorithm, we first introduce some related definitions.

For an order-p tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_p}$, the frontal slices of $\mathcal{A}$ are referenced using linear indexing by reshaping the tensor into an $N_1 \times N_2 \times N_3 N_4 \cdots N_p$ third-order tensor, then the i-th frontal slice is $\mathcal{A}(:, :, i)$ Using this representation, one version of MATLAB pseudocode of the t-svd algorithm for order-p tensors is provided below.

Algorithm 2 t-svd for order-p tensors [45]

Input: $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_p}, \iota = N_3 N_4 \cdots N_p$

for $i = 3, \cdots, p$ **do**
  $\hat{\mathcal{A}} = \text{fft}(\mathcal{A}, [], i)$;
end for
for $i = 1, \cdots, \iota$ **do**
  $[U, S, V] = \text{svd}(\hat{\mathcal{A}}(:, :, i))$;
  $\hat{\mathcal{U}}(:, :, i) = U; \hat{\mathcal{S}}(:, :, i) = S; \hat{\mathcal{V}}(:, :, i) = V$;
end for
for $i = p, \cdots, 3$ **do**
  $\mathcal{U} = \text{ifft}(\hat{\mathcal{U}}, [], i); \mathcal{S} = \text{ifft}(\hat{\mathcal{S}}, [], i); \mathcal{V} = \text{ifft}(\hat{\mathcal{V}}, [], i)$;
end for

In the t-svd literature, the diagonal elements of the tensor $\mathcal{S}$ are called the singular values of $\mathcal{A}$. Moreover, the l_2 norms of the nonzero tubes $\mathcal{S}(i, i, :)$ are in descending order, i.e.,

$$||\mathcal{S}(0, 0, :)||_2 \geqslant ||\mathcal{S}(1, 1, :)||_2 \geqslant \cdots$$
$$\geqslant ||\mathcal{S}(\min(N_1, N_2) - 1, \min(N_1, N_2) - 1, :)||_2.$$

However, it should be noticed that the diagonal elements of $\mathcal{S}$ may be unordered and even negative due to the inverse DFT. Thus, the truncated t-svd method for data approximation or tensor completion is designed by truncating the diagonal elements of $\hat{\mathcal{S}}$ instead of $\mathcal{S}$, as the diagonal elements of the former are non-negative and ordered in descending order; see Lemma 4.1 in Chapter 4.

3.3 Quantum t-svd for third-order tensors

3.3.1 The algorithm

In this section, we present our quantum quasilinear-time t-svd algorithm for third-order tensors, provided that every frontal slice of the tensor is stored in the data structure as introduced in Lemma 2.1 and the tensor as a quantum state can be

efficiently prepared. We first present the algorithm (Algorithm 3) in Section 3.3.1, then we analyze its computational complexity in Section 3.3.2.

Assumption 1. *Every frontal slice of $\mathcal{A}$ is stored in a tree structure introduced in Lemma 2.1.*

Assumption 2. *We can prepare the state*

$$|\mathcal{A}\rangle = \frac{1}{||\mathcal{A}||_F} \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} \sum_{k=0}^{N_3-1} \mathcal{A}(i,j,k) \, |i\rangle \, |j\rangle \, |k\rangle \tag{3.1}$$

efficiently. Without loss of generality, we assume that $||\mathcal{A}||_F = 1$.

Algorithm 3 Quantum t-svd for third-order tensors

Input: tensor $\mathcal{A} = (a_{ijk}) \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ prepared in a quantum state $|\mathcal{A}\rangle$, precision $\epsilon_{\text{SVE}}^{(m)}$, $m = 0, \cdots, N_3 - 1$, $r = \min\{N_1, N_2\}$.

Output: state $|\phi\rangle$.

1: Perform the QFT on the third register of the quantum state $|\mathcal{A}\rangle$ in (3.1) to get

$$|\hat{\mathcal{A}}\rangle = \frac{1}{\sqrt{N_3}} \sum_{m=0}^{N_3-1} \left(\sum_{i,j,k} \omega^{km} a_{ijk} |i\rangle^c |j\rangle^d \right) |m\rangle^e . \tag{3.2}$$

2: Perform the controlled operation

$$U \triangleq \sum_{m=0}^{N_3-1} U_{\text{SVE}}^{(m)} \otimes |m\rangle\langle m| \tag{3.3}$$

on the state $|\hat{\mathcal{A}}\rangle$ to obtain

$$U |\hat{\mathcal{A}}\rangle = \sum_{m=0}^{N_3-1} \sum_{l=0}^{r-1} \hat{\sigma}_l^{(m)} |\hat{u}_l^{(m)}\rangle^c |\hat{v}_l^{(m)}\rangle^d |\overline{\hat{\sigma}}_l^{(m)}\rangle^a |m\rangle^e , \tag{3.4}$$

where $\overline{\hat{\sigma}}_l^{(m)}$ is the estimated value of $\hat{\sigma}_l^{(m)}$, and the singular value decomposition of $\hat{A}^{(m)}$ is $\hat{A}^{(m)} = \sum_{l=0}^{r-1} \hat{\sigma}_l^{(m)} \hat{u}_l^{(m)} \hat{v}_l^{(m)\dagger}$.

3: Perform the inverse QFT on the last register of (3.4) and output the state $|\phi\rangle$ expressed as

$$\frac{1}{\sqrt{N_3}} \sum_{t,m=0}^{N_3-1} \sum_{l=0}^{r-1} \hat{\sigma}_i^{(m)} \omega^{-tm} |\hat{u}_l^{(m)}\rangle^c |\hat{v}_l^{(m)}\rangle^d |\overline{\hat{\sigma}}_l^{(m)}\rangle^a |t\rangle^e , \tag{3.5}$$

where $\omega = e^{2\pi \mathrm{i}/N_3}$.

The quantum circuit of Algorithm 3 is shown in FIG. 3.2, where the blocks $U_{\text{SVE}}^{(m)}$, $m = 0, \cdots, N_3 - 1$, are illustrated in FIG. 3.3.

Before illustrating the algorithm, we first interpret the final quantum state $|\phi\rangle$. Similar to the quantum singular value decomposition for matrices [65] that the output allows singular values and associated singular vectors to be revealed in a quantum form, the output state $|\phi\rangle$ in our algorithm also finds the estimated values of $\hat{\sigma}_l^{(m)}$,

$\overline{\hat{\sigma}}_l^{(m)}$, which are stored in the third register, in superposition with corresponding singular vectors. Although the singular values of the tensor $\mathcal{A}$ are defined as $\sigma_l^{(k)} = \frac{1}{\sqrt{N_3}} \sum_{m=0}^{N_3-1} \omega^{-km} \hat{\sigma}_l^{(m)}$, according to Algorithm 1 for the classical t-svd, the singular values of $\hat{A}^{(m)}$, $\hat{\sigma}_l^{(m)}$, have wider applications than the singular values of $\mathcal{A}$, $\sigma_l^{(k)}$. For example, some low-rank tensor completion problems are solved by minimizing the tensor nuclear norm of the tensor, which is defined as the sum of all the singular values of $\hat{A}^{(m)}$ [93, 95]. Moreover, the theoretical minimal error truncation is also based on the singular values of $\hat{A}^{(m)}$; see Lemma 4.1. Therefore, in Algorithm 3, we estimate the values of $\hat{\sigma}_l^{(m)}$, $m = 0, \cdots, N_3$; $l = 0, \cdots, r-1$, and store them in the third register of the final state $|\phi\rangle$ for future use. Furthermore, in terms of the circulant matrix $\mathrm{circ}(\mathcal{A})$ defined in Definition 2.4, $\frac{1}{\sqrt{N_3}} \sum_{t=0}^{N_3-1} \omega^{-tm} |t\rangle |\hat{v}_l^{(m)}\rangle$ is the right singular vector corresponding to its singular value $\hat{\sigma}_l^{(m)}$. Similarly, the corresponding left singular vector is $\frac{1}{\sqrt{N_3}} \sum_{t=0}^{N_3-1} \omega^{-tm} |t\rangle |\hat{u}_l^{(m)}\rangle$.

Next, we give a detailed explanation on Step 2. We first rewrite the state in (3.2) for further use. For every fixed m, the unnormalized state

$$\frac{1}{\sqrt{N_3}} \sum_{i,j,k} \omega^{km} a_{ijk} |i\rangle |j\rangle \tag{3.6}$$

in (3.2) corresponds to the matrix

$$\hat{A}^{(m)} = \frac{1}{\sqrt{N_3}} \sum_{i,j,k} \omega^{km} a_{ijk} |i\rangle\langle j| , \tag{3.7}$$

namely, the m-th frontal slice of the tensor $\hat{\mathcal{A}}$. Normalizing the state in (3.6) produces a quantum state

$$|\hat{A}^{(m)}\rangle \triangleq \frac{1}{\sqrt{N_3}||\hat{A}^{(m)}||_F} \sum_{i,j,k} \omega^{km} a_{ijk} |i\rangle^c |j\rangle^d . \tag{3.8}$$

Therefore, the state $|\hat{\mathcal{A}}\rangle$ in (3.2) can be rewritten as

$$|\hat{\mathcal{A}}\rangle = \sum_{m=0}^{N_3-1} ||\hat{A}^{(m)}||_F \, |\hat{A}^{(m)}\rangle^{cd} \, |m\rangle^e \, . \tag{3.9}$$

In Step 2, we utilize a controlled operation U defined in (4.6) to estimate the singular values of $\hat{A}^{(m)}$ parallelly, $m = 0, \cdots, N_3-1$. Due to the quantum parallelism, the operator U performed on the superposition state $|\hat{\mathcal{A}}\rangle$ is thus equivalent to $U_{\mathrm{SVE}}^{(m)}$ performed on each of the components $|\hat{A}^{(m)}\rangle$ as a single input. That is,

$$U \, |\hat{\mathcal{A}}\rangle = \sum_{m=0}^{N_3-1} ||\hat{A}^{(m)}||_F \left(U_{\mathrm{SVE}}^{(m)} \, |\hat{A}^{(m)}\rangle^{cd} \right) |m\rangle^e \, . \tag{3.10}$$

Next, we focus on the result of $U_{\mathrm{SVE}}^{(m)} \, |\hat{A}^{(m)}\rangle$ in (3.10). The state $|\hat{A}^{(m)}\rangle$ can be rewritten in the form

$$|\hat{A}^{(m)}\rangle = \sum_{l=0}^{r-1} \frac{\hat{\sigma}_l^{(m)}}{||\hat{A}^{(m)}||_F} \, |\hat{u}_l^{(m)}\rangle^c \, |\hat{v}_l^{(m)}\rangle^d \, , \tag{3.11}$$

where $\frac{\hat{\sigma}_l^{(m)}}{||\hat{A}^{(m)}||_F}$ is the rescaled singular value of $\hat{A}^{(m)}$. The following theorem describes $U_{\mathrm{SVE}}^{(m)}$, a modified quantum singular value estimation process on each matrix $\hat{A}^{(m)}$ utilizing the corresponding input $|\hat{A}^{(m)}\rangle$ represented in (3.11).

Theorem 3.2. *Given every frontal slice of the original tensor $\mathcal{A}$ stored in the data structure (Lemma 2.1), there is a quantum algorithm, denoted as $U_{\mathrm{SVE}}^{(m)}$, that uses the input $|\hat{A}^{(m)}\rangle$ in (3.11) and outputs the state*

$$U_{\mathrm{SVE}}^{(m)} \, |\hat{A}^{(m)}\rangle = \frac{1}{||\hat{A}^{(m)}||_F} \sum_{l=0}^{r-1} \hat{\sigma}_l^{(m)} \, |\hat{u}_l^{(m)}\rangle^c \, |\hat{v}_l^{(m)}\rangle^d \, |\overline{\hat{\sigma}}_l^{(m)}\rangle^a \tag{3.12}$$

with probability at least $1 - 1/\mathrm{poly}(N_2)$, where $(\hat{\sigma}_l^{(m)}, \hat{u}_l^{(m)}, \hat{v}_l^{(m)})$ is the singular triplet of the matrix $\hat{A}^{(m)}$ in (3.7), and $\epsilon_{\mathrm{SVE}}^{(m)}$ is the precision such that $|\overline{\hat{\sigma}}_l^{(m)} - \hat{\sigma}_l^{(m)}| \leqslant$

$\epsilon_{\mathrm{SVE}}^{(m)}||\hat{A}^{(m)}||_F$ *for all* $l = 0, \cdots, r-1$. *For tensor* $\mathcal{A}$ *with same dimension* N *on every order, the running time to implement* $U_{\mathrm{SVE}}^{(m)}$ *is* $\mathcal{O}\left(N\mathrm{polylog}N/\epsilon_{\mathrm{SVE}}^{(m)}\right)$.

Proof. See also the proof of Theorem 3.2.

Actually, the process $U_{\mathrm{SVE}}^{(m)}$ proposed in Theorem 3.2 is quite different from the original QSVE technique introduced in Lemma 2.2. In Theorem 3.2, it is proved that we can estimate the singular values of $\hat{A}^{(m)}$ on the condition that each frontal slice of the original tensor $A^{(k)}$ is stored in the binary tree. The proof of Theorem 3.2 presents a detailed illustration of the procedure of $U_{\mathrm{SVE}}^{(m)}$, and the circuit shown in FIG. 3.3 can help to understand it.

Thus after Step 2, the state in (3.10) becomes the state in (3.4) based on Theorem 3.2.

Our quantum t-svd algorithm can be used as a subroutine of other algorithms, that is, it is suitable for some specific applications where the singular values of $\hat{A}^{(m)}$ are used. For example, some third order tensor completion problems can be efficiently solved by extracting the singular values of $\hat{A}^{(m)}$ and only keeping the greater ones. Moreover, some context-aware recommendation systems also utilize tensor factorizations, such as the truncated t-svd [93] and the truncated HOSVD [77].

Remark 3.1. *Although the classical t-svd is essentially a matrix SVD in the Fourier basis, but there are certain subtleties which makes it difficult to apply QSVE directly. Specifically, given a third-order tensor* $\mathcal{A}$*, in step 1 of the t-svd algorithm, DFT is implemented on the tube* $\mathcal{A}(i, j, :)$ *for each pair* (i, j)*, resulting in a linear combination of elements in this tube. After that, in step 2, matrix SVD is performed on each frontal slice of* $\hat{\mathcal{A}}$*, namely* $\hat{\mathcal{A}}(:, :, k)$ *for each given* k*. Hence, DFT and matrix SVD are performed on different directions.*

Remark 3.2. *Note that the input of* $U_{\mathrm{SVE}}^{(m)}$ *is* $|\hat{A}^{(m)}\rangle$ *instead of an arbitrary quan-*

tum state commonly used in some quantum svd algorithms [65]. This input fits our quantum t-svd algorithm better since it keeps the entire singular information $(\hat{\sigma}_l^{(m)}, \hat{u}_l^{(m)}, \hat{v}_l^{(m)})$ *completely, thus our algorithm can output a quantum state whose representation is similar to the matrix svd. Another consideration is that we do not need the information unrelated to the tensor* $\mathcal{A}$ *(e.g. an arbitrary state) to be involved in our algorithm.*

Remark 3.3. *In terms of the circulant matrix* $\mathrm{circ}(\mathcal{A})$ *defined in Definition 2.4,* $\frac{1}{\sqrt{N_3}} \sum_{t=0}^{N_3-1} \omega^{-tm} |t\rangle |\hat{v}_l^{(m)}\rangle$ *is the right singular vector corresponding to its singular value* $\hat{\sigma}_l^{(m)}$*, the diagonal entry of the* $\hat{S}^{(m)}$*. Similarly, the corresponding left singular vector is* $\frac{1}{\sqrt{N_3}} \sum_{t=0}^{N_3-1} \omega^{-tm} |t\rangle |\hat{u}_l^{(m)}\rangle$*.*

In some works, the singular values of $\hat{A}^{(m)}$ are defined as the singular values of the tensor $\mathcal{A}$, and most of the tensor reconstruction algorithms are achieved by truncating these singular values [93] because they are ordered and their theoretical basis is Lemma 4.1. Also, some low-rank tensor completion problems based on tensor factorization are solved by some optimization methods whose objective is to minimize the TNN of the tensor [93, 95]. Therefore, the singular values of $\hat{A}^{(m)}$ have wider applications. Inspired by this, our quantum t-svd outputs a state with the third register storing the estimated singular values of $\hat{A}^{(m)}$ superposed with the corresponding singular vectors.

3.3.2 Complexity analysis

For simplicity, we consider a tensor $\mathcal{A} \in \mathbb{R}^{N \times N \times N}$ with the same dimensions on each mode. In Steps 1 and 3 of Algorithm 3, performing the QFT or the inverse QFT parellelly on the third register of the state $|\mathcal{A}\rangle$ has the complexity of $\mathcal{O}((\log N)^2)$, compared with the complexity $\mathcal{O}(N^3 \log N)$ of the DFT performed on N^2 tubes of the tensor $\mathcal{A}$ in the classical t-svd algorithm. Moreover, in the classical t-svd, the

complexity of performing the matrix svd (Step 2 of Algorithm 1) for all frontal slices of $\hat{\mathcal{A}}$ is $\mathcal{O}(N^4)$. In contrast, in our quantum t-svd algorithm, this step is accelerated by Theorem 3.2 (the modified QSVE) whose complexity is $\mathcal{O}\left(N\text{polylog}(N)\right)$ on each frontal slice $\hat{A}^{(m)}$. Since we perform this modified QSVE on each $\hat{A}^{(m)}$ parallely, $m = 0, \cdots, N-1$, the running time of Step 2 is still $\mathcal{O}\left(N\text{polylog}(N)\right)$. Therefore, the total computational complexity of Algorithm 3 is $\mathcal{O}\left(N\text{polylog}(N)\right)$.

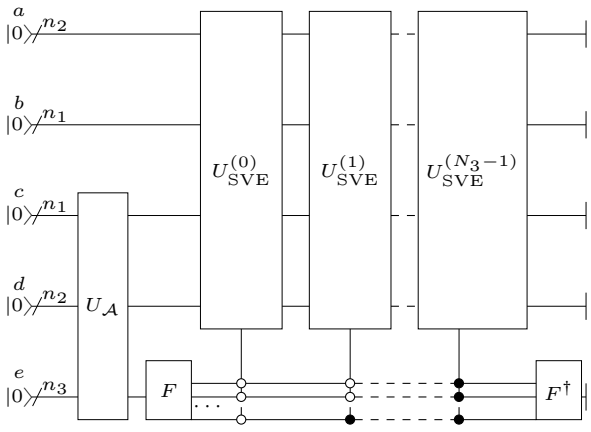


Figure 3.2: Circuit for Algorithm 3. $U_{\mathcal{A}}$ is the unitary operator for preparing the state $|\mathcal{A}\rangle$. The QFT is denoted by F. $N_i = 2^{n_i}, i = 1, 2, 3$. The blocks $U_{\text{SVE}}^{(m)}$ are further illustrated in FIG. 3.3.

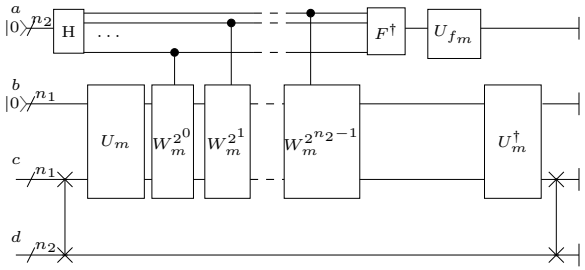


Figure 3.3: Circuit for $U_{\text{SVE}}^{(m)}$, $m = 0, \cdots, N_3 - 1$. The initial state of register c and d is $|\hat{A}^{(m)}\rangle$. U_m refers to $U_{\hat{Q}_m}$. U_{f_m} is a unitary operator implemented through oracle with a computable function $f_m(x)$. The notation is further explained in the proof of Theorem 4.1.

3.4 Quantum t-svd for order-p tensors

Following a similar procedure, we can extend the quantum t-svd for third-order tensors to order-p tensors easily.

We assume that the quantum state $|\mathcal{A}\rangle$ corresponding to the tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times \cdots \times N_p}$ can be prepared efficiently, where $N_i = 2^{n_i}$ with n_i being the number of qubits on the corresponding mode and

$$|\mathcal{A}\rangle = \sum_{i_1=0}^{N_1-1} \cdots \sum_{i_p=0}^{N_p-1} \mathcal{A}(i_1, \cdots, i_p) |i_1, \cdots, i_p\rangle . \tag{3.13}$$

Next, we perform the QFT on the third to the p-th mode of the state $|\mathcal{A}\rangle$, and then use $|m\rangle$ to denote $|m_3\rangle \cdots |m_p\rangle$, i.e., $m = m_3 \Pi_{i=4}^{p} N_i + m_4 \Pi_{i=5}^{p} N_i + \cdots + m_p$. The value of m ranges from 0 to $\iota - 1$, where $\iota = N_3 N_4 \cdots N_p + N_4 N_5 \cdots N_p + \cdots + N_p$.Specially, $\iota = \left(N^{p-1} - N\right)/(N-1)$ when $N_3 = \cdots = N_p = N$. Then we obtain

$$|\hat{\mathcal{A}}\rangle = \frac{1}{\sqrt{\iota}} \sum_{m=0}^{\iota-1} \sum_{i_1, \cdots, i_p} \omega^{\sum_{j=3}^{p} i_j m_j} \mathcal{A}(i_1, \cdots, i_p) |i_1, i_2\rangle |m\rangle . \tag{3.14}$$

Let the matrix

$$\hat{A}^{(m)} = \frac{1}{\sqrt{\iota}} \sum_{i_1, i_2} \sum_{i_3, \cdots, i_p} \omega^{\sum_{j=3}^{p} i_j m_j} \mathcal{A}(i_1, \cdots, i_p) |i_1\rangle \langle i_2|$$

and perform the modified QSVE on $\hat{A}^{(m)}$, $m = 0, \cdots, \iota - 1$, parallelly using the same strategy described in Section 3.3.1, we can get the state

$$|\psi_p\rangle = \sum_{m=0}^{\iota-1} \left(\sum_{l=0}^{r-1} \hat{\sigma}_l^{(m)} |\hat{u}_l^{(m)}\rangle |\hat{v}_l^{(m)}\rangle |\overline{\hat{\sigma}}_l^{(m)}\rangle \right) |m\rangle \tag{3.15}$$

after Step 2.

Finally, we recover the $|m_3\rangle \cdots |m_p\rangle$ expression and perform the inverse QFT on the p-th to the third register, obtaining the final state

$$|\phi_p\rangle = \frac{1}{(\sqrt{N})^{p-2}} \sum_{m_3=0}^{N_3-1} \cdots \sum_{m_p=0}^{N_p-1} \sum_{l=0}^{r-1} \hat{\sigma}_l^{(m)} \omega^{-\sum_{j=3}^{p} i_j m_j} |\hat{u}_l^{(m)}\rangle |\hat{v}_l^{(m)}\rangle |\overline{\hat{\sigma}}_l^{(m)}\rangle |i_3\rangle \cdots |i_p\rangle \tag{3.16}$$

corresponding to the quantum t-svd of order-p tensor $\mathcal{A}$.

Algorithm 4 Quantum t-svd for order-p tensors

Input: tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times \cdots \times N_p}$ prepared in a quantum state, precision $\epsilon_{\text{SVE}}^{(m)}$, $m = 0, \cdots, \iota - 1$.

Output: state $|\phi_p\rangle$.

1: Perform the QFT parallelly from the third to the p-th register of quantum state $|\mathcal{A}\rangle$, obtain the state $|\hat{\mathcal{A}}\rangle$.
2: Perform the modified QSVE for each matrix $\hat{A}^{(m)}$ with precision $\epsilon_{\text{SVE}}^{(m)}$ parallelly, $m = 0, \cdots, \iota - 1$, by using the controlled-U_{SVE} acting on the state $|\hat{\mathcal{A}}\rangle$, to obtain the state $|\psi_p\rangle$.
3: Perform the inverse QFT parallelly from the third to the p-th register of the above state and output the state $|\phi_p\rangle$.

For order-p tensor $\mathcal{A} \in \mathbb{R}^{N \times \cdots \times N}$, compared with the classical t-svd algorithm [45] whose time complexity is $\mathcal{O}(N^{p+1})$, our algorithm output a quantum state with the classical t-svd information encoded, and the time complexity of our quantum algorithm is $\mathcal{O}\left(N\text{polylog}(N)\right)$ followed by a similar analysis.

3.5 Conclusion

In this chapter, we present a quantum t-svd algorithm for third-order tensors with complexity $\mathcal{O}(N\text{polylog}(N))$. The key tools that accelerate this process are quantum Fourier transform and quantum singular value estimation, then we extend this third-order tensor algorithm to order-p tensors. Moreover, based on the quantum t-svd algorithm, a quantum 3D context-aware recommendations algorithm is proposed in Chapter 4, which achieves a polynomial speedup compared with the classical counterpart.

Chapter 4

Quantum context-aware recommendation systems

4.1 Introduction

It is well known that truncating k-term matrix svd provides the best rank-k approximation of a matrix in both ℓ_2 norm and Frobenius norm. This raises a question whether the truncated tensor decompositions also have the similar optimality. The CP decompostion expresses a tensor as a sum of outer products of vectors (rank-1 tensor), and CP-rank is defined as the minimal number of rank-1 tensor necessary to construct the tensor, but calculating the CP-rank and the rank-k CP approximation are numerically unstable. Also, truncated Tucker decomposition does not yield the best fit of the original tensor [90]. In contrast, the t-svd gives an optimal approximation of a tensor in Frobenius norm [90]. Unfortunately, the cost of computing t-svd factorization is prohibitively expensive especially for very high dimensional tensors, e.g., the cost is $\mathcal{O}(N^4 + N^3 \log N)$ for a third-order tensor with dimension N. Therefore, much work focuses on computing low-rank tensor approximations based on t-svd with comparatively low cost. For example, randomized tensor low-rank representations based on the t-svd can give the nearly optimal approximation with complexity $\mathcal{O}(kN^3 + N^3 \log N)$ for a third-order tensor with dimension N [90].

The quantum algorithm that we propose in this work implements a machine learning task, namely, context-aware recommendation systems in which preference information is encoded in a third-order tensor. For recommendation systems modeled by an $m \times n$ preference matrix, Kerenidis and Prakash designed a quantum algorithm that offers recommendations by just sampling from an approximated preference matrix [33]. Therefore, the running time is only $\mathcal{O}(\text{poly}(k)\text{polylog}(mn))$ if the preference matrix has a good rank-k approximation. To achieve this, they projected a state corresponding to a user's preferences to the approximated row space spanned by singular vectors whose corresponding singular values are greater than the prescribed threshold. After measuring this projected state in a computational basis, they got recommended product indices for the input user.

In recommendation systems algorithms, most model-based Collaborative Filtering approaches, e.g. matrix factorization, fail to model context information [31]. Context is an important factor to consider in personalized recommendation systems. In [88], it is demonstrated by an experiment that when a recommendation system is modeled by a tensor whose third dimension is context (e.g. time), accuracy could be improved compared to the non-contextual modeling. Hence, most recent research has focused on developing context-aware recommendation systems modeled by tensors; see, e.g., [77, 93, 61, 31]. The classical third-order tensor recommendation systems algorithms based on tensor factorizations, such as the truncated t-svd [93] and the truncated HOSVD (T-HOSVD) [77], all have the computational complexity at least $\mathcal{O}(kN^3)$.

Taking into account the effectiveness of third-order tensor modeling and high cost of the truncated t-svd algorithm, in this chapter we extend Kerenidis and Prakash's algorithm [33] from matrices to third-order tensors and propose a quantum context-aware recommendation systems algorithm based on truncated t-svd factorization. In general, a user's preference in a certain context is very likely to affect the recommen-

dation for him/her at other contexts. As the quantum Fourier transform (QFT) used in our t-svd algorithm is performed to bind a user's preferences in different context together, our quantum algorithm is well suited to context-aware recommendation systems. The general idea of our algorithm is to approximate the observed preference tensor using truncated matrix svd in the Fourier domain. Similar to the low-rank assumption adopted in quantum 2D recommendation systems [33], we exploit low structural and informational complexity of the data, expressed as low tubal-rank hypothesis of the underlying data, by which it is feasible to predict the missing entries. Moreover, taking advantage of the quantum parallelism, the quantum singular value estimation of all frontal slices after QFT can be performed parallelly, which dramatically reduces the time complexity to $\mathcal{O}(N\sqrt{k}\text{polylog}(N))$.

In Section 4.5, our numerical experiments show that the truncated t-svd is better than T-HOSVD and TT decompositions when applied to context-aware recommendation systems. In fact, our quantum algorithm suits the context-aware recommendation systems model very well since it is not necessary to reconstruct the entire tensor. All we need is to recommend products that a user prefers, which corresponds to measuring certain times a quantum state representing a user' approximated preference, so our quantum algorithm can provide good recommendations with much lower complexity.

The rest of this chapter is organized as follows. Preliminaries are given in Section 4.2. In Section 4.3, we propose our main algorithm, quantum context-aware recommendation systems algorithm. In Section 4.4, we calculate the quantum cost of QSVE algorithm and our algorithm. In Section 4.5, we numerically validate the classical counterpart of our model with real datasets. At last, we compare Tang's 2D quantum-inspired recommendation systems with our algorithm in Section 4.6 and we conclude this chapter in Section 4.7.

4.2 Preliminaries

In this section, we first present the definition of the tensor nuclear norm (TNN), and then introduce Lemma 4.1 which is the theoretical basis of our quantum algorithm for recommendation systems algorithm.

TNN is frequently used as an objective function to be minimized in many optimization algorithms for data completion [92, 93, 14, 95]. Since directly minimizing the tensor multi-rank (defined as a vector whose i-th entry is the rank of $\hat{A}^{(i)}$) is NP-hard, some works approximate the rank function by its convex surrogate, i.e., TNN [93]. It is proved that TNN is the tightest convex relaxation to l_1 norm of the tensor multi-rank [93] and the problem is reduced to a convex one when transformed into minimizing TNN.

Definition 4.1. *[93]* ***Tensor nuclear norm***
The tensor nuclear norm (TNN) of $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, denoted by $||\mathcal{A}||_{TNN}$, is defined as the sum of the singular values of $\hat{A}^{(l)}$, i.e., $||\mathcal{A}||_{TNN} = \sum_{l=0}^{N_3-1} ||\hat{A}^{(l)}||_*$, where $||\cdot||_*$ refers to the matrix nuclear norm, namely the sum of the singular values.

Many important applications of the t-svd algorithm, such as data compression and completion, utilize the optimality of truncated t-svd in the sense that it gives an optimal approximation of a tensor measured by the Frobenius norm. The following theorem describes this property, which is the theoretical basis of our quantum algorithm for recommendation systems to be developed in Sections 4.3.2.

Lemma 4.1. *[34, 90] Suppose the t-svd of the tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$. Then we have*

$$\mathcal{A} = \sum_{i=0}^{\min(N_1,N_2)-1} \mathcal{U}(:,i,:) * \mathcal{S}(i,i,:) * \mathcal{V}(:,i,:)^T,$$

where the matrices $\mathcal{U}(:,i,:)$ and $\mathcal{V}(:,i,:)$ and the vector $\mathcal{S}(i,i,:)$ are regarded as tensors of order 3. For $1 \leqslant k < \min(N_1, N_2)$, define $\mathcal{A}_k \triangleq \sum_{i=0}^{k-1} \mathcal{U}(:,i,:) * \mathcal{S}(i,i,:) * \mathcal{V}(:$

$, i, :)^T$. *Then*

$$\mathcal{A}_k = \arg\min_{\tilde{\mathcal{A}} \in \mathcal{M}_k} ||\mathcal{A} - \tilde{\mathcal{A}}||_F, \tag{4.1}$$

where $\mathcal{M}_k = \{\mathcal{X} * \mathcal{Y} | \mathcal{X} \in \mathbb{R}^{N_1 \times k \times N_3}, \mathcal{Y} \in \mathbb{R}^{k \times N_2 \times N_3}\}$. *Therefore,* $||\mathcal{A} - \mathcal{A}_k||_F$ *is the theoretical minimal error, given by* $||\mathcal{A} - \mathcal{A}_k||_F = \sqrt{\sum_{i=k}^{\min(N_1, N_2)-1} ||\mathcal{S}(i, i, :)||_2^2}$.

4.3 Quantum algorithm for recommendation systems modeled by third-order tensors

In this section, we will first introduce the notation adopted in this section and then give an overview of Algorithm 5. In Section 4.3.1, the main ideas and assumptions of the algorithm are summarized. In Section 4.3.2, we explain each step of our algorithm in detail, followed by a summary in Algorithm 5. Error analysis is given in Section 4.3.3 and complexity analysis is conducted in Section 4.3.4.

The preference information of users is stored in a third-order tensor $\mathcal{A} \in \mathbb{R}^{N \times N \times N}$, called the preference tensor, whose three modes represent user(i), product(j) and context(t) respectively. The tube $\mathcal{A}(i, j, :)$ is regarded as the rating scores of the user i for the product j under different contexts. For user i in context t, the entry $\mathcal{A}(i, j, t)$ takes value 1 indicating the product j is "good" and value 0 otherwise. In this sense, a triplet (i, j, t) is called a good recommendation if $\mathcal{A}(i, j, t) = 1$ or a bad recommendation otherwise. Let tensor $\mathcal{T}$ be the random tensor obtained by sampling from the tensor $\mathcal{A}$ with probability p and $\hat{\mathcal{T}}$ be the tensor obtained by performing the QFT along the third mode of $\mathcal{T}$. We use $\hat{T}^{(m)}$ to denote the m-th frontal slice of tensor $\hat{\mathcal{T}}$. $\hat{T}^{(m)}_{\geqslant \tau_m}$ is formed by projecting $\hat{T}^{(m)}$ onto the space spanned by the singular vectors whose corresponding singular values are greater than the threshold τ_m. $\hat{\mathcal{T}}_{\geqslant \boldsymbol{\tau}}$ denotes the tensor whose m-th frontal slice is $\hat{T}^{(m)}_{\geqslant \tau_m}$. Here, the threshold $\boldsymbol{\tau}$ of tensor $\hat{\mathcal{T}}$ actually denotes a list of thresholds $\{\tau_0, \cdots, \tau_{N-1}\}$ since different frontal slices

$\hat{T}^{(m)}$ have their corresponding thresholds τ_m. $\mathcal{T}_{\geqslant \boldsymbol{\tau}}$ denotes the tensor obtained by performing the inverse QFT along the third mode of $\hat{\mathcal{T}}_{\geqslant \boldsymbol{\tau}}$.

4.3.1 Main ideas

We will propose Algorithm 5 which recommends a product j to a user i at a certain context t_0. The algorithm is inspired by the matrix recommendation systems method [1, 33] and a tensor reconstruction algorithm [93]. The main ideas are summarized in the following flow chart:

$$\mathcal{T}(i,:,:) \xrightarrow{\text{QFT}} \hat{\mathcal{T}}(i,:,:) \xrightarrow{\text{tube}} \hat{\mathcal{T}}(i,:,m) \xrightarrow{\text{approximation}}$$

$$\hat{\mathcal{T}}_{\geqslant \boldsymbol{\tau}}(i,:,m) \xrightarrow{\text{stack up}} \hat{\mathcal{T}}_{\geqslant \boldsymbol{\tau}}(i,:,:) \xrightarrow{\text{iQFT}} \mathcal{T}_{\geqslant \boldsymbol{\tau}}(i,:,:).$$

Suppose there is a hidden preference tensor $\mathcal{A}$ which is assumed to have a low tubal-rank k. This low tubal-rank assumption is also adopted in the classical truncated t-svd data completion problem [92, 75]. In practical applications, only a part of the entries of $\mathcal{A}$ can be observed. Our goal is to predict the missing entries and recommend product which has a high predicted value. This partially observed tensor is called the subsample tensor $\mathcal{T}$ which is sparse in general. Here, we use the subsampling method proposed in [1] to get $\mathcal{T}$, which is also adopted in [33]. Specifically, $\mathcal{T}_{ijt} = \mathcal{A}_{ijt}/p$ with probability p and $\mathcal{T}_{ijt} = 0$ otherwise, where p is called the subsampling probability. Clearly, the expectation $\mathrm{E}(\mathcal{T}) = \mathcal{A}$. In this sense, our algorithm can be understood as an approximate reconstruction of the hidden preference tensor $\mathcal{A}$, of which we are only given a sample ratings in the form of the sparse tensor $\mathcal{T}$.

Our algorithm recommends products according to the low tubal-rank tensor $\mathcal{T}_{\geqslant \boldsymbol{\tau}}$ which can be proved to an approximation of the hidden preference tensor $\mathcal{A}$. Inspired by the t-svd, this approximation process is conducted under the Fourier domain, as shown in the above flow chart. Next we explain each step of the flow chart in detail. Given a state $|\mathcal{T}(i,:,:)\rangle$ representing the observed preference information of user i, we

first perform QFT on its last register, obtaining the state $|\hat{\mathcal{T}}(i,:,:)\rangle$. For each $\hat{T}^{(m)}$, which is the m-th frontal slice of the tensor $\hat{\mathcal{T}}$, we perform the modified QSVE on this matrix using the input state $|\hat{\mathcal{T}}(i,:,m)\rangle$ and truncate the resulting singular values with threshold τ_m, $m = 0, \cdots, N_3 - 1$ obtaining the state $|\hat{\mathcal{T}}_{\geqslant \boldsymbol{\tau}}(i,:,m)\rangle$. Stacking tubes $\hat{\mathcal{T}}_{\geqslant \boldsymbol{\tau}}(i,:,m)$ $(m = 0, \ldots, N-1)$ yields the horizontal slice $\hat{\mathcal{T}}_{\geqslant \boldsymbol{\tau}}(i,:,:)$ which can be regarded as an approximation of $\hat{\mathcal{T}}(i,:,:)$. After the inverse QFT on $\hat{\mathcal{T}}_{\geqslant \boldsymbol{\tau}}(i,:,:)$, the horizontal slice $\mathcal{T}_{\geqslant \boldsymbol{\tau}}(i,:,:)$ is obtained. We prove in Section 4.3.3 that $\mathcal{T}_{\geqslant \boldsymbol{\tau}}(i,:,:)$ is an approximation of $\mathcal{T}(i,:,:)$, hence it can approximate user i's preference $\mathcal{A}(i,:,:)$.

The approximation tensor $\mathcal{T}_{\geqslant \boldsymbol{\tau}}$ is non-sparse in general, thus we can get a recommended index based on the non-zero entries of $\mathcal{T}_{\geqslant \boldsymbol{\tau}}$. For a given context t, our algorithm provides a recommended product index j for a user i by just measuring the output quantum state $|\mathcal{T}_{\geqslant \boldsymbol{\tau}}(i,:,:)\rangle$ in the computational basis. In Theorem 4.2, we provide an upper bound on the probability of the pair (i,j,t) being a bad recommendation. This upper bound could be small by taking reasonable values for the related parameters.

Assumption 3. *The following assumptions are used in Algorithm 5.*

1. *Every frontal slice of the subsample tensor $\mathcal{T} \in \mathbb{R}^{N\times N\times N}$ is stored in the data structure as mentioned in Lemma 2.1.*

2. *For all $i, m = 0, \cdots, N-1$, we assume the tubes $\mathcal{A}(i,:,m)$ satisfy*

$$\frac{1}{1+\gamma}\frac{||\mathcal{A}||_F^2}{N^2} \leqslant ||\mathcal{A}(i,:,m)||_2^2 \leqslant (1+\gamma)\frac{||\mathcal{A}||_F^2}{N^2} \tag{4.2}$$

   for a given $\gamma > 0$.

The second assumption indicates that users are all typical users, that is, the number of preferred products of users is close to the average in any context m. These assumptions are also adopted in [33] for matrices, where they explain the rationality of these assumptions.

4.3.2 The algorithm

Given the hidden preference tensor $\mathcal{A}$, the sampling probability p , the assumed low tubal-rank k, and the precision $\epsilon_{\text{SVE}}^{(m)}$ of the modified QSVE on each $\hat{T}^{(m)}$, Algorithm 5 outputs the state corresponding to the approximation of the i-th horizontal slice $\mathcal{A}(i,:,:)$. The quantum circuits of Algorithm 5 is shown in FIG. 4.1. In what follows we explain the steps of this algorithm.

The dynamic preference tensor $\mathcal{A} \in \mathbb{R}^{N\times N\times N}$ can be interpreted as the preference matrix $\mathcal{A}(:,:,t)$ evolving over the context t. It is reasonable to believe that the tubes $\mathcal{A}(i,:,0),\cdots,\mathcal{A}(i,:,N-1)$ are related to each other because the preference of the same user i in different contexts is mutually influenced. Considering these relations, we merge tubes in the same horizontal slice together through the QFT after getting the subsample tensor $\mathcal{T}$. In other words, in **Step 1**, the QFT is performed on the last register of the input state

$$|\mathcal{T}(i,:,:)\rangle = \frac{1}{||\mathcal{T}(i,:,:)||_F}\sum_{j,t=0}^{N-1}\mathcal{T}(i,j,t)\,|j\rangle^d\,|t\rangle^e \tag{4.3}$$

to get

$$|\hat{\mathcal{T}}(i,:,:)\rangle = \frac{1}{||\mathcal{T}(i,:,:)||_F}\sum_{m=0}^{N-1}||\hat{\mathcal{T}}(i,:,m)||_2\,|\hat{\mathcal{T}}(i,:,m)\rangle^d\,|m\rangle^e\,, \tag{4.4}$$

where $\omega = e^{2\pi \mathrm{i}/N}$ and

$$|\hat{\mathcal{T}}(i,:,m)\rangle = \frac{1}{\sqrt{N}||\hat{\mathcal{T}}(i,:,m)||_2}\sum_{j,t=0}^{N-1}\omega^{tm}\mathcal{T}(i,j,t)\,|j\rangle\,. \tag{4.5}$$

Note that $||\hat{\mathcal{T}}(i,:,:)||_F = ||\mathcal{T}(i,:,:)||_F$, since the Frobenius norm does not change under the Fourier transform. In FIG. 4.1, the input state $|\mathcal{T}(i,:,:)\rangle$ is represented in lines d and e with $\lceil \log N \rceil$ qubits, and the QFT is denoted as F in line e.

In **Step 2**, a controlled operator

$$U = \sum_{m=0}^{N-1} U_{\mathrm{SVE}}^{(m)} \otimes |m\rangle\langle m|^e \tag{4.6}$$

is performed on the state $|\hat{\mathcal{T}}(i,:,:)\rangle$, where $U_{\mathrm{SVE}}^{(m)}$ denotes the modified quantum singular value estimation process on the matrix $\hat{T}^{(m)}$ with the input $|\hat{\mathcal{T}}(i,:,m)\rangle$. As $U_{\mathrm{SVE}}^{(m)}$ follows similar steps of QSVE, its circuit is also similar to that of QSVE given in FIG. 4.2. The quantum cost of implementing the operator U is analyzed in Appendix 4.4.2. Because of the quantum parallelism, the operator U performed on the superposition state $|\hat{\mathcal{T}}(i,:,:)\rangle$ is thus equivalent to $U_{\mathrm{SVE}}^{(m)}$ performed on each of the components $|\hat{\mathcal{T}}(i,:,m)\rangle$ as a single input, i.e.,

$$U\,|\hat{\mathcal{T}}(i,:,:)\rangle = \frac{1}{||\mathcal{T}(i,:,:)||_F} \sum_{m=0}^{N-1} ||\hat{\mathcal{T}}(i,:,m)||_2 \left(U_{\mathrm{SVE}}^{(m)}\,|\hat{\mathcal{T}}(i,:,m)\rangle^d\right) |m\rangle^e . \tag{4.7}$$

Next, we focus on $U_{\mathrm{SVE}}^{(m)}\,|\hat{\mathcal{T}}(i,:,m)\rangle^d$ in (4.7). Suppose $\sum_{j=0}^{N-1} \hat{\sigma}_j^{(m)} \hat{u}_j^{(m)} \hat{v}_j^{(m)\dagger}$ is the svd of $\hat{T}^{(m)}$, we first express $|\hat{\mathcal{T}}(i,:,m)\rangle$ under the basis of $\hat{v}_j^{(m)}, j = 0, \cdots, N-1$, i.e.,

$$|\hat{\mathcal{T}}(i,:,m)\rangle = \sum_{j=0}^{N-1} \beta_j^{(im)}\,|\hat{v}_j^{(m)}\rangle, \tag{4.8}$$

then we perform the modified QSVE $U_{\mathrm{SVE}}^{(m)}$ on $\hat{T}^{(m)}$, The detail of this operation is further illustrated in the following theorem.

Theorem 4.1. *Given every frontal slice of $\mathcal{T}$ stored in the data structure as mentioned in Lemma 2.1, there is a quantum algorithm, denoted as $U_{\mathrm{SVE}}^{(m)}$, that uses the input $|\hat{\mathcal{T}}(i,:,m)\rangle$ in the form (4.8) and outputs the state*

$$\sum_j \beta_j^{(im)}\,|\hat{v}_j^{(m)}\rangle^d\,|\bar{\hat{\sigma}}_j^{(m)}\rangle^b \tag{4.9}$$

with probability at least $1-1/\mathrm{poly}(N)$*, where* $\hat{v}_j^{(m)}$ *is the right singular vector of* $\hat{T}^{(m)}$*, and* $\overline{\hat{\sigma}}_j^{(m)}$ *is an estimate of* $\hat{\sigma}_j^{(m)}$ *satisfying* $|\overline{\hat{\sigma}}_j^{(m)} - \hat{\sigma}_j^{(m)}| \leqslant \epsilon_{\mathrm{SVE}}^{(m)} ||\hat{T}^{(m)}||_F$*. The running time to implement* $U_{\mathrm{SVE}}^{(m)}$ *is* $\mathcal{O}\left(N\mathrm{polylog}N/\epsilon_{\mathrm{SVE}}^{(m)}\right)$.

Before proving Theorem 4.1, we would like to sketch the proof first. According to Assumption 3, each frontal slice $T^{(k)}$ is stored in the binary tree structure. Hence, based on the proof of Lemma 2.2 in [33], the states $|\mathcal{T}(i,:,k)\rangle$, corresponding to the i-th row of $T^{(k)}$, can be prepared efficiently by operators P_k. Based on these operators, two new isometries $\hat{P}_m$ and $\hat{Q}_m$ are constructed in order to perform QSVE on $\hat{T}^{(m)}$. The detail of the proof is given below.

Proof. Since every $T^{(k)}$, $k = 0, \cdots, N-1$, is stored in the binary tree structure, the quantum computer can perform the following mapping in $\mathcal{O}(\mathrm{polylog}(N))$ time, as shown in Theorem 5.1 in [33]:

$$U_{P_k} : |i\rangle|0\rangle \quad \to |i\rangle|\mathcal{T}(i,:,k)\rangle = \frac{1}{||\mathcal{T}(i,:,k)||_2} \sum_{j=0}^{N-1} \mathcal{T}_{ijk} |i\rangle|j\rangle, \tag{4.10}$$

where $\mathcal{T}(i,:,k)$ is the i-th row of $T^{(k)}$.

Define the degenerate operator $P_k \in \mathbb{R}^{N^2 \times N}$ related to U_{P_k} as

$$P_k : \quad |i\rangle \to |i\rangle|\mathcal{T}(i,:,k)\rangle. \tag{4.11}$$

That is,

$$P_k = \sum_{i=0}^{N-1} |i\rangle|\mathcal{T}(i,:,k)\rangle\langle i|. \tag{4.12}$$

Based on the efficiently implemented operator U_{P_k}, we define another operator

$$U_{\hat{P}_m} \triangleq \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{km} U_{P_k}. \tag{4.13}$$

It can be easily seen that the operator $U_{\hat{P}_m}$ achieves the state preparation of the rows of the matrix $\hat{T}^{(m)}$, i.e., $U_{\hat{P}_m} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \sum_{i=0}^{N-1} \omega^{km} |i\rangle |\mathcal{T}(i,:,k)\rangle \langle i| \langle 0| = \sum_i |i\rangle |\hat{\mathcal{T}}(i,:,m)\rangle \langle i| \langle 0|$, where $|\hat{\mathcal{T}}(i,:,m)\rangle$ is the state of the i-th row of $\hat{T}^{(m)}$. Similarly, the isometry corresponding to $U_{\hat{P}_m}$ is $\hat{P}_m = \sum_i |i\rangle |\hat{\mathcal{T}}(i,:,m)\rangle \langle i|$. It can be easily shown that $\hat{P}_m$ is an isometry since $\hat{P}_m^\dagger \hat{P}_m = I_N$. Since U_{P_k} can be implemented in time $\mathcal{O}(\text{polylog}(N))$, $U_{\hat{P}_m}$ can be implemented in time $\mathcal{O}(N\text{polylog}N)$ using the linear combination of unitaries (LCU) technique [8, 72, 40, 73, 39].

The LCU technique was first proposed by Long in their work [39] in a more general form, and Shao summarized this result in [73]. The problem of LCU can be formulated as follows: Given $\alpha_i \in \mathbb{C}$ and unitary operators U_i, $i = 0, 1, \cdots, N-1$, implement linear operator $L = \sum_{j=0}^{N-1} \alpha_j U_j$. The algorithm stated in [8] implements L in time $\mathcal{O}((T_{\text{in}} + \log N) N \max_j |\alpha_j| / ||L |\psi\rangle||)$, where $|\psi\rangle$ is any given initial state and T_{in} is the time to implement $U_0, U_1, \cdots, U_{N-1}$. In our case, $T_{\text{in}} = N\text{polylog}N$, $\alpha_j = \omega^{km}/\sqrt{N}$ and the input state is chosen as $|\psi\rangle = \sum_{i=0}^{N} |i\rangle |0\rangle$. Thus, $||L |\psi\rangle|| = ||\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{km} U_{P_k} |\psi\rangle|| = ||\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \sum_{i=0}^{N-1} \omega^{km} |i\rangle |\mathcal{T}(i,:,k)\rangle|| = \sqrt{N}$. Therefore, the complexity to implement $U_{\hat{P}_m}$ is $\mathcal{O}(N\text{polylog}N)$.

Next, we define the mapping

$$U_{\hat{Q}_m} : |0\rangle |j\rangle \to |\boldsymbol{s}_{\hat{T}^{(m)}}\rangle |j\rangle = \frac{1}{||\hat{T}^{(m)}||_F} \sum_i ||\hat{\mathcal{T}}(i,:,m)||_2 |i\rangle |j\rangle, \tag{4.14}$$

where $\boldsymbol{s}_{\hat{T}^{(m)}}$ is a vector whose i-th entry is $\frac{||\hat{\mathcal{T}}(i,:,m)||}{||\hat{T}^{(m)}||_F}$. Similar with $U_{\hat{P}_m}$, the corresponding isometry is defined as $\hat{Q}_m = \sum_j |\boldsymbol{s}_{\hat{T}^{(m)}}\rangle |j\rangle \langle j|$ satisfying $\hat{Q}_m^\dagger \hat{Q}_m = I_N$, which can be verified easily.

Now we perform QSVE on the matrix $\hat{T}^{(m)}$. First, the factorization $\frac{\hat{T}^{(m)}}{||\hat{T}^{(m)}||_F} = \hat{P}_m^\dagger \hat{Q}_m$ can be easily verified. Second, we can prove that $2\hat{P}_m \hat{P}_m^\dagger - I_{N^2}$ is a reflection

and it be implemented through $U_{\hat{P}_m}$. Actually,

$$\begin{aligned} 2\hat{P}_m\hat{P}_m^\dagger - I_{N^2} =& 2\sum_i |i\rangle |\hat{\mathcal{T}}(i,:,m)\rangle\langle i| \langle\hat{\mathcal{T}}(i,:,m)| - I_{N^2} \\ =& U_{\hat{P}_m}\left[2\sum_i |i\rangle|0\rangle\langle i|\langle 0| - I_{N^2}\right]U_{\hat{P}_m}^\dagger, \end{aligned} \tag{4.15}$$

where $2\sum_i |i\rangle|0\rangle\langle i|\langle 0| - I_{N^2}$ is a reflection. The similar result holds for $2\hat{Q}_m\hat{Q}_m^\dagger - I_{N^2}$.

Now denote

$$W_m = \left(2\hat{P}_m\hat{P}_m^\dagger - I_{N^2}\right)\left(2\hat{Q}_m\hat{Q}_m^\dagger - I_{N^2}\right). \tag{4.16}$$

Let $\hat{T}^{(m)} = \sum_{i=0}^{r-1}\hat{\sigma}_i^{(m)}\hat{u}_i^{(m)}\hat{v}_i^{(m)\dagger}$ be the singular value decomposition of $\hat{T}^{(m)}$. We can prove that the subspace spanned by $\{\hat{Q}_m|\hat{v}_i^{(m)}\rangle, \hat{P}_m|\hat{u}_i^{(m)}\rangle\}$ is invariant under the unitary transformation W_m:

$$W_m\hat{Q}_m|\hat{v}_i^{(m)}\rangle = \frac{2\hat{\sigma}_i^{(m)}}{||\hat{T}^{(m)}||_F}\hat{P}_m|\hat{u}_i^{(m)}\rangle - Q|\hat{v}_i^{(m)}\rangle,$$

$$W_m\hat{P}_m|\hat{u}_i^{(m)}\rangle = \left(\frac{4\hat{\sigma}_i^2}{||\hat{T}^{(m)}||_F} - 1\right)\hat{P}_m|\hat{u}_i^{(m)}\rangle - \frac{2\hat{\sigma}_i}{||\hat{T}^{(m)}||_F}\hat{Q}_m|\hat{v}_i^{(m)}\rangle.$$

The matrix W_m can be calculated under an orthonormal basis using the Schmidt orthogonalization. It is a rotation in the subspace spanned by its eigenvectors $|\omega_{i\pm}^{(m)}\rangle$ with correspondent eigenvalues $e^{\pm i\theta_i^{(m)}}$, where $\theta_i^{(m)}$ is the rotation angle satisfying

$$\cos(\theta_i^{(m)}/2) = \frac{\hat{\sigma}_i^{(m)}}{||\hat{T}^{(m)}||_F}, \tag{4.17}$$

that is,

$$\begin{aligned} \hat{Q}_m|\hat{v}_i^{(m)}\rangle &= \sqrt{2}\left(|\omega_{i+}^{(m)}\rangle + |\omega_{i-}^{(m)}\rangle\right), \\ \hat{P}_m|\hat{u}_i^{(m)}\rangle &= \sqrt{2}\left(e^{i\theta_i/2}|\omega_{i+}^{(m)}\rangle + e^{-i\theta_i/2}|\omega_{i-}^{(m)}\rangle\right). \end{aligned}$$

Here, we choose the input state $|\hat{\mathcal{T}}(i,:,m)\rangle$ represented in (4.8), then

$$\hat{Q}_m |\hat{\mathcal{T}}(i,:,m)\rangle = \sum_{j=0}^{N-1} \sqrt{2}\beta_j^{(im)} \left(|\omega_{i+}^{(m)}\rangle + |\omega_{i-}^{(m)}\rangle \right). \tag{4.18}$$

Performing the phase estimation on W_m with running time $\mathcal{O}\left(N\text{polylog}N/\epsilon_{\text{SVE}}^{(m)}\right)$, and computing the estimated singular value of $\hat{T}^{(m)}$ through oracle with a computable function $f(x) = ||\hat{T}^{(m)}||_F \cos(x/2)$. According to the relations between $\theta_i^{(m)}$ and $\hat{\sigma}_i^{(m)}$ in (4.17), we obtain

$$\sum_{j=0}^{N-1} \sqrt{2}\beta_j^{(im)} \left(|\omega_{i+}^{(m)}\rangle |\overline{\theta}_i^{(m)}\rangle + |\omega_{i-}^{(m)}\rangle |-\overline{\theta}_i^{(m)}\rangle \right) |\overline{\sigma}_i^{(m)}\rangle. \tag{4.19}$$

we next uncompute the phase estimation procedure and then apply the inverse of $U_{\hat{Q}_m}$ to obtain the desired state (4.9) in Theorem 4.1. □

Similar with the quantum singular value decomposition for matrices [65] that the output allows singular values and associated singular vectors to be revealed in a quantum form, the state in (4.9) also finds the estimated singular values of $\hat{T}^{(m)}$ and store them in the third register, superposed with corresponding singular vectors. Therefore, combining (4.7) and (4.9), the state after **Step 2** is

$$\frac{1}{||\mathcal{T}(i,:,:)||_F} \sum_{m,j} ||\hat{\mathcal{T}}(i,:,m)||_2 \beta_j^{(im)} \left|\hat{v}_j^{(m)}\right\rangle^d \left|\overline{\hat{\sigma}}_j^{(m)}\right\rangle^b |m\rangle^e \triangleq |\xi_1\rangle, \tag{4.20}$$

In **Steps 3-5**, our goal is to project each tube $\hat{\mathcal{T}}(i,:,m)$ onto the subspace spanned by the right singular vectors $\hat{v}_j^{(m)}$ corresponding to singular values greater than the threshold τ_m. As shown in FIG. 4.1, in Step 3, we first add an ancillary register $|0\rangle^a$ and then apply a unitary operator

$$V = \sum_{m=0}^{N-1} V^{(m)} \otimes |m\rangle\langle m|^e \tag{4.21}$$

acting on the registers b and a controlled by the register e, where $V^{(m)}$ is a 2-qubit conditional rotation gate that maps $|h\rangle^b|0\rangle^a \to |h\rangle^b|1\rangle^a$ if $h < \tau_m$ and $|h\rangle^b|0\rangle^a \to |h\rangle^b|0\rangle^a$ otherwise. Therefore, after **Step 3**, we get

$$|\xi_2\rangle = \frac{1}{||\mathcal{T}(i,:,:)||_F}\sum_{m=0}^{N-1}||\hat{\mathcal{T}}(i,:,m)||_2\left(\sum_{j:\bar{\tilde{\sigma}}_j^{(m)}\geqslant\tau_m}\beta_j^{(im)}\,|\hat{v}_j^{(m)}\rangle^d\,|\bar{\tilde{\sigma}}_j^{(m)}\rangle^b\,|0\rangle^a + \sum_{j:\bar{\tilde{\sigma}}_j^{(m)}<\tau_m}\beta_j^{(im)}\,|\hat{v}_j^{(m)}\rangle^d\,|\bar{\tilde{\sigma}}_j^{(m)}\rangle^b\,|1\rangle^a\right)|m\rangle^e\,. \tag{4.22}$$

In **Step 4**, we apply the inverse modified QSVE which is denoted as $U_{\text{SVE}}^{(m)\dagger}$ in FIG. 4.1, and discard the register b. Then we measure line a, i.e. the last register of (4.22), and postselect the outcome $|0\rangle^a$, getting

$$|\xi_3\rangle = \frac{1}{\alpha}\sum_{m=0}^{N-1}\sum_{j,\geqslant\tau_m}\beta_j^{(im)}||\hat{\mathcal{T}}(i,:,m)||_2\,|\hat{v}_j^{(m)}\rangle^d\,|m\rangle^e\,, \tag{4.23}$$

where $\alpha = \sqrt{\sum_{m=0}^{N-1}\sum_{j,\geqslant\tau_m}||\hat{\mathcal{T}}(i,:,m)||_2^2\cdot|\beta_j^{(im)}|^2}$. The probability that we obtain the outcome $|0\rangle$ in **Step 4** is

$$\frac{||\hat{\mathcal{T}}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F^2}{||\mathcal{T}(i,:,:)||_F^2}, \tag{4.24}$$

where $\hat{\mathcal{T}}_{\geqslant\boldsymbol{\tau}}$ denotes the tensor whose m-th frontal slice is $\hat{T}_{\geqslant\tau_m}^{(m)}$ obtained by truncating $\hat{T}^{(m)}$ with threshold τ_m. Hence, based on amplitude amplification, we have to repeat the measurement $\mathcal{O}\left(\frac{||\mathcal{T}(i,:,:)||_F}{||\hat{\mathcal{T}}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F}\right)$ times to ensure the success probability of getting the outcome $|0\rangle$ is close to 1.

Comparing (4.8) with (4.23), we find that the unnormalized state $\sum_{j,\geqslant\tau_m}\beta_j^{(im)}||\hat{\mathcal{T}}(i,:,m)||_2\,|\hat{v}_j^{(m)}\rangle$ can be seen as an approximation of $\hat{\mathcal{T}}(i,:,m)$, $m=0,\cdots,N-1$. Hence, $|\xi_3\rangle$ corresponds to an approximation of $|\hat{\mathcal{T}}(i,:,:)\rangle$.

In **Step 5**, we perform the inverse QFT, denoted as $F^\dagger$ in line e of FIG. 4.1, on $|\xi_3\rangle$ in (4.23) to get the final state

$$|\xi_4\rangle=\frac{1}{\alpha\sqrt{N}}\sum_{t,m=0}^{N-1}\sum_{j,\geqslant\tau_m}\beta_j^{(im)}\omega^{-tm}||\hat{\mathcal{T}}(i,:,m)||_2\,|\hat{v}_j^{(m)}\rangle^d\,|t\rangle^e\,, \tag{4.25}$$

which corresponds to an approximation of $\mathcal{T}(i,:,:)$, and thus it can also be regarded as an approximation of $\mathcal{A}(i,:,:)$.

In **the last step**, user i is recommended a product j varying with different contexts as needed by measuring the output state $|\xi_4\rangle$. For example, if we need the recommended index at a certain context t_0, we can first measure the last register of $|\xi_4\rangle$ in the computational basis and postselect the outcome $|t_0\rangle$ in line e, as is shown in FIG. 4.1, obtaining the state propositional to (unnormalized)

$$\sum_{m=0}^{N-1}\sum_{j,\geqslant\tau_m}\beta_j^{(im)}\omega^{-t_0m}||\hat{\mathcal{T}}(i,:,m)||_2\,|\hat{v}_j^{(m)}\rangle^d\,. \tag{4.26}$$

We next measure this state in the computational basis to get an index j which is proved to be a good recommendation for user i at context t_0.

Algorithm 5 is summarized below, whose circuit is shown in FIG. 4.1.

Algorithm 5 Quantum algorithm for recommendation systems modeled by third-order tensors

Require: Assumption 3, a user index i, the state $|\mathcal{T}(i,:,:)\rangle$ corresponding to the preference information of user i, precision $\epsilon_{\text{SVE}}^{(m)}$, the truncation threshold τ_m, $m = 0, \cdots, N-1$, and a context t_0.

Output: the recommended index j for the user i at the context t_0.

1: Perform the QFT on the last register of the input state $|\mathcal{T}(i,:,:)\rangle$, to obtain $|\hat{\mathcal{T}}(i,:,:)\rangle$ in (4.4).
2: Perform the modified QSVE on the matrix $\hat{T}^{(m)}$ parallelly, using the input $|\hat{\mathcal{T}}(i,:,:)\rangle$ with precision $\epsilon_{\text{SVE}}^{(m)}$, $m = 0, \cdots, N-1$, to get the state $|\xi_1\rangle$ defined in (4.20).
3: Add an ancilla qubit $|0\rangle^a$ and apply a unitary transformation V in (4.21) to obtain the state $|\xi_2\rangle$ in (4.22).
4: Apply the inverse modified QSVE and discard the register b, then measure the ancilla register a in the computational basis and postselect the outcome $|0\rangle$, then delete the register a, obtaining the state $|\xi_3\rangle$ in (4.23).
5: Perform the inverse QFT on the register e, to get $|\xi_4\rangle$ in (4.25).
6: Measure the register e in the computational basis and postselect the outcome $|t_0\rangle$. Then measure the register d in the computational basis to get the index j.

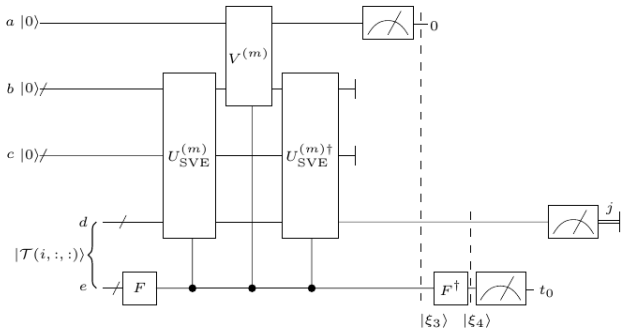


Figure 4.1: Circuit for Algorithm 5, and the process of $U_{\text{SVE}}^{(m)}$ is given in the proof of Theorem 4.1.

4.3.3 Error analysis

In this section, the i-th horizontal slice of the tensor $\mathcal{T}_{\geqslant \boldsymbol{\tau}}$ is proved to be an approximation of $\mathcal{A}(i,:,:)$. Then sampling from the matrix $\mathcal{T}_{\geqslant \boldsymbol{\tau}}(i,:,:)$ yields good recommendations for user i; see Theorem 4.2. The following two lemmas, Lemmas

4.2 and 4.3, are used in the proof of Theorem 4.2.

Lemma 4.2. *[33] Let $\tilde{A}$ be an approximation of the matrix A such that $||A-\tilde{A}||_F \leqslant \epsilon||A||_F$. Then, the probability that sampling from $\tilde{A}$ provides a bad recommendation is*

$$\Pr_{(i,j)\sim\tilde{A}}[(i,j)\text{bad}] \leqslant \left(\frac{\epsilon}{1-\epsilon}\right)^2. \tag{4.27}$$

Lemma 4.3. *Let $A \in \mathbb{R}^{N\times N}$ be a matrix and A_k be the best rank-k approximation such that $||A-A_k||_F \leqslant \epsilon||A||_F$. If the threshold for truncating the singular values of A is chosen as $\sigma = \frac{\epsilon||A||_F}{\sqrt{k}}$, then*

$$||A - A_{\geqslant\sigma}||_F \leqslant 2\epsilon||A||_F. \tag{4.28}$$

Proof. Let σ_i denote the singular value of A and l be the largest integer for which $\sigma_l \geqslant \frac{\epsilon||A||_F}{\sqrt{k}}$. By the triangle inequality, $||A - A_{\geqslant\sigma}||_F \leqslant ||A - A_k||_F + ||A_k - A_{\geqslant\sigma}||_F$. If $k \leqslant l$, it's easy to conclude that $||A_k - A_{\geqslant\sigma}||_F \leqslant ||A - A_k||_F \leqslant \epsilon||A||_F$. If $k > l$, $||A_k - A_{\geqslant\sigma}||_F^2 = \sum_{i=l+1}^{k}\sigma_i^2 \leqslant k\sigma_{l+1}^2 \leqslant k\left(\frac{\epsilon||A||_F}{\sqrt{k}}\right)^2 \leqslant (\epsilon||A||_F)^2$. In either case, we have $||A - A_{\geqslant\sigma}||_F \leqslant 2\epsilon||A||_F$. □

Theorem 4.2. *Let Assumption 3 holds. For each $m = 0, \ldots, N-1$, assume $\epsilon^{(m)}$ satisfies $||\hat{T}^{(m)} - \hat{T}_k^{(m)}||_F \leqslant \epsilon^{(m)}||\hat{T}^{(m)}||_F$. Define $\tau_m = \frac{\epsilon^{(m)}||\hat{T}^{(m)}||_F}{\sqrt{k}}$. Algorithm 5 outputs the state $|\mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,:)\rangle$ corresponding to an approximation of $\mathcal{A}(i,:,:)$ such that there are at least $(1-\delta)N$ users, of which each user i satisfies*

$$||\mathcal{A}(i,:,:) - \mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F \leqslant \epsilon||\mathcal{A}(i,:,:)||_F \tag{4.29}$$

with probability at least $p_1 = 1 - e^{-\zeta^2\left(\frac{1}{p}-1\right)\frac{||\mathcal{A}||_F^2}{3N(1+\gamma)}}$, where $\delta, \gamma, \zeta \in (0,1)$. The precision ϵ in (4.29) is $\epsilon = \frac{10(1+\zeta)(1-p)\delta + 11\epsilon_0(1+\gamma)}{10\delta p}$, where p is the subsampling probability and $\epsilon_0 = 2\max_m \epsilon^{(m)}$. Moreover, let t be chosen uniformly from 0 to $N-1$. The probability

that sampling according to $\mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,t)$ (equivalent to measuring the state $|\mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,t)\rangle$ in the computational basis) provides a bad recommendation is

$$\Pr_{t\sim\mathcal{U}_N, j\sim\mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,t)}[(i,j,t)\text{bad}] \leqslant \left(\frac{\epsilon}{1-\epsilon}\right)^2 \tag{4.30}$$

Proof. Based on Lemma 4.3 in the main text, if the best rank-k approximation satisfies $||\hat{T}^{(m)} - \hat{T}_k^{(m)}||_F \leqslant \epsilon^{(m)}||\hat{T}^{(m)}||_F$, then

$$||\hat{T}^{(m)} - \hat{T}^{(m)}_{\geqslant\tau_m}||_F \leqslant 2\epsilon^{(m)}||\hat{T}^{(m)}||_F \leqslant \epsilon_0||\hat{T}^{(m)}||_F \tag{4.31}$$

for $m = 0, \cdots, N-1$. By summing both sides of (4.31) over m, we get

$$||\hat{\mathcal{T}} - \hat{\mathcal{T}}_{\geqslant\boldsymbol{\tau}}||_F^2 = \sum_{m=0}^{N-1} ||\hat{T}^{(m)} - \hat{T}^{(m)}_{\geqslant\tau_m}||_F^2 \leqslant \epsilon_0^2||\hat{\mathcal{T}}||_F^2. \tag{4.32}$$

Since the inverse QFT along the third mode of the tensor $\mathcal{T}$ cannot change the Frobenius norm of its horizontal slice, (4.32) can be be re-written as

$$||\mathcal{T} - \mathcal{T}_{\geqslant\boldsymbol{\tau}}||_F^2 \leqslant \epsilon_0^2||\mathcal{T}||_F^2. \tag{4.33}$$

Moreover, noticing that $||\mathcal{T} - \mathcal{T}_{\geqslant\boldsymbol{\tau}}||_F^2 = \sum_{i=0}^{N-1} ||\mathcal{T}(i,:,:) - \mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F^2$, we have $\mathrm{E}\left(||\mathcal{T}(i,:,:) - \mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F^2\right) \leqslant \frac{\epsilon_0^2||\mathcal{T}||_F^2}{N}$. Due to Markov's Inequality ([68, Proposition 2.6]), for $\delta \in (0,1)$,

$$\Pr\left(||\mathcal{T}(i,:,:) - \mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F^2 > \frac{\epsilon_0^2||\mathcal{T}||_F^2}{\delta N}\right) \leqslant \frac{\mathrm{E}\left(||\mathcal{T}(i,:,:) - \mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F^2\right)\delta N}{\epsilon_0^2||\mathcal{T}||_F^2} \leqslant \delta \tag{4.34}$$

holds. That means at least $(1-\delta)N$ users i satisfy

$$||\mathcal{T}(i,:,:) - \mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F^2 \leqslant \frac{\epsilon_0^2||\mathcal{T}||_F^2}{\delta N}. \tag{4.35}$$

Notice $\mathrm{E}\left(||\mathcal{T}||_F^2\right) = ||\mathcal{A}||_F^2/p$. Using the Chernoff bound, we have $\Pr\left(||\mathcal{T}||_F^2 > (1+\theta)||\mathcal{A}||_F^2/p\right) \leqslant e^{-\theta^2||\mathcal{A}||_F^2/3p}$ for $\theta \in [0,1]$, which is exponentially small. Here, we choose $\theta = 1/10$, then $||\mathcal{T}||_F^2 \leqslant 11||\mathcal{A}||_F^2/10p$.

Based on the second assumption in Assumption 3, we sum both sides of (4.2) for m and i respectively, obtaining

$$\frac{1}{1+\gamma}\frac{||\mathcal{A}||_F^2}{N} \leqslant ||\mathcal{A}(i,:,:)||_F^2 \leqslant (1+\gamma)\frac{||\mathcal{A}||_F^2}{N}, \tag{4.36}$$

and

$$\frac{1}{1+\gamma}\frac{||\mathcal{A}||_F^2}{N} \leqslant ||A^{(m)}||_F^2 \leqslant (1+\gamma)\frac{||\mathcal{A}||_F^2}{N}. \tag{4.37}$$

Then, (4.35) becomes

$$||\mathcal{T}(i,:,:) - \mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F^2 \leqslant \frac{11\epsilon_0^2(1+\gamma)}{10\delta p}||\mathcal{A}(i,:,:)||_F^2. \tag{4.38}$$

Meanwhile, since

$$\mathrm{E}\left(||\mathcal{A}(i,:,:) - \mathcal{T}(i,:,:)||_F^2\right) = \left(\frac{1}{p} - 1\right)||\mathcal{A}(i,:,:)||_F^2,$$

then

$$\Pr\left(||\mathcal{A}(i,:,:) - \mathcal{T}(i,:,:)||_F^2 > \nu||\mathcal{A}(i,:,:)||_F^2\right) \leqslant e^{-\zeta^2\left(\frac{1}{p}-1\right)\frac{||\mathcal{A}||_F^2}{3N(1+\gamma)}}, \tag{4.39}$$

where $\nu = (1+\zeta)\left(\frac{1}{p} - 1\right)$ and $\zeta \in (0,1)$. That means with probability at least p_1,

$$||\mathcal{A}(i,:,:) - \mathcal{T}(i,:,:)||_F^2 \leqslant \nu||\mathcal{A}(i,:,:)||_F^2. \tag{4.40}$$

Combining (4.38) and (4.40) together and by triangle inequality, we obtain

$$||\mathcal{A}(i,:,:) - \mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F \leqslant \epsilon||\mathcal{A}(i,:,:)||_F. \tag{4.41}$$

According to Lemma 4.2, the probability that sampling according to $\mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,:)$ provides a bad recommendation is

$$\Pr_{t\sim\mathcal{U}_N, j\sim\mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,t)}[(i,j,t)\text{bad}] \leqslant \left(\frac{\epsilon}{1-\epsilon}\right)^2. \tag{4.42}$$

□

It should be noted that by taking reasonable values for the parameters δ, γ, ζ, p, our algorithm can produce good recommendations with high probability, in order words, the probability p_1 is close to 1 and the precision ϵ could be comparatively small.

4.3.4 Complexity analysis

The complexity of Algorithm 5 is given by the following result.

Theorem 4.3. *With notation given in Theorem 4.2, for at least* $(1-\delta)N$ *users, Algorithm 5 outputs an approximation state of* $|\mathcal{A}(i,:,:)\rangle$ *with time complexity* $\mathcal{O}\left(\frac{\sqrt{k}N\text{polylog}(N)(1+\gamma)}{\min_m \epsilon^{(m)}(1+\epsilon)\sqrt{p}}\right)$. *For suitably chosen values of parameters* δ, ζ, γ, p *and* $\epsilon^{(m)}$, *the running time of Algorithm 5 is* $\mathcal{O}\left(\sqrt{k}N\text{polylog}(N)\right)$.

Proof. According to Theorem 4.1, the modified QSVE algorithm performed on the frontal slice $\hat{T}^{(m)}$ takes time $\mathcal{O}\left(N\text{polylog}(N)/\epsilon_{\text{SVE}}^{(m)}\right)$.

In Step 5 of Algorithm 5, we need to repeat the measurement $\mathcal{O}\left(\frac{||\mathcal{T}(i,:,:)||_F}{||\hat{\mathcal{T}}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F}\right)$ times in order to ensure the probability of getting the outcome $|0\rangle$ in this step is close to 1. For most users, we can prove that $\frac{||\mathcal{T}(i,:,:)||_F}{||\hat{\mathcal{T}}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F}$ is bounded and the upper bound is a constant for appropriate parameters. The proof is in the following.

Since $\text{E}\left(||\mathcal{T}(i,:,:)||_F^2\right) = \frac{||\mathcal{T}(i,:,:)||_F^2}{p} \leqslant (1+\gamma)\frac{||\mathcal{A}||_F^2}{pN}$, then by Chernoff bound,

$$||\mathcal{T}(i,:,:)||_F^2 \leqslant \frac{2(1+\gamma)||\mathcal{A}||_F^2}{pN} \tag{4.43}$$

holds with probability close to 1. Moreover, by Theorem 4.2, there are at least $(1-\delta)N$ users satisfying $||\mathcal{A}(i,:,:) - \mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F \leqslant \epsilon||\mathcal{A}(i,:,:)||_F$, then $(1+\epsilon)||\mathcal{A}(i,:,:)||_F \leqslant ||\mathcal{T}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F \leqslant (1+\epsilon)||\mathcal{A}(i,:,:)||_F$. Since the Frobenius norm is unchanged under the Fourier transform, we get

$$(1+\epsilon)||\hat{T}_{(i)}||_F \leqslant ||\hat{\mathcal{T}}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F \leqslant (1+\epsilon)||\hat{T}_{(i)}||_F. \tag{4.44}$$

Therefore,

$$||\hat{\mathcal{T}}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F^2 \geqslant (1+\epsilon)^2||\hat{T}_{(i)}||_F^2 \geqslant \frac{(1+\epsilon)^2}{1+\gamma}\frac{||\mathcal{A}||_F^2}{N}. \tag{4.45}$$

Combining (4.43) and (4.45) together, we can conclude that for at least $(1-\delta)N$ users, $\frac{||\mathcal{T}(i,:,:)||_F}{||\hat{\mathcal{T}}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F}$ is bounded, that is,

$$\frac{||\mathcal{T}(i,:,:)||_F}{||\hat{\mathcal{T}}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F} \leqslant \left(\frac{(1+\gamma)\frac{2||\mathcal{A}||_F^2}{pN}}{\frac{(1+\epsilon)^2}{1+\gamma}\frac{||\mathcal{A}||_F^2}{N}} \right)^{1/2} = \frac{\sqrt{2}(1+\gamma)}{(1+\epsilon)\sqrt{p}}. \tag{4.46}$$

The precision for the singular value estimation algorithm on the matrix $||\hat{T}^{(m)}||_F$ can be chosen as $\epsilon_{\text{SVE}}^{(m)} = \frac{\tau_m}{||\hat{T}^{(m)}||_F}$. Therefore, the total time complexity of Algorithm 5 is

$$\begin{aligned}
&(\log N)^4 \cdot \frac{N\text{polylog}(N)}{\min\limits_m \epsilon_{\text{SVE}}^{(m)}} \cdot \frac{||\mathcal{T}(i,:,:)||_F}{||\hat{\mathcal{T}}_{\geqslant\boldsymbol{\tau}}(i,:,:)||_F} \\
\leqslant &(\log N)^4 N\text{polylog}(N) \max_m \frac{||\hat{T}^{(m)}||_F}{\tau_m} \cdot \frac{\sqrt{2}(1+\gamma)}{(1+\epsilon)\sqrt{p}} \\
\approxeq &\frac{\sqrt{k}N\text{polylog}(N)(1+\gamma)}{\min\limits_m \epsilon^{(m)}(1+\epsilon)\sqrt{p}},
\end{aligned}$$

where $\epsilon^{(m)}$ and ϵ are defined in Theorem 4.2. □

4.4 The quantum cost

Defining the cost of quantum circuits is not an easy task due to the fact that each quantum computer model may have a different cost for a given quantum gate. Here, the quantum cost of a reversible gate is defined to be the number of 1×1 and 2×2 reversible gates or quantum logic gates required in its design. The quantum costs of all reversible 1×1 and 2×2 gates are taken as unity [4, 70, 82, 83]. The cost of a circuit is calculated by summing up the costs of the gates composing the circuit. First, we analysize the quantum cost of the QSVE circuit, then we analyze the cost of each gate in FIG 4.1.

4.4.1 The cost of QSVE

The QSVE algorithm, originally proposed in [33], is introduced in the preliminary part of the manuscript. The QSVE algorithm and its circuit are respectively given in Algorithm 6 and FIG. 4.2 below. In what follows, we focus on the quantum cost of this algorithm.

Let $A \in \mathbb{R}^{N\times N}$ be a matrix with singular value decomposition $A = \sum_i \sigma_i u_i v_i^T$ stored in the binary tree data structure. In Step 2 of Algorithm 6, the unitary U_Q corresponding to Q is defined as

$$U_Q : |0\rangle |j\rangle \to |\widetilde{A}, j\rangle = \frac{1}{\|A\|_F} \sum_{i\in[N]} \|A_i\| \, |i, j\rangle \tag{4.47}$$

for $j = 0, \cdots, N-1$, where $\widetilde{A}$ is a vector whose i-th row is $\widetilde{A}_i = \|A_i\|$ for $i \in [N]$. Since $\widetilde{A}$ is stored in a classical binary tree with depth $\lceil \log N \rceil$, U_Q can be implemented by performing $\lceil \log N \rceil$ controlled rotations on $|0\rangle^{\otimes \lceil \log N \rceil}$ (see [33, Appendix A]). To implement each controlled rotation $\sum_{\tilde{\theta}\in\{0,1\}^{\otimes \lceil \log N \rceil}} |\widetilde{\theta}\rangle\langle\widetilde{\theta}| \otimes e^{-\mathrm{i}Y\tilde{\theta}}$ with Pauli Y matrix $Y = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix}$, we use one rotation controlled on each qubit of the first register which can be implemented with cost $\mathcal{O}(\lceil \log N \rceil)$ (see [86, Lemma 2]). Therefore, the cost of implementing U_Q in Step 2 is $\sum_{\lceil \log N \rceil} \lceil \log N \rceil^2$, i.e., $\mathcal{O}(\lceil \log N \rceil^3)$.

Algorithm 6 Quantum singular value estimation

Input: $A \in \mathbb{R}^{N\times N}, x \in \mathbb{R}^N$ in the data structure in Lemma 2.1, precision parameter $\epsilon > 0$.

1: Create $|x\rangle = \sum_i \alpha_i |v_i\rangle$
2: Append a first register $\left|0^{\lceil \log m \rceil}\right\rangle$ and create the state $|Qx\rangle = \sum_i \alpha_i |Qv_i\rangle$ as in (4.47).
3: Perform phase estimation with precision parameter $2\epsilon > 0$ on the input $|Qx\rangle$ for the unitary W and obtain $\sum_i \alpha_i \left|Qv_i, \overline{\theta_i}\right\rangle$.
4: Compute $\overline{\sigma_i} = \cos\left(\overline{\theta_i}/2\right) \|A\|_F$ where $\overline{\theta_i}$ is the estimate from phase estimation, and uncompute the output of the phase estimation.
5: Apply the inverse of the transformation in Step 2 to obtain $\sum_i \alpha_i |v_i\rangle |\overline{\sigma_i}\rangle$.

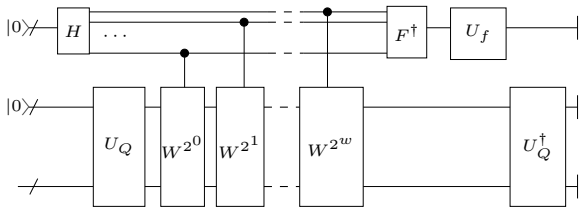


Figure 4.2: The circuit of QSVE algorithm. U_f is a unitary operator implemented through oracle with a computable function $f(x) = ||A||_F \cos(x/2)$. $w = \log N - 1$.

In Step 3, we can prove that the quantum cost of implementing the unitary W is $\mathcal{O}(N\lceil\log N\rceil^3)$. To be specific, notice that $W = U \cdot V$, where $V = 2QQ^T - I_{N^2}$, $U = 2PP^T - I_{N^2} = U_P\left(2\sum_i |i\rangle|0\rangle\langle i|\langle 0| - I_{N^2}\right)U_P^\dagger$, and $U_P : |i\rangle|0\rangle \to |i, A_i\rangle$ for $i \in [N]$. The unitary operator $2\sum_i |i\rangle|0\rangle\langle i|\langle 0| - I_{N^2}$ can be realized by the circuit shown in FIG. 4.3. Since each row of the matrix A is stored in the classical binary tree structure, U_P can be implemented with cost $\mathcal{O}(N\lceil\log N\rceil^3)$, as compared to the cost $\mathcal{O}(\lceil\log N\rceil^3)$ of implementing U_Q. To summarize, a total cost of $\mathcal{O}(N\lceil\log N\rceil^3)$ is required to implement W if we ignore less significant cost. Next, we perform phase estimation on W. When the singular values of W is precise to the $\lceil\log N\rceil$-th bit, the total number of rotation gate invocations is $\mathcal{O}\left(\lceil\log N\rceil^2\right)$ (See [7, Section 5.2]). The sequence of controlled-W^{2^j}, $j = 0, \cdots, \lceil\log N\rceil - 1$ operations in the phase estimation procedure can be implemented using $\mathcal{O}(\lceil\log N\rceil^3)$ gates by the modular exponentiation technique (see [52, Box 5.2]). Thus, the cost of Step 3 is $\mathcal{O}(N\lceil\log N\rceil^3)$, which dominates the cost of QSVE algorithm.

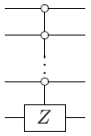


Figure 4.3: Quantum circuit to run $2\sum_i |i\rangle^{\otimes n}|0\rangle\langle i|^{\otimes n}\langle 0| - I^{\otimes(n+1)}$ for the input state with $n+1$ qubits. The block Z is the Pauli Z gate.

To summarize, the quantum cost of QSVE is $\mathcal{O}(N\lceil\log N\rceil^3)$.

4.4.2 The cost of Algorithm 5

In this section, we will analyze the quantum cost of Algorithm 5 whose circuit is given in FIG. 4.1. As discussed in Section 2.1.1 in Chapter 2, the circuit of QFT in Step 1 of Algorithm 2 is composed of $\mathcal{O}(\lceil \log N \rceil^2)$ H gates, CNOT gates, and 2-qubit controlled phase gates.

In Step 2, we give a rough estimation of the cost of the controlled-unitary operator U defined in (4.6). Current standard methods for realizing controlled-unitary gates rely on the decomposition of $U_{\mathrm{SVE}}^{(m)}$ into a set of 1×1 and 2×2 reversible gates. Specifically, based on the method proposed in [96], the number of additional operations required to add a control to each $U_{\mathrm{SVE}}^{(m)}$ will generally be far less than the cost of the constructed $U_{\mathrm{SVE}}^{(m)}$, so we next focus on estimating the cost of achieving $U_{\mathrm{SVE}}^{(m)}, m = 0, \cdots, N-1$. The most costy step of this process lies in constructing the operator $U_{\hat{P}_m}$ in (4.13), $m = 0, \cdots, N-1$. According to the analysis in the paragraph above (4.14), implementing all U_{P_k} in (4.10) for $k = 0, \cdots, N-1$ occupies the major cost of achieving $U_{\hat{P}_m}, m = 0, \cdots, N-1$, which takes a total of $\mathcal{O}(N^2 \lceil \log N \rceil^3)$ 1×1 and 2×2 reversible gates, since the cost of each U_{P_k} is $\mathcal{O}(N \lceil \log N \rceil^3)$ based on the cost of U_P analyzed in Appendix 4.4.1. To summarize, the quantum cost of implementing the operator U is $\mathcal{O}(N^2 \lceil \log N \rceil^3)$.

In conclusion, the quantum cost of implementing Algorithm 5 is $\mathcal{O}(N^2 \lceil \log N \rceil^3)$, which is mainly concentrated in achieving the operator U if we ignore the insignificant cost, such as the cost of implementing the operator V in (4.10). Actually, it is reasonable that the cost of Algorithm 5 is N times more expensive than that of the QSVE algorithm proposed in [33]. The cost of the QSVE algorithm on matrix $A \in \mathbb{C}^{N \times N}$ mainly concentrates on the cost of quantum access to classical data with N rows of A stored in N binary trees. By contrast, our algorithm deals with tensor $\mathcal{T} \in \mathbb{R}^{N \times N \times N}$ whose N^2 tubes are assumed to be stored in N^2 binary trees, so our

algorithm needs N times more gates than the QSVE algorithm to quantum access to the data structure.

Since implementing the physical operations depends on many factors, such as the initial quantum circuit, the quantum computer to be performed on, or the Hamiltonian of the system, the cost presented here is not necessarily the true cost but it provides a reference value. Also, it is not necessarily minimal because it can be decreased by finding more efficient quantum circuits.

The number of 1×1 and 2×2 reversible gates required by our quantum algorithm scales polynomially with the dimension of the preference tensor, which presents a big obstacle to practical realization. In fact, not only our algorithm and the QSVE algorithm, but also some benchmark algorithms, such as the HHL algorithm [23], appear too expensive to be executed efficiently by a quantum computer and are not likely to be feasible in the Noisy Intermediate-Scale Quantum (NISQ) era; see [56, Sections 6.6 and 6.7], [73, Section 1.10]. Finally, we point out most quantum machine learning algorithms focus on the time complexity rather than their quantum cost; see, e.g., [23, 63, 32, 8, 64].

4.5 Simulations

In this section, we numerically validate our model with recommendation systems tasks. As we do not have a large-scale quantum computer, to validate our quantum algorithm, we investigate the performance of its classical counterpart, namely, truncated t-svd [34], on real datasets in the classical computer. Due to the closeness, the testing results should be true for a fault-tolerant quantum computer.

4.5.1 Experimental setting

We choose three multiverse recommendation systems algorithms that are based on tensor decompositions: T-HOSVD [37], TT decomposition [53], a Collaborative

Filtering method (TF & SGD) [31], and compare them to truncated t-svd. All the experiments are performed under Windows 10, Python 3.7 and MATLAB R2016b running on a desktop (Intel Core i7 @ 3.60 GHz, 32.0G RAM). In each experiment we repeat 10 times and average the results.

In order to evaluate the performance of different methods, we compare the relative square error (RSE) defined in dB, mean absolute error (MAE), and root mean square error (RMSE), which are defined as follows

$$\text{RSE} = 20\log_{10}(\|\mathcal{T}_{\geqslant\tau} - \mathcal{A}\|_F / \|\mathcal{A}\|_F), \tag{4.48}$$

$$\text{MAE} = \frac{1}{K}\sum_{i,j,k} \mathcal{D}_{ijk}\left|(\mathcal{T}_{\geqslant\tau})_{ijk} - \mathcal{A}_{ijk}\right|, \tag{4.49}$$

$$\text{RMSE} = \|\mathcal{T}_{\geqslant\tau} - \mathcal{A}\|_F / \sqrt{K}, \tag{4.50}$$

where K is the total number of observed ratings, $\mathcal{D} \in \{0,1\}^{N_1\times N_2\times N_3}$ is a binary tensor with nonzero entries D_{ijk} whenever $\mathcal{A}_{ijk}$ is observed. These three measures are widely used in the recommendation systems literature.

4.5.2 Data

We test these four algorithms on two real datasets: Yahoo! Webscope Movies and Movielens, which are described as follows:

- For Yahoo! Webscope Movies dataset[1] with 7642 users, 11915 movies and 221K ratings in a $\{1,\cdots,5\}$ scale, we select the first 800 users and 4623 corresponding movies. Besides, the original Yahoo! Webscope Movies dataset contains user age and gender features. We choose user's year of birth as the third dimension and consider it as the context variable. Therefore, the size of resulting tensor is $800 \times 4623 \times 51$ with 23782 nonzero entries.

[1] Webscope v1.0, http://research.yahoo.com/

- Movielens[2] is the benchmark dataset in recommendation systems. Here we choose the "Movielens-latest-small" dataset, which has 610 users, 9742 movies and 100K ratings with a timestamp. For preprocessing the data, we divide the timestamps that users give ratings into 60 timeslots, obtaining a third-order tensor $\mathcal{A} \in \mathbb{R}^{610\times 9742\times 60}$ in which the three modes represents users, movies, and time respectively.

4.5.3 Comparison result

In FIGs. 4.4 and 4.5, the RSE, MAE, and RMSE values obtained by different methods are plotted for Yahoo! Webscope Movies and Movielens datasets respectively with 80% sampling probability. We first compare truncated t-svd, T-HOSVD, and TT methods since they do not require optimization and they all follow similar procedures. T-HOSVD is composed of a core tensor and unitary matrices storing the principal compoments of each mode. TT decomposition represents a tensor as the link structure of each core tensor. The truncation rank is (k, k, k) for k ranging from 5 to 50. Here, (k, k, k) refers to the multi-rank of t-svd, the multilinear rank of T-HOSVD, and the TT rank of TT decomposition. We can observe that the RSE values obtained by truncated t-svd are much lower than those obtained by T-HOSVD and TT decompositions. When k is greater than 20 (for Yahoo!) or 8 (for Movielens), the RMSE and MAE values of truncated t-svd are also lower than T-HOSVD and TT.

Moreover, to validate the effectiveness of our algorithm, we also evaluate the average probability of providing a bad recommendation, denoted as Pr[bad], when we sample according to $\mathcal{T}_{\geqslant\tau}$. Here, we calculate Pr[bad] by

$$\Pr[\text{bad}] = \frac{1}{N}\sum_{i=0}^{N-1}\left(\frac{\epsilon_i}{1-\epsilon_i}\right)^2, \tag{4.51}$$

[2] grouplens.org/datasets/movielens

where $\epsilon_i = \|\mathcal{T}_{\geqslant\tau}(i,:,:) - \mathcal{A}(i,:,:)\|_F / \|\mathcal{A}(i,:,:)\|_F$ and N refers to the number of users in the dataset. Our theoretical basis is Lemma 4.2 and Theorem 4.2. The curves of Pr[bad] are plotted in the bottom right of FIGs. 4.4 and 4.5. In summary, truncated t-svd has better performance than T-HOSVD and TT when applied to context-aware recommendation systems datasets.

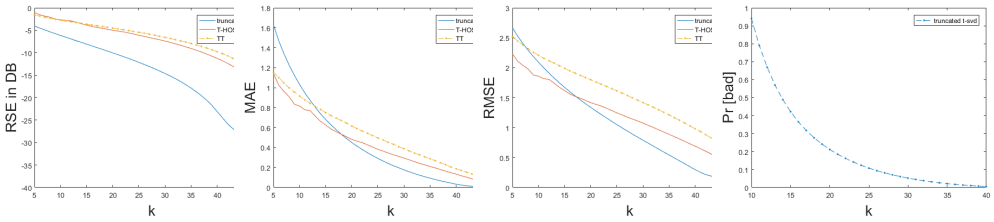


Figure 4.4: The comparison results on Yahoo! Webscope Movies dataset. The top-left, top-right and bottom-left figures plot the RSE, MAE, RMSE against truncation rank k respectively. The bottom-right figure shows that the average probability of providing a bad recommendation.

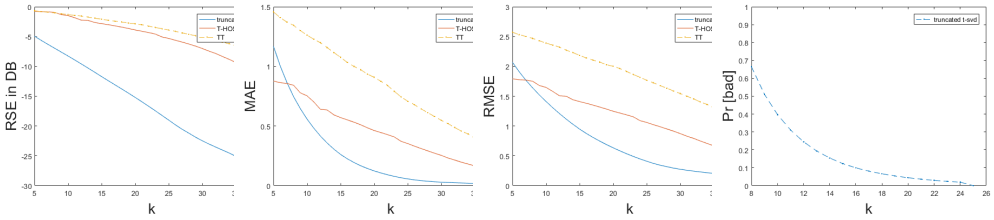


Figure 4.5: The comparison results on Movielens dataset.

In [31], the authors introduce a Collaborative Filtering method based on Tensor Factorization (TF & SGD) in Multiverse Recommendation model. To train this model, they minimize the regularized risk function by stochastic gradient descent (SGD). Due to the use of SGD algorithm, the computational cost is increased and the algorithm can be sensitive to initial guess. We compare the MAE values and average running time of truncated t-svd and TF & SGD on Yahoo! Webscope Movies dataset; see FIG. 4.6. For TF & SGD, the regularization parameters for factor matrices and

core tensor are all set as 0.1, and their initial learning rates are all set as 0.001. The multilinear rank is set as $d_U = d_M = d_C = k$. We can observe that the MAE values of TF & SGD is comparatively stable with k. When $k \geqslant 14$, the truncated t-svd performs better than TF & SGD in MAE. Moreover, the average running time of truncated t-svd is only around one fifth to one third of the time of TF & SGD.

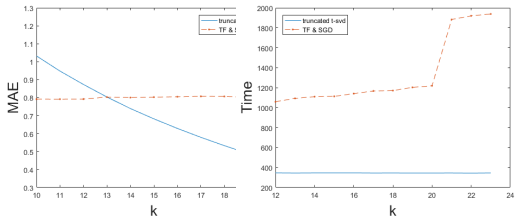


Figure 4.6: The MAE and average running time of truncated t-svd and TF & SGD algorithms on Yahoo! Webscope Movies dataset.

4.6 Relations with Tang's algorithms

In this section, we compare our 3D quantum algorithm with Tang's 2D quantum-inspired recommendation systems algorithm. For a 2D recommendation system modeled by an $N \times N$ matrix, Tang's quantum-inspired algorithm [79] was shown to have complexity $\mathcal{O}(\text{poly}(k)\text{polylog}(N))$, an exponential speedup compared to other classical methods, only polynomially slower than Kerenidis and Prakash's quantum algorithm [33]. This raised the question of whether Tang's algorithm was actually useful in practice. In order to study its practical performance, Lloyd et al. investigated Tang's 2D recommendation systems algorithm [2]. They commented that Tang's algorithm is only advantageous for preference matrix with extremely large dimension, very low rank and low condition number. However, it remains unclear whether such kind of datasets actually exists in practice. They also found that Tang's algorithm takes more time and suffers higher inaccuracies than the classical exact diagonalization method. Kerenidis and Prakash [32] also commented that this high

dependence on the rank and other parameters makes Tang's algorithm impractical. Therefore, it is unlikely that Tang's algorithm can replace Kerenidis and Prakash's algorithm in practical 2D recommendation systems.

Moreover, both Tang and Kerenidis and Prakash's algorithms are for 2D recommendation systems, and our proposed quantum algorithm, Algorithm 5, is for 3D recommendation systems. To our best knowledge, there is no previous work on quantum-inspired classical algorithm of 3D recommendation systems. Clearly, this is a very interesting and practically important problem. Hence, here we would like to share our understandings on how to extend Tang's quantum-inspired techniques to third-order recommendation systems.

To our understanding, there are at least two difficulties in extending quantum-inspired classical algorithms from 2D to 3D recommendation systems. First, as the product that a user prefers in a certain context is very likely to affect the recommendation for him/her at other contexts, the relations among different frontal slices (a matrix of users $\times$ products) of the preference tensor should be taken into account. Our quantum algorithm addresses this problem very well because the QFT is performed to combine a user's preferences in different contexts. On the other hand, if we extend Tang' idea to third-order recommendation systems, the immediate idea is to apply Tang's 2D recommendation systems algorithm to every frontal slice of the preference tensor, but this idea fails to consider the preference correlations among different contexts for a certain user. In order to take this factor into consideration, the rough idea is to apply DFT to each horizontal slice of the preference tensor in order to achieve the same effect as QFT. It is easy to see that the complexity of this step is $\mathcal{O}(N^3\log(N))$. If we choose any unitary transformation to substitute DFT, which is adopted in the transformed t-svd [75], the complexity will be $\mathcal{O}(N^{3.3})$. In either case, the complexity of this step alone exceeds that of our quantum algorithm for 3D recommendation systems. Therefore, how to control the complexity of

quantum-inspired classical algorithms for 3D recommendation systems is the first obstacle we need to overcome. Another difficulty is how to guarantee that the designed quantum-inspired algorithm is useful in practice, or suitable for real datasets.

4.7 Conclusion

In this chapter, we proposed the first quantum algorithm for context-aware recommendation systems modeled by third-order tensors. Moreover, we showed that this quantum algorithm can provide good recommendations varying with contexts and run in expected time $\mathcal{O}(\sqrt{k}N\text{polylog}(N))$ for some suitable parameters. The numerical simulation validates our algorithm.

Chapter 5

A Monte-Carlo algorithm for low tubal-rank tensor approximations

5.1 Introduction

The low-rank randomized tensor approximations algorithms plays an crucial role in many applications, such as machine learning, clustering, combinatorial optimization, etc. [29]. In the next two sections, we introduce some related works on randomized algorithms of low-rank matrix and tensor approximation respectively.

5.1.1 Randomized algorithms of low-rank matrix approximation

In this section, we give a outline of the low-rank matrix approximations randomized algorithms. The goal of computing a low-rank matrix approximation is to find a low-dimensional subspace which has most of the properties of the given matrix. Generally, the problem of low-rank matrix approximation falls into two categories: fixed-rank approximation problems and fixed-precision approximation problems. In the fixed-rank problem, the desired rank k (always together with the oversampling parameter p) is given in advance, and our task is to find a matrix Q with $k + p$ orthonormal columns satisfying

$$\|A - QQ^*A\| \approx \min_{\text{rank}(X)\leqslant k} \|A - X\|. \tag{5.1}$$

Fixed-precision problem, on the other hand, is to find Q with rank as small as possible such that

$$||A - QQ^*A|| \leqslant \epsilon \tag{5.2}$$

holds, for a given target precision parameter ϵ.

The process of low-rank randomized matrix approximations algorithms can always be divided into two stages. First, we compute the sampling probabilities, and compress the input matrix into the sampled subspace. This sampled matrix is then operated to compute the standard decompositions in the second stage. The most common approximation schemes include, but not limited to, sparsification, column selection methods, approximation by dimension reduction and submatrices. Sparsification is the simplest approach to matrix approximation and the subsampling method adopted in our quantum context-aware recommendation systems algorithm is a type of sparsification method. The randomized algorithms using the column selection methods compute each column's sampling probability by either length-squared sampling or leverage scores. The length-squared sampling method is first proposed in [17] and it samples columns according to the distribution of Euclidean norm of the columns of the input matrix.

In [17], Frieze, Kannan and Vempala propose a Monte-Carlo randomized algorithm (FKV) for finding the low-rank approximations of a matrix. The FKV algorithm constructs a smaller matrix by first sampling r rows of $A \in \mathbb{R}^{m\times n}$ using the length-squared sampling technique, then rescaling each sampled row. Suppose the obtained matrix is R. We then sample and rescale c columns of R again to get the matrix C. The length-squared sampling guarantees that the entry of RR^T is an unbiased estimator of AA^T and the sum of all variance of the entry of RR^T is bounded by $\epsilon^2||A||_F^4$. It can be proved that the length-squared sampling minimizes the variance among all the unbiased estimators [30]. The space spanned by the top k

left singular vectors of R is a good approximation to A, thus the problem of low-rank approximation of A can be converted to computing the singular value decomposition of R instead, which reduces computation cost remarkably. Similar properties also holds for matrices R and C. Finally, we perform matrix SVD on this smaller matrix C in order to retrieve some of the singular information of A. To be more specific, the top-k singular values of C can be regarded as an approximation of the top-k singular values of A, and the right singular vectors of a matrix A can be approximated by $\tilde{\boldsymbol{v}}_l = R^T \boldsymbol{w}_l / \tilde{\sigma}_l$ while the left singular vector of A is given by $\tilde{\boldsymbol{u}}_l = A\left(R^T \boldsymbol{w}_l / \tilde{\sigma}_l\right)$, where $\tilde{\sigma}_l$, $\boldsymbol{w}_l$ are the singular value and left singular vectors of C.

The only requirement for the sampled number r and c is that they have to be greater than the rank of A. Actually, they determine the quality of the approximation. They should be larger enough to capture the most features of A and much smaller than the size of A so that the complexity of computing SVD of C is far less than for the original matrix A. In [17], the authors show that the FKV algorithm can provide an approximate SVD with error ϵ by setting $r = c = \mathcal{O}(\max k^4/\epsilon^2, k^2/\epsilon^4)$.

5.1.2 Related works on randomized tensor factorization and approximation

For matrices, there are polynomial time algorithms in finding optimal approximations, but this is not the case for high-order tensors. However, despite finding the best low cp-rank approximation for tensors is NP-hard, there exist randomized algorithms utilizing length-squared sampling to obtain a good approximation of a tensor with high probability in polynomial time, where the approximated tensor is expressed as the sum of k rank-1 tensors.

In [90], Jiani Zhang proposes a randomized t-svd method which can produce a factorization with similar properties to the t-svd. The computational complexity of computing t-svd is reduced from $\mathcal{O}(kN^4)$ to $\mathcal{O}(kN^3 + N^3 \log N)$.

Also in [12], the authors propose a randomized tensor approximation algorithm based on HOSVD. This method first unfolds the tensor along each mode, then chooses columns randomly from these unfolding matrices and projects the tensor along that mode onto the space spanned by the sampled columns adaptively. The similar algorithm also found in [47], in which the full SVD of each mode unfolding of HOSVD is replaced with the matrix randomized SVD algorithm to obtain the factor matrix. In [47], the authors propose an adaptive tensor approximation which adaptively finds a low-rank representation satisfying a given tolerance. In [44], the authors propose a tensor-CUR algorithm which computes an approximation tensor in the form of linear combination of subtensors of the original data tensor. It applies the matrix-CUR to one of the unfolding matrix modes (called the "distinguished" mode) and provides an approximation based on those few tubes and frontal slices.

Compared with other tensor decompositions, t-svd has been shown to be superior in capturing the spatial-shifting correlation [75] that is ubiquitous in real-world data, so it is suitable for tensors with strong relations between slices. Inspired by the t-svd and randomized algorithms discussed above, we combine length-squared sampling with truncated t-svd method, obtaining a good low tubal-rank tensor approximation. Our algorithm extends the fast Monte-Carlo methods for low-rank matrix approximations to third-order tensors and approximates the original tensor in Fourier domain. Similar to the idea of FKV, we select a random subset of data instead of entire data, then perform the corresponding operations on this small sampled tensor. As to the sampling method, it may be unreasonable to choose slices at random uniformly especially for a tensor of which only a few slices are significant. Therefore, we adopt the length-squared sampling method, i.e., sample according to the distribution of Frobenius norm of slices, which leads to many provable error bounds.

The rest of this chapter is organized as follows. In Section 5.2, we present our

randomized low-rank approximation algorithm for third-order tensors; see Algorithm 7. Then in Section 5.3, a detailed theoretical error analysis of Algorithm 7 is presented. The complexity of Algorithm 7 is analysed in Section 5.4. In Section 5.5, we show the effectiveness and accuracy of Algorithm 7 on real dataset and then compare it with the random t-svd algorithm proposed in [90].

5.2 A Monte-Carlo algorithm for low tubal-rank tensor approximations

In this section, we present our random algorithm for computing low-rank approximation of third-order tensors $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, provided that tubes $\mathcal{A}(i, j, :), i \in [N_1], j \in [N_2]$ satisfy certain norm constraint.

Algorithm 7 A random Monte-Carlo method for finding low-rank approximation of third-order tensors

Input: tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, precision $\epsilon^{(m)}$, $m = 0, \cdots, N_3 - 1$, truncation number k.

Output: the approximated tensor $\tilde{\mathcal{A}}$.

1: Perform the DFT along the third mode of tensor $\mathcal{A}$, getting tensor $\hat{\mathcal{A}}$.

2: Sample r_1 horizontal slice indices $i_1, i_2, \cdots, i_{r_1}$ according to the norm square of horizontal slices $\frac{||\hat{\mathcal{A}}(i,:,:)||_F^2}{||\hat{\mathcal{A}}||_F^2}$, $i \in [N_1]$. Let $\mathcal{R}$ be the tensor whose r-th horizontal slice is $\frac{||\hat{\mathcal{A}}||_F}{\sqrt{r_1}} \frac{\hat{\mathcal{A}}(i_r,:,:)}{||\hat{\mathcal{A}}(i_r,:,:)||_F}$, $r \in [r_1]$.

3: Sample $s \in [r_1]$ uniformly, then sample a tube index j according to the distribution $\frac{||\mathcal{R}(s,j,:)||^2}{||\mathcal{R}(s,:,:)||_F^2} = \frac{||\hat{\mathcal{A}}(i_s,j,:)||^2}{||\hat{\mathcal{A}}(i_s,:,:)||_F^2}$. Repeat this procedure for r_2 times to obtain sampled indices $j_1, j_2, \cdots, j_{r_2}$. Define tensor $\mathcal{C}$ to be the tensor whose r-th lateral slice is $\frac{||\mathcal{R}||_F}{\sqrt{r_2}} \frac{\mathcal{R}(:,j_r,:)}{||\mathcal{R}(:,j_r,:)||_F}$, $r \in [r_2]$.

4: Query all elements of $\hat{\mathcal{A}}$ corresponding to the sampled indices of horizontal and lateral slices. For each $t \in [N_3]$, compute the left singular vectors of each frontal slice of $C^{(t)}$, i.e. $\boldsymbol{u}_1(C^{(t)}), \cdots, \boldsymbol{u}_k(C^{(t)})$ corresponding to singular values $\sigma_1(C^{(t)}), \cdots, \sigma_k(C^{(t)})$.

5: Compute $V^{(t)} = R^{(t)\dagger} U^{(t)} \Sigma^{(t)^{-1}}$ for each $t \in [N_3]$, where $U^{(t)} \in \mathbb{C}^{r_1 \times k}$ denotes the matrix whose i-th column is $\boldsymbol{u}_i(C^{(t)})$ and $\Sigma^{(t)} \in \mathbb{R}^{k \times k}$ is the diagonal matrix whose i-th diagonal entry is $\sigma_i(C^{(t)})$. Note $i \in T = \{i : \sigma_i^2(C^{(t)}) \geqslant \gamma ||C^{(t)}||_F^2\}$.

6: Let $\tilde{\hat{\mathcal{A}}}$ be the tensor whose t-th frontal slice is $\tilde{\hat{A}}^{(t)} \triangleq \hat{A}^{(t)} V^{(t)} V^{(t)\dagger}$, $t \in [N_3]$.

7: Perform the inverse FFT along the third mode of tensor $\tilde{\hat{\mathcal{A}}}$ to obtain the approximated tensor $\tilde{\mathcal{A}}$.

Remark 5.1. *The probability of getting the sampled index j in Step 3 of Algorithm 7 is $\frac{||\mathcal{R}(:,j,:)||_F^2}{||\mathcal{R}||_F^2}$. To be more specific,*

$$\mathrm{P}_j = \sum_{t=1}^{r_1} \frac{||\hat{\mathcal{A}}(i_t, j, :)||_2^2}{r_1 ||\hat{\mathcal{A}}(i_t, :, :)||_F^2} = \sum_{t=1}^{r_1} \frac{\mathcal{R}(t, j, :)||_2^2}{||\mathcal{A}||_F^2} = \frac{||\mathcal{R}(:, j, :)||_F^2}{||\mathcal{R}||_F^2}. \tag{5.3}$$

Thus the sampling method in Step 3 is similar to Step 2, i.e., sampling according to the distribution of slice norms.

Remark 5.2. *For the simplicity of analysis, we set the sampled number $r_1 = r_2 = \frac{10^7 k^2}{\epsilon^2}$, where k and ϵ are the preset rank and precision respectively.*

Our algorithm extends the fast Monte-Carlo methods for low-rank matrix ap-

proximations proposed by A. Frieze et al. (FKV) to third-order tensors. Our general idea is to approximate tensor $\mathcal{A}$ in Fourier domain. Specifically, after performing the Fourier transform along the third mode of $\mathcal{A}$, $\hat{\mathcal{A}}$ is obtained. It can be regarded as a matrix whose (i, j) entry is a tube $\hat{\mathcal{A}}(i, j, :)$, thus we can adopt a sampling approach similar to matrix FKV algorithm (Step 2 to Step 4). After Step 5, it can be proved that the column of matrix $V^{(t)}$ is an approximation of the right singular vector of $\hat{A}^{(t)}$, thus $\hat{A}^{(t)}V^{(t)}V^{(t)\dagger}$ represents projecting matrix $\hat{A}^{(t)}$ onto the space spanned by the columns of $V^{(t)}$. Let the best rank-k approximation of $\hat{A}^{(t)}$ computed by matrix SVD be $\hat{A}_k^{(t)}$. Then for each $\hat{A}^{(t)}$, we can prove that

$$\mathrm{E}\left(||\hat{A}^{(t)} - \hat{A}^{(t)}V^{(t)}V^{(t)\dagger}||_F^2\right) \leqslant ||\hat{A}^{(t)} - \hat{A}_k^{(t)}||_F^2 + \epsilon^{(t)}||\hat{A}^{(t)}||_F^2. \tag{5.4}$$

Thus, we have the result

$$\mathrm{E}\left(||\mathcal{A} - \mathcal{A} * \mathcal{V} * \mathcal{V}^\dagger||_F^2\right) \leqslant ||\mathcal{A} - \mathcal{A}_k||_F^2 + \epsilon||\mathcal{A}||_F^2, \tag{5.5}$$

where $\mathcal{V}$ denotes the tensor whose t-th frontal slice is $V^{(t)}$, and $||\mathcal{A} - \mathcal{A}_k||_F^2$ is the theoretical minimum error introduced in Lemma 4.1. Note that $\mathcal{A} * \mathcal{V}$ refers to the t-product between tensors $\mathcal{A}$ and $\mathcal{V}$.

Remark 5.3. *Our algorithm can also be used to approximate the singular value, singular vectors of a tensor based on t-svd, thus it can be seen as a fast Monte-Carlo algorithm for approximated tensor singular value decomposition. Specifically, the top-k singular values of $\mathcal{A}$, i.e., the top-k singular values of $\hat{A}^{(t)}$ for each $t \in [N_3]$, can be approximated by the truncated-k singular values of $\hat{C}^{(t)}$. Moreover, the top-k right singular vectors of $\hat{A}^{(t)}$ can be approximated by the columns of $V^{(t)}$ calculated in Step 5 of Algorithm 7, while the top-k approximated left singular vectors of $\hat{A}^{(t)}$ are given by*

$$\frac{\hat{A}^{(t)}V^{(t)}(:, l)}{\sigma_l(C^{(t)})}, \quad l \in [k]. \tag{5.6}$$

5.3 Theoretical Analysis

The following theorems assert that sampling a certain number of slices can provide a good approximation to the original tensor. That is, the sampled tensor $\mathcal{R}$ can be proved to be approximations to tensor $\mathcal{A}$ in the sense that $\mathcal{R}^\dagger * \mathcal{R}$ is close to $\mathcal{A}^\dagger * \mathcal{A}$.

Theorem 5.1. *Suppose $\mathcal{R}$ is formed by sampling horizontal slices of tensor $\hat{\mathcal{A}}$ described by Algorithm 7, then*

$$\mathrm{E}\left(R^{(t)\dagger}R^{(t)}\right) = \hat{A}^{(t)\dagger}\hat{A}^{(t)} \tag{5.7}$$

holds for $t \in [N_3]$.

Proof. The expectation of the (r,s)-th entry of $R^{(t)\dagger}R^{(t)}$ is the corresponding entry of $A^{(t)\dagger}A^{(t)}$.

$$\mathrm{E}\left(\left(R^{(t)\dagger}R^{(t)}\right)_{rs}\right) = \sum_{k=1}^{r_1} \mathrm{E}\left(\mathcal{R}(i_k,r,t)^*\mathcal{R}(i_k,s,t)\right) \tag{5.8}$$

$$= \sum_{k=1}^{r_1}\sum_{i=1}^{N_1} \frac{\hat{\mathcal{A}}(i,r,t)^*}{||\hat{\mathcal{A}}(i,:,:)||_F}\frac{||\hat{\mathcal{A}}||_F}{\sqrt{r_1}}\frac{\hat{\mathcal{A}}(i,s,t)}{||\hat{\mathcal{A}}(i,:,:)||_F}\frac{||\hat{\mathcal{A}}||_F}{\sqrt{r_1}}\frac{||\hat{\mathcal{A}}(i,:,:)||_F^2}{||\hat{\mathcal{A}}||_F^2} \tag{5.9}$$

$$= \sum_{k=1}^{r_1}\sum_{i=1}^{N_1} \frac{\hat{\mathcal{A}}(i,r,t)^*\hat{\mathcal{A}}(i,s,t)}{r_1} \tag{5.10}$$

$$= \left(\hat{A}^{(t)\dagger}\hat{A}^{(t)}\right)_{rs}. \tag{5.11}$$

Note A^* denotes the conjugate of A and $A^\dagger$ denotes the conjugate transpose of A. □

Assumption 4. *For $i \in [N_1]$, $t \in [N_3]$, suppose*

$$||\hat{\mathcal{A}}(i,:,t)||_2^2 = \frac{1}{\lambda_{i,t}N_3}||\hat{\mathcal{A}}(i,:,:)||_F^2. \tag{5.12}$$

Denote $\lambda_t' = \max_{i\in[N_1]}\lambda_{i,t}$, $\lambda_t = \min_{i\in[N_1]}\lambda_{i,t}$ and $\alpha_t \triangleq \frac{\lambda_t'}{\lambda_t}$.

Theorem 5.2. *For all $\theta > 0$,*

$$\Pr\left(||R^{(t)\dagger}R^{(t)} - \hat{A}^{(t)\dagger}\hat{A}^{(t)}||_F \geqslant \theta||\hat{A}^{(t)}||_F^2\right) \leqslant \frac{\alpha_t}{\theta^2 r_1}, \tag{5.13}$$

where α_t is introduced in Assumption 4.

Proof.

$$\mathrm{E}\left(||R^{(t)\dagger}R^{(t)} - \hat{A}^{(t)\dagger}\hat{A}^{(t)}||_F^2\right) = \sum_{r,s=1}^{N_2} \mathrm{E}\left(\left|\left(R^{(t)\dagger}R^{(t)} - \hat{A}^{(t)\dagger}\hat{A}^{(t)}\right)_{rs}\right|^2\right) \tag{5.14}$$

$$\leqslant \sum_{r,s=1}^{N_2} \mathrm{E}\left(\left|\left(R^{(t)\dagger}R^{(t)}\right)_{rs}\right|\right)^2 \tag{5.15}$$

$$= \sum_{r,s=1}^{N_2}\sum_{k=1}^{r_1} \mathrm{E}\left(|\mathcal{R}(i_k,r,t)^*\mathcal{R}(i_k,,s,t)|^2\right) \tag{5.16}$$

$$= \frac{1}{r_1}\sum_{i=1}^{N_1} \frac{\left(\sum_{r=1}^{N_2}\hat{\mathcal{A}}(i,r,t)^2\right)^2 ||\hat{\mathcal{A}}||_F^2}{||\hat{\mathcal{A}}(i,:,:)||_F^2} \tag{5.17}$$

$$= \frac{||\hat{\mathcal{A}}||_F^2}{r_1}\sum_{i=1}^{N_1} \frac{||\hat{\mathcal{A}}(i,:,t)||_2^4}{||\hat{\mathcal{A}}(i,:,:)||_F^2}. \tag{5.18}$$

According to Assumption 4,

$$||\hat{A}^{(t)}||_F^2 = \sum_{i=1}^{N_1}||\hat{\mathcal{A}}(i,:,t)||_2^2 = \sum_{i=1}^{N_1}\frac{||\hat{\mathcal{A}}(i,:,:)||_F^2}{\lambda_{i,t}N_3} \geqslant \frac{||\mathcal{A}||_F^2}{\lambda'_t N_3}, \tag{5.19}$$

thus

$$\mathrm{E}\left(||R^{(t)\dagger}R^{(t)} - \hat{A}^{(t)\dagger}\hat{A}^{(t)}||_F^2\right) = \frac{||\hat{\mathcal{A}}||_F^2}{r_1 N_3}\sum_{i=1}^{N_1}\frac{||\hat{\mathcal{A}}(i,:,t)||_2^2}{\lambda_{i,t}} \tag{5.20}$$

$$\leqslant \left(\frac{||\hat{\mathcal{A}}||_F^2}{N_3}\right)\frac{||\hat{A}^{(t)}||_F^2}{r_1\lambda_t} \tag{5.21}$$

$$\leqslant \frac{\alpha_t}{r_1}||\hat{A}^{(t)}||_F^4. \tag{5.22}$$

According to Markov inequality, for any $\theta > 0$,

$$\Pr\left(||R^{(t)\dagger}R^{(t)} - \hat{A}^{(t)\dagger}\hat{A}^{(t)}||_F \geqslant \theta||\hat{A}^{(t)}||_F^2\right) \leqslant \frac{\alpha_t}{\theta^2 r_1}. \tag{5.23}$$

□

Next, we introduce a notation for the convenience of the rest analysis. Denote

$$\Delta\left(M; \boldsymbol{x}_i, i \in I\right) = ||M||_F^2 - ||M - M\sum_{i\in I}\boldsymbol{x}_i\boldsymbol{x}_i^\dagger||_F^2 \tag{5.24}$$

for matrix M and vector $\boldsymbol{x}_i$. Specifically, if vectors $\boldsymbol{x}_i$ are orthogonormal, $\Delta\left(M; \boldsymbol{x}_i, i \in I\right)$ is the norm of the projection of M onto the subspace spanned by the $\boldsymbol{x}_i$.

In fact, Algorithm 7 needs that there exists a good low-rank approximation to each frontal slice of the tensor $\hat{A}^{(t)}$ in the subspace spanned by a part of its rows, and the cumulative error is related to the tubal-rank k and the sampled number r_1. We have the following theorem.

Theorem 5.3. *Given tensor $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}$ and tensor $\mathcal{R} \in \mathbb{R}^{r_1\times N_2\times N_3}$ generated by Algorithm 7. Then with probability at least $9/10$, there exists a set of orthonormal vectors $\boldsymbol{y}_1^{(t)}, \boldsymbol{y}_2^{(t)}, \cdots, \boldsymbol{y}_k^{(t)}$ in the row space of $R^{(t)}$ such that*

$$\Delta\left(\hat{A}^{(t)}; \boldsymbol{y}_j^{(t)}, j \in [k]\right) \geqslant \sum_{i=1}^{k}\hat{\sigma}_i^{(t)2} - \frac{10k\alpha_t}{r_1}||\hat{A}^{(t)}||_F^2, \tag{5.25}$$

where $\hat{\sigma}_i^{(t)}$ is the singular value of $\hat{A}^{(t)}$ with the corresponding left, right singular vectors $\hat{\boldsymbol{u}}_i^{(t)}$, $\hat{\boldsymbol{v}}_i^{(t)}$ respectively, $t \in [N_3]$.

Proof. Define

$$\boldsymbol{w}_j^{(t)} = \frac{||\mathcal{A}||_F^2}{r_1}\sum_{i=1}^{r_1}\frac{\left(\hat{\boldsymbol{u}}_j^{(t)}\right)_i^*}{||\hat{\mathcal{A}}(i,:,:)||_F^2}\hat{\mathcal{A}}(i,:,t). \tag{5.26}$$

Based on the sampling method of Step 2 in Algorithm 7, we have

$$\mathrm{E}\left(\boldsymbol{w}_j^{(t)}\right) = \frac{||\mathcal{A}||_F^2}{r_1}\sum_{i=1}^{r_1}\mathrm{E}\left(\frac{\left(\hat{\boldsymbol{u}}_j^{(t)}\right)_i^*}{||\hat{\mathcal{A}}(i,:,:)||_F^2}\hat{\mathcal{A}}(i,:,t)\right) \tag{5.27}$$

$$= \frac{||\mathcal{A}||_F^2}{r_1}\sum_{i=1}^{r_1}\sum_{k=1}^{N_1}\frac{\left(\hat{\boldsymbol{u}}_j^{(t)}\right)_k^*}{||\hat{\mathcal{A}}(k,:,:)||_F^2}\hat{\mathcal{A}}(k,:,t)\frac{||\hat{\mathcal{A}}(k,:,:)||_F^2}{||\mathcal{A}||_F^2} \tag{5.28}$$

$$= \frac{1}{r_1}\sum_{i=1}^{r_1}\hat{\boldsymbol{u}}_j^{(t)\dagger}\hat{A}^{(t)} = \sigma_j^{(t)}\hat{\boldsymbol{v}}_j^{(t)}. \tag{5.29}$$

Also, according to Assumption 4, we get

$$\mathrm{E}\left(\left|\boldsymbol{w}_j^{(t)} - \sigma_j^{(t)}\hat{\boldsymbol{v}}_j^{(t)}\right|^2\right) \leqslant \frac{||\mathcal{A}||_F^2}{r_1}\sum_{i=1}^{N_1}\frac{\left|\left(\hat{\boldsymbol{u}}_j^{(t)}\right)_i^*\right|^2||\hat{\mathcal{A}}(i,:,t)||^2}{||\hat{\mathcal{A}}(i,:,:)||_F^2} \tag{5.30}$$

$$= \frac{||\mathcal{A}||_F^2}{r_1}\sum_{i=1}^{N_1}\frac{\left|\left(\hat{\boldsymbol{u}}_j^{(t)}\right)_i^*\right|^2}{\lambda_{i,t}N_3} \tag{5.31}$$

$$\leqslant \frac{\alpha_t||\hat{A}^{(t)}||_F^2}{r_1}. \tag{5.32}$$

Let $\boldsymbol{y}_1^{(t)}, \boldsymbol{y}_2^{(t)}, \cdots, \boldsymbol{y}_l^{(t)}$ be an orthonormal basis of $\mathbb{C}^{N_2}$ such that they span the space spanned by $\frac{1}{\sigma_1^{(t)}}\boldsymbol{w}_1^{(t)}, \frac{1}{\sigma_2^{(t)}}\boldsymbol{w}_2^{(t)}\cdots, \frac{1}{\sigma_k^{(t)}}\boldsymbol{w}_k^{(t)}$, then

$$\mathrm{E}\left(||\hat{A}^{(t)} - \sum_{i=1}^{l}\hat{A}^{(t)}\boldsymbol{y}_i^{(t)}\boldsymbol{y}_i^{(t)\dagger}||_F^2\right) \leqslant \sum_{i=k+1}^{N_2}\hat{\sigma}_i^{(t)2} + \sum_{i=1}^{k}\mathrm{E}\left(\left|\boldsymbol{w}_j^{(t)} - \sigma_j^{(t)}\hat{\boldsymbol{v}}_j^{(t)}\right|^2\right) \tag{5.33}$$

$$\leqslant \sum_{i=k+1}^{N_2}\hat{\sigma}_i^{(t)2} + \frac{k\alpha_t}{r_1}||\hat{A}^{(t)}||_F^2. \tag{5.34}$$

The result of Theorem 5.3 can be obtained directly from Markov inequality. □

Before proving the main theorem (Theorem 5.4), we would like to introduce two lemmas which are first adopted in [17].

Lemma 5.1. *[17] Let A and S be matrices with the same number of columns satisfying*

$$||A^T A - S^T S||_F \leqslant \theta ||A||_F^2, \tag{5.35}$$

then for any set of unit vectors $\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_l, l \leqslant k$ in the row space of A,

$$|\Delta(A; \boldsymbol{z}_i, i \in [l]) - \Delta(S; \boldsymbol{z}_i, i \in [l])| \leqslant k^2 \theta ||A||_F^2. \tag{5.36}$$

Lemma 5.2. *[17]*

- $\Delta(R^{(t)}; \boldsymbol{v}_i^{(t)}, i \in T) \geqslant \Delta(R^{(t)\dagger}; \boldsymbol{u}_i^{(t)}, i \in T) - \frac{\epsilon}{8} ||\hat{A}^{(t)}||_F^2$
- $|\boldsymbol{v}_i^{(t)}|^2 \leqslant 1 + \epsilon/16$

Theorem 5.4. *Given a third-order tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, truncation number k, precision ϵ, and δ, Algorithm 7 outputs the approximated tensor $\tilde{\mathcal{A}} = \mathcal{A} * \mathcal{V} * \mathcal{V}^\dagger$ of rank at most k with probability at least $1 - \delta$ such that*

$$\mathrm{E}\left(||\mathcal{A} - \mathcal{A} * \mathcal{V} * \mathcal{V}^\dagger||_F^2\right) \leqslant \sum_{t=1}^{N_3} \sum_{i=k+1}^{\boldsymbol{r}(t)} \sigma_i^{(t)2} + \epsilon ||\mathcal{A}||_F^2, \tag{5.37}$$

where $\boldsymbol{r}$ is the multi-rank of $\mathcal{A}$.

Proof. First, we apply Theorem 5.2 twice to row and column samples respectively. Based on the analysis of Lemma 5.1, Step 3 of Algorithm 7 is equivalent to Step 2 except in the sampling direction. Therefore, with probability at least 9/10 and set $\theta = \frac{\epsilon}{k 10^{7/2}}$, we have

$$||R^{(t)\dagger} R^{(t)} - \hat{A}^{(t)\dagger} \hat{A}^{(t)}||_F \leqslant \theta ||\hat{A}^{(t)}||_F^2 \qquad \text{and} \qquad ||C^{(t)\dagger} C^{(t)} - R^{(t)\dagger} R^{(t)}||_F \leqslant \theta ||R^{(t)}||_F^2. \tag{5.38}$$

According to Theorem 5.3, with probability at least 9/10, there exists a set of orthonormal vectors $\boldsymbol{y}_j$, $j \in [k]$ in the row space of $R^{(t)}$ such that

$$\Delta\left(\hat{A}^{(t)}; \boldsymbol{y}_j^{(t)}, j \in [k]\right) \geqslant \sum_{i=1}^{k} \hat{\sigma}_i^{(t)2} - \frac{10 k \alpha_t}{r} ||\hat{A}^{(t)}||_F^2 \geqslant \sum_{i=1}^{k} \hat{\sigma}_i^{(t)2} - \frac{\epsilon}{8} ||\hat{A}^{(t)}||_F^2. \tag{5.39}$$

Then following from Lemma 5.1, we have

$$\Delta\left(R^{(t)}; \boldsymbol{y}_j^{(t)}, j \in [k]\right) \geqslant \Delta\left(\hat{A}^{(t)}; \boldsymbol{y}_j^{(t)}, j \in [k]\right) - k^2\theta||\hat{A}^{(t)}||_F^2 \tag{5.40}$$

$$\geqslant \sum_{i=1}^{k} \hat{\sigma}_i^{(t)2} - \frac{3\epsilon}{16}||\hat{A}^{(t)}||_F^2, \tag{5.41}$$

where $k^2\theta \leqslant \epsilon/16$ for the low-rank $k \leqslant 200$. Suppose $\boldsymbol{y}_j = \sum_{i=1}^{r} \beta_i^{(j)} \boldsymbol{u}_i(R^{(t)})$, $j \in [k]$, we define $\boldsymbol{z}_j = \sum_{i=1}^{r} \beta_i^{(j)} \boldsymbol{v}_i(R^{(t)})$, where $\boldsymbol{u}_i(R^{(t)}) \in \mathbb{C}^r$, $\boldsymbol{v}_i(R^{(t)}) \in \mathbb{C}^{N_2}$ are the left and right singular vectors of $R^{(t)}$ respectively. Then $\boldsymbol{z}_j$ in the column space of $R^{(t)}$ satisfies

$$\Delta\left(R^{(t)\dagger}; \boldsymbol{z}_j^{(t)}, j \in [k]\right) = \Delta\left(R^{(t)}; \boldsymbol{y}_j^{(t)}, j \in [k]\right) \tag{5.42}$$

$$\geqslant \sum_{i=1}^{k} \hat{\sigma}_i^{(t)2} - \frac{3\epsilon}{16}||\hat{A}^{(t)}||_F^2. \tag{5.43}$$

Similarly, we apply Lemma 5.1 to $R^{(t)\dagger}$ and $C^{(t)\dagger}$ respectively,

$$\Delta\left(C^{(t)\dagger}; \boldsymbol{z}_j^{(t)}, j \in [k]\right) \geqslant \Delta\left(R^{(t)\dagger}; \boldsymbol{z}_j^{(t)}, j \in [k]\right) - k^2\theta||R^{(t)}||_F^2. \tag{5.44}$$

The orthonormal vectors $\boldsymbol{u}_j(C^{(t)})$ calculated in Step 4 of Algorithm 7 are in the column space of $C^{(t)}$, also in the column space of $R^{(t)}$, then (5.44) becomes

$$\Delta\left(C^{(t)\dagger}; \boldsymbol{u}_j(C^{(t)}), j \in [k]\right) \geqslant \sum_{i=1}^{k} \hat{\sigma}_i^{(t)2} - \frac{3\epsilon}{16}||\hat{A}^{(t)}||_F^2 - \frac{\epsilon}{16}||R^{(t)}||_F^2. \tag{5.45}$$

Apply Lemma 5.1, and Lemma 5.2 to $\hat{A}^{(t)}$, $R^{(t)}$ and $\boldsymbol{v}_i^{(t)}$, we get

$$\Delta(\hat{A}^{(t)}; \boldsymbol{v}_i^{(t)}, i \in T) \geqslant \Delta(R^{(t)}; \boldsymbol{v}_i^{(t)}, i \in T) - (1 + \epsilon/16)k^2\theta||\hat{A}^{(t)}||_F^2 \tag{5.46}$$

$$\geqslant \Delta(R^{(t)\dagger}; \boldsymbol{u}_i^{(t)}, i \in T) - (1 + 3\epsilon/16)k^2\theta||\hat{A}^{(t)}||_F^2 \tag{5.47}$$

$$\geqslant \Delta(C^{(t)\dagger}; \boldsymbol{u}_i^{(t)}, i \in T) - k^2\theta||R^{(t)}||_F^2 - (1 + 3\epsilon/16)k^2\theta||\hat{A}^{(t)}||_F^2 \tag{5.48}$$

$$\geqslant \Delta(C^{(t)\dagger}; \boldsymbol{u}_i^{(t)}, i \in [k]) - k\gamma||C^{(t)}||_F^2 - k^2\theta||R^{(t)}||_F^2 - (1 + 3\epsilon/16)k^2\theta||\hat{A}^{(t)}||_F^2 \tag{5.49}$$

Here, we set the threshold for truncating singular values of $C^{(t)}$, $\gamma \triangleq \frac{\epsilon}{8k}$, then $k\gamma \leqslant \frac{\epsilon}{8}$ and $(1+3\epsilon/16)k^2\theta \leqslant \frac{9\epsilon}{16}$. Taking these inequalities into (5.49), we have

$$\Delta(\hat{A}^{(t)}; \boldsymbol{v}_i^{(t)}, i \in T) \geqslant \sum_{i=1}^{k} \hat{\sigma}_i^{(t)2} - \frac{3\epsilon}{4}||\hat{A}^{(t)}||_F^2 - \frac{\epsilon}{8}||R^{(t)}||_F^2 - \frac{\epsilon}{8}||C^{(t)}||_F^2. \tag{5.50}$$

Then for the tensor $\mathcal{A}$, we have

$$\mathrm{E}\left(||\mathcal{A} - \mathcal{A} * \mathcal{V} * \mathcal{V}^{\dagger}||_F^2\right) \leqslant \sum_{t=1}^{N_3} \mathrm{E}\left(||\hat{A}^{(t)} - \hat{A}^{(t)} V^{(t)} V^{(t)\dagger}||_F^2\right) \tag{5.51}$$

$$\leqslant \sum_{t=1}^{N_3} \sum_{i=k+1}^{\boldsymbol{r}(t)} \sigma_i^{(t)2} + \sum_{t=1}^{N_3} \left(\frac{3\epsilon}{4}||\hat{A}^{(t)}||_F^2 + \frac{\epsilon}{8}||R^{(t)}||_F^2 + \frac{\epsilon}{8}||C^{(t)}||_F^2\right) \tag{5.52}$$

$$= \sum_{t=1}^{N_3} \sum_{i=k+1}^{\boldsymbol{r}(t)} \sigma_i^{(t)2} + \epsilon||\hat{A}^{(t)}||_F^2. \tag{5.53}$$

Note $||\mathcal{A}||_F^2 = ||\hat{\mathcal{A}}||_F^2 = ||\mathcal{R}||_F^2 = ||\mathcal{C}||_F^2$ due to the normalization method adopted in Algorithm 7.

□

5.4 Complexity analysis

For simplicity, we consider a tensor $\mathcal{A} \in \mathbb{R}^{N \times N \times N}$ with the same dimensions. The computational cost of classical trunated t-svd for tensor approximation is $\mathcal{O}(N^4 + N^3 \log N)$, where $\mathcal{O}(N^4)$ is the cost of computing the SVD of all frontal slices and $\mathcal{O}(N^3 \log N)$ is for transforming to the Fourier domain. By contrast, the complexity of Algorithm 7 for finding a description of the approximation tensor $\mathcal{A} * \mathcal{V} * \mathcal{V}^{\dagger}$ is only $\mathcal{O}(k^5 N/\epsilon^4 + N^3 \log N)$. The first part of the cost is mainly focus on the truncated-k SVD performed on each frontal slice of the small sampled tensor $C^{(t)} \in \mathcal{R}^{r \times r}$, where $r = \mathcal{O}(k^2/\epsilon^2)$. The second part is for performing the DFT on all tubes $\hat{\mathcal{A}}(i, j, :)$. On the other hand, if we need to calculate the approximated tensor $\mathcal{A} * \mathcal{V} * \mathcal{V}^{\dagger}$ definitely,

the complexity of Algorithm 7 is $\mathcal{O}(k^5N/\epsilon^4 + kN^3 + N^3\log N)$ and it is advantageous in the case of $k << N$ when compared with the classical t-svd.

In Table 5.1, we compare the computational and storage cost of several similar works on randomized tensor approximation using different tensor decompositions. The complexity of these algorithms are calculated without power iterations. It turns out that our algorithm is the most efficient one either in computational or storage complexity.

The random t-svd (rt-svd) approach proposed by Jiani Zhang et al. [90] is nearly four times more expensive than our algorithm. Compared with rt-svd algorithm, our algorithm calculates the compact matrix SVD on $C^{(t)} \in \mathbb{C}^{r_1 \times r_2}$ and the projector $V^{(t)}$ after random choosing the slices. Note that the cost of computing the projector $V^{(t)} = R^{(t)\dagger}U^{(t)}{\Sigma^{(t)}}^{-1}$ is greatly reduced due to decrease on the dimension of $R^{(t)}$ and $U^{(t)}$. In Step 6, we project each $\hat{A}^{(t)}$ on its approximated top rank-k space just like the final step of rt-svd algorithm [90]. Therefore, our algorithm is more computationally efficient than the rt-svd method. In terms of storage cost, unlike the rt-svd method storing three SVD decomposition matrices of $B^{(t)}$, our algorithm only needs to store the information of projector matrices $V^{(t)}$, so the storage cost is only half of that of rt-svd.

The complexity of random Tucker3 method (r-Tucker3) [94] is nearly eight times as expensive computationally compared with our algorithm. As to the storage cost, it has to store all factor matrices and core tensor, so storage cost is much more expensive than our algorithm. Moreover, truncated rank-(k_1, k_2, k_3) tensor for HOSVD is not an optimal approximation in most cases. By contrast, our algorithm is designed based on the t-svd whose truncating form do yield an optimal approximation in the Frobenius norm (see Lemma 4.1).

In [15], N. Benjamin Erichson et al. propose a randomized CP algorithm (r-cp). They first apply random projections with power iteration to get a compressed tensor,

then perform CPD on this compressed tensor by some optimizaton algorithms, and finally utilize this compressed CPD to approximate the original CP factor matrices. However, the cost of the step of compressing tensor alone is nearly 12 times as computationally expensive as our algorithm, since it needs to multiply each tensor mode with a Gaussian matrix, then applies QR decomposition on these products. The storage cost is also greater than our algorithm since for compressing the tensor, since it requires to save the projectors of all modes.

In [80], the authors propose a fast Monte-Carlo algorithm for tensor approximation and tensor decomposition (including tensor CX and CUR decompositions). For rank-k approximation based on t-CX decomposition, this algorithm projects tensor $\mathcal{A}$ onto the space spanned by the lateral slices of $\mathcal{A}$ chosen based on their leverage scores. However, the cost for computing the leverage scores of the lateral slices of $\mathcal{A}$ is extremely high, and it is still high if using leverage scores of $\mathcal{C}$ as a substitute. Besides, this algorithm needs to calculate the tensor project operator with computational complexity $\mathcal{O}(N^4)$. By contrast, Algorithm 7 calculates the tubal rank-k approximation of the sampled tensor with size $r \times r \times N_3$, and the cost is reduced to only $\mathcal{O}(r^3 N)$.

Table 5.1: The complexity comparison of randomized tensor approximation methods based on different tensor decompositons. The cost of r-cp algorithm listed below refers to the cost of computing compressed tensor which is just one step of this algorithm.

Algorithm	computational cost	storage cost
r-Tucker3	$8kN^3$	$k^3 + 3Nk$
rt-svd	$4kN^3 + \mathcal{O}(N^3 \log N)$	$2kN^2$
r-cp	$(9k + 3k^2)N^2 + 12kN^3$	$3kN$
t-CX	$N^4 + 2cN^3 + kN^3$	kN^2
Algorithm 7	$kN^3 + kr_1r_2 + kr_1N^2 + \mathcal{O}(N^3 \log N)$	kN^2

5.5 Simulation result

In this section, we study the accuracy and the computational time of Algorithm 7 on several synthetic and real-world tensors. The results are organized as follows: In Section 5.5.1, we investigate the accuracy of our proposed algorithms varying with different parameters, including dimension, tubal-rank and the subsampled number using randomized tensors. In Section 5.5.2, we compare the performance of our algorithm to the matrix FKV algorithm [17] using synthetic data. Finally, in Section 5.5.3, we apply the proposed algorithm on datasets Movielens and Yahoo! Webscope Movies and compare the result with the rt-svd algorithm proposed in [90].

All the experiments are performed under Windows 10, Python 3.7 and MATLAB R2016b running on a desktop (Intel Core i7 @ 3.60 GHz, 32.0G RAM). In each experiment we repeat 10 times and average the results.

In Theorem 5.4, we estimate the approximated error of Algorithm 7 by the standard of the theoretical minimum error. Here, the relative error RSE and the theoretical minimum relative error $e_k^{\min}$ defined as

$$\text{RSE} \triangleq \frac{||\mathcal{A} - \tilde{\mathcal{A}}_k||_F}{||\mathcal{A}||_F}, \tag{5.54}$$

$$e_k^{\min} \triangleq \frac{||\mathcal{A} - \mathcal{A}_k||_F}{||\mathcal{A}||_F} = \frac{||\mathcal{S}(k+1:N, k+1:N)||_F}{||\mathcal{S}||_F}, \tag{5.55}$$

respectively, are used as performance metrics of Algorithm 7.

5.5.1 Test on randomized tensors

We next test Algorithm 7 on randomized third-order tensors. A random data tensor $A \in \mathbb{R}^{N\times N\times N}$ is generated by summarizing r outer product of three Gaussian random vectors, where r is the CP-rank (also the tubal-rank) of the tensor. The singular values of $\mathcal{A}$ are randomly generated using the quadrant law.

Table 5.2 shows the relative error RSE of Algorithm 7 decreases as the dimension N and sampled number r_1 increase. The sampling numbers r_1 for horizontal slices and r_2 for lateral slices could be constant, independent of the dimension of tensor, or logarithmic in the size of N. Thus, our algorithm can be performed efficiently even for extremely high-dimensional data, outperforming other random tensor approximation algorithms achieved by multiplying a Gaussian randomized tensor such as [44, 90].

In Table 5.2, we observe that our algorithm is only advantageous for very large dimensional tensors. When the tensor dimension is low, even we decrease the tubal-rank and truncation term, the relative error RSE is still much higher than $e_{\min}$, see the first three lines of Table 5.2. For tensor $\mathcal{A}$ of dimension 300, we gradually increase the sampled number from 80 to 200, and drop the tubal-rank and truncation term to 4. However, we only observe a minor decline in the relative error RSE, which is still very large. That means we need to even increase the dimension in order to get a low relative errorRSE. For large dimensional tensor $800 \times 800 \times 500$, the relative error RSE drops noticeably with tubal-rank. Specifically, when we drop the tubal-rank and truncation term to 3, RSE can drop to very low 0.0483. In summary, our algorithm can have comparatively low relative errors in short times even for extremely large-dimensional tensors. However, our algorithm need the original tensor to be very low tubal-rank.

It should be notice that the truncation term k is set to be the tubal-rank of $\mathcal{A}$. This follows the idea of matrix FKV algorithm [17], in which the truncate-k matrix SVD is performed on the sampled matrix on the condition that the original matrix A has a good rank-k approximation. Here in numerical experiment, in order to better capture the change of RSE with truncation term k, we suppose that the truncation term k is not greater than the tubal-rank.

Table 5.2: The comparison of RSE with $e_k^{\min}$ of tensor $\mathcal{A} \in \mathbb{R}^{N\times N\times N}$, where $r_1 = r_2$.

N	$\boldsymbol{r}$	k	iter	r_1	RSE	$e_k^{\min}$
100	16	10	30	80	0.3302	0.1467
100	8	7	30	80	0.3325	0.0413
100	3	3	30	80	0.1435	0
300	16	10	30	80	0.4330	0.1662
300	16	10	30	150	0.309	0.1662
300	16	10	30	200	0.2915	0.1662
300	4	4	30	150	0.1763	0
500	10	8	30	350	0.1801	0.0859
500	3	3	30	350	0.0713	0
$800 \times 800 \times 500$	10	8	30	350	0.1471	0.0831
$800 \times 800 \times 500$	3	3	30	350	0.0483	0

5.5.2 Comparison with matrix FKV

In this section, we compare the result of Algorithm 7 with the fast Monte-Carlo algorithm for finding low-rank approximation proposed by Alan Frieze, Ravi Kannan and Santosh Vempala [17].

In Table 5.3, we evaluate running time and the accuracy of the matrix FKV algorithm, in terms of the relative error of approximated singular values and reconstructed matrix A. To generate Gaussian random matrices, we follow the methodology adopted in [2]. That is, in order to generate a Gaussian random matrix A of dimension $m \times n$, rank k, conditional number κ, we build the singular vectors and singular values respectively, i.e., suppose $A = U\Sigma V$, we first generate an $m \times k$ Gaussian random matrix U' whose entries drawn independently from the standard normal distribution $\mathcal{N}(0, 1)$, then perform a reduced QR decomposition on $U' = QR$ and let $U \triangleq Q$. A similar way is followed to generate V. For the given κ, we first sample the largest singular value uniformly in $[1, 500]$ and then the minimal singular value is fixed. Other singular values are sampled using the quadrant law introduced

in [74].

In Table 5.4, we test the same metrics of Algorithm 7. Here, we follow the similar way to generate Gaussian randomized tensors with matrix FKV algorithm [17]. We first generate a Gaussian random tensor $\mathcal{U}_1 \in \mathbb{R}^{N_1 \times k \times N_3}$ with entries drawn from the standard normal distribution, then perform DFT along the third mode, getting $\hat{\mathcal{U}}_1$. Then we perform the reduced QR decomposition on every frontal slice of $\hat{\mathcal{U}}_1$, i.e., $\hat{\mathcal{U}}_1(:,:,t) = \hat{\mathcal{Q}}(:,:,t)\hat{\mathcal{R}}(:,:,t)$, for $t \in [N_3]$, and let $\hat{\mathcal{Q}} = \hat{\mathcal{U}}$. Similarly, we generate tensor $\hat{\mathcal{V}}$ in this way. Finally, the singular values of $\mathcal{A}$ are generated using the quadrant law for matrices adopted in simulating matrix FKV.

From Tables 5.3 and 5.4, we can see that the accuracy of Algorithm 7 is better than matrix FKV algorithm. It should be notice that Algorithm 7 can better capture the relations between different frontal slices of the original tensor due to the DFT, so it could have better performance in some applications where the relations of two modes of the tensor are addressed. However, the computational time of Algorithm 7 is more than the matrix FKV due to the extra time of performing DFT on N^2 tubes of the tensor.

Table 5.3: The result of matrix FKV algorithm [17] performed on matrix $A \in \mathbb{R}^{m \times n}$. k is the rank of A and r is the sample number. e_A is the relative error for reconstructing A using the approximated singular values $\tilde{\sigma}_l$ and singular vectors $\tilde{\boldsymbol{u}}_l$, $\tilde{\boldsymbol{v}}_l$, i.e., $e_A = ||\tilde{A} - A||_F / ||A||_F$ with $\tilde{A} = \sum_{l=1}^{k} \tilde{\sigma}_l \tilde{\boldsymbol{u}}_l \tilde{\boldsymbol{v}}_l$. The mean relative error of approximated singular values $e_\sigma = \frac{1}{k} \sum_{l=1}^{k} |\sigma_l - \tilde{\sigma}_l| / \sigma_l$.

$m * n$	k	r	e_A	e_σ	sampling time	SVD on C and reconstruct A time
1000*3000	3	100	0.1475	0.089	5.9	1.34
10000*30000	5	100	0.1696	0.02	551	744
10000*30000	5	300	0.1434	0.033	543	760
10000*30000	3	300	0.08	0.063	540	462

Table 5.4: The result of Algorithm 7. k is the tubal-rank of $\mathcal{A}$ and r is the sampled number for sampling the horizontal and lateral slices.

$N_1 * N_2 * N_3$	k	r	e_A	e_σ	sampling time	SVD on $\mathcal{C}$ and reonstruct $\mathcal{A}$ time
1000*1000*3	3	100	0.105	0.0642	6.08	0.207
$10^4 * 10^4 * 3$	5	100	0.156	0.029	652	626
$10^4 * 10^4 * 3$	5	300	0.107	0.12	717	667
$10^4 * 10^4 * 3$	3	300	0.08	0.0622	686	607

5.5.3 Test on real datasets

In this section, we perform Algorithm 7 on Movielens and Yahoo! Webscope Movies datasets. The Movielens dataset has 610 users, 9742 movies and 100K ratings with a timestamp. For preprocessing the data, we divide the timestamps that users give ratings into 60 timeslots, obtaining a third-order tensor $\mathcal{A} \in \mathbb{R}^{610\times 9742\times 60}$ in which the three modes represent users, movies, and time respectively. The tubal-rank of this dataset is 610.

The second dataset, Yahoo! Webscope Movies dataset, has 7642 users, 11915 movies and 221K ratings in a $\{1, \cdots, 5\}$ scale. We select the first 800 users and 4623 corresponding movies. Besides, the original Yahoo! Webscope Movies dataset contains user age and gender features. We choose user's year of birth as the third dimension and consider it as the context variable. Therefore, the size of resulting tensor is $800 \times 4623 \times 51$ with 23782 nonzero entries. The tubal-rank of this dataset is 742.

The compared algorithm we choose is rt-svd algorithm proposed by Jiani Zhang et al. [90]. This algorithm first performs DFT along the third mode of the tensor, then combines the columns of $\hat{A}^{(t)}$ by multiplying a Gaussian randomized tensor. After that it performs QR decomposition to obtain the projector matrix $Q^{(t)}$, then computes the compact SVD of $B^{(t)} = Q^{(t)\dagger}\hat{A}^{(t)}$, finally multiply $Q^{(t)}$ to get the approximated tensor $\tilde{\mathcal{A}}$. We first apply rt-svd algorithm to the original Movielens

dataset whose corresponding tensor has full rank 610. When we set the truncation term as $k = 5, 6, 7, 8, 9$ and the oversampling parameter $p = 2$, the relative error is [0.82 0.81 0.79 0.77 0.76], corresponding to the the theoretical minimum relative error [0.81 0.79 0.77 0.75 0.74]. We can see that although the relative error RSE and the theoretical minimum relative error $e_k^{\min}$ are close, they are all very large. Similar result also appears in our algorithm.

From the above discussion, we find that the original datasets have to be preprocessed in order to satisfy the low tubal-rank assumption of Algorithm 7. Here, we keep the top 10 tubes under Fourier domain and use this dataset as the original dataset. The result of Algorithm 7 and rt-svd [90] can be found in Table 5.5. It can be seen that Algorithm 7 outperforms rt-svd algorithm in terms of running time and accuracy.

Table 5.5: The result of Algorithm 7 and rt-svd algorithm applied to Movielens dataset. k denotes the truncate term. RSE, re_A denote the relative errors, and t_1, t_2 denote the running time (seconds) of Algorithm 7 and rt-svd algorithm respectively.

k	RSE	$e_k^{\min}$	t_1	re_A	t_2
7	0.3691	0.3388	1567	0.4117	3016
8	0.3012	0.2708	1595	0.3235	3146
9	0.2243	0.1861	1617	0.2254	3292
10	0.086	0	1618	0	3494

Table 5.6: The result of Algorithm 7 and rt-svd algorithm applied to Yahoo! Webscope Movies dataset.

k	RSE	$e_k^{\min}$	t_1	re_A	t_2
7	0.3701	0.3354	2279	0.3743	3790
8	0.3033	0.2659	2306	0.2957	3965
9	0.2110	0.1810	2367	0.2034	4230
10	0.089	0	2409	0.1285	4421

5.6 Conclusion

In this chapter, we introduce a randomized low tubal-rank tensor approximation algorithm using length-squared sampling and the truncated t-svd algorithm. Our algorithm extends the fast Monte-Carlo method for low-rank matrix approximations to third-order tensors and approximates the original tensor in Fourier domain. It can be effectively used in the approximation of high dimensional tensors with low tubal-rank which is validated by numerical experiments.

Chapter 6

Large-size Schrödinger cat states generation

6.1 Introduction

As a typical mesoscopic system, Schrödinger cat states show the superposition of two classically distinguishable states. Typical examples of optical Schrödinger cat states are coherent states [10] represented by $|\beta\rangle + e^{i\phi}|-\beta\rangle$ for even cat and odd cat, when $\phi = 0$ and $\phi = \pi$, respectively, where β and $-\beta$ are the amplitudes of coherent states with opposite phase. $|\beta|$ is defined as the size of Schrödinger cat states. Such superpositions are different from statistical mixtures by revealing interference features between superposed components, which plays an important role in validation of quantum theory[27], quantum communication[69, 6, 36, 50], continuous-variable quantum computation [24, 62, 42, 19] and quantum metrology[48, 28, 16].

Several protocols have been proposed to generate Schrödinger cat states with small amplitudes, which are also called Schrödinge kitten states. The most popular approach is to subtract odd/even photon(s) from a squeezed vacuum state [10] from an optical parametric oscillator (OPO) to produce odd/even kittens at based band [55, 51, 85, 49, 18] as well as sidebands[71]. The maximum amplitude of generated odd cat state of $|\beta| = 1.76$ and fidelity of 0.59 was reported[18]. Another method

is based on Fock state and homodyne detection[54], by which an even cat with $|\beta| = \sqrt{2.6}$ was demonstrated.

However, the overlap between two superposed coherent states in a Schrödinger cat state, i.e. $\langle\beta| - \beta\rangle = exp(-2|\beta|^2)$, is required to approach zero to effectively work as a qubit in quantum information science. As a result, the amplitude of a Schrödinger cat state must satisfy $|\beta| \geqslant 2$. Therefore, large-size Schrödinger cat states generation attracted intense interest. An Schrödinger odd cat with $|\beta| = 1.15$ was enlarged to be an Schrödinger even cat with $|\beta| = 1.85$ with a fidelity of 0.77 [76] by combining two Schrödinger cat states with small size by a beam splitter and conducting homodyne detection[41]. Recently, Eaton et al proposed a sequential photon catalysis scheme to generated large-size squeezed Schrödinger cat states [13], in which several-stage photon catalysis are required to breed the cat state. Takese et al reported a high-generate-rate optical cat state scheme based on the general photon subtraction of two squeezed vacua [78].

Recently, a new theory based on the α-representation in infinite Hilbert space was developed, by which two schemes based on two-mode entangled state and separated Fock states, respectively were proposed[46]. Schrödinger cat states of large-size $|\beta| > 2$ with a fidelity around 0.99 based on a two-mode 9-photon entangled state and $|\beta| = 2$ with the maximum fidelity of 0.973 based on five Fock states, respectively, were predicted[46].

In this chapter, we report an approach to generate large-size Schrödinger cat states with high fidelity based on conditioned measurement. Besides photon subtraction from a squeezed vacuum state, photon adding is also implemented, which effectively engineers the high-order photon number of the generated state. When two sets of such scheme is combined, an amplification on the amplitude of Schrödinger cat is obtained with high fidelity.

This chapter is organized as follows. In section 6.2, a Schrödinger cat state model

is derived and the fidelity is quantitatively analysed. In section 6.3, an effective approach to produce a large-size Schrödinger cat states is proposed and analysed. Concluding remarks are provided in section 6.4.

6.2 Photon-adding-subtraction model

6.2.1 Schrödinger cat states

Optical Schrödinger cat states are the superposition state of two coherent states of with opposite phase,$|\beta\rangle$ and $|-\beta\rangle$[1], which can be written as

$$|\beta\rangle = e^{\frac{-|\beta|^2}{2}} \sum_{n=0}^{\infty} \frac{\beta^n}{\sqrt{n!}} |n\rangle, \tag{6.1}$$

$$|-\beta\rangle = e^{\frac{-|\beta|^2}{2}} \sum_{n=0}^{\infty} \frac{(-\beta)^n}{\sqrt{n!}} |n\rangle. \tag{6.2}$$

An even cat and odd are written as

$$|EvenCat\rangle = N_+(|\beta\rangle + |-\beta\rangle) = N_+ e^{\frac{-|\beta|^2}{2}} \sum_{n=0}^{\infty} \frac{2\beta^{(2n)}}{\sqrt{(2n)!}} |2n\rangle, \tag{6.3}$$

Where,

$$N_+ = \frac{1}{\sqrt{2(1+e^{-2|\beta|^2})}} \tag{6.4}$$

and an odd cat is expressed as

$$|OddCat\rangle = N_-(|\beta\rangle - |-\beta\rangle) = N_- e^{\frac{-|\beta|^2}{2}} \sum_{n=0}^{\infty} \frac{2\beta^{(2n+1)}}{\sqrt{(2n+1)!}} |2n+1\rangle, \tag{6.5}$$

Where,

$$N_- = \frac{1}{\sqrt{2(1-e^{-2|\beta|^2})}}. \tag{6.6}$$

To ensure two superposed coherent states are orthogonal, the overlap between $|\beta\rangle$ and $|-\beta\rangle$ should approach to zero, i.e.

$$\langle\beta \mid -\beta\rangle = e^{-|\beta|^2} \sum_{n=0}^{\infty} \frac{(-\beta^2)^n}{n!} = e^{-2|\beta|^2} \ll 0 \tag{6.7}$$

When $|\beta| \geqslant 2$, $\langle\beta| - \beta\rangle \approx 0$.

6.2.2 *l*-adding and *k*-subtraction squeezed vacuum state

Similar to an even cat state, a squeezed vacuum state only contains even-photon distribution, which is written as [10, 5] when the squeezing angle is taken as zero,

$$\hat{S}(\xi)|0\rangle = \sum_{n=0}^{\infty} \alpha_{2n}|2n\rangle, \tag{6.8}$$

where

$$\alpha_{2n} = \frac{1}{\sqrt{\cosh\xi}} \frac{\sqrt{(2n)!}\tanh^n\xi}{2^n n!}. \tag{6.9}$$

Thus, squeezed vacuum states approximate even cat states with small size. Odd-photon distribution can be obtained if odd photons such as 1 or 3 are subtracted from a squeezed vacuum states. So it is an effective approach to produce odd cat states with small amplitudes($|\beta| \leqslant 1.2$ for $F = 0.99$)[41, 5]. To enlarge the size of the generated Schrödinger cat state, we add $l$photons while subtracting k photons from a squeezed vacuum state as shown in Fig. 6.1.

The total input of the system can be written as,

$$|\Psi_{in}\rangle = |l\rangle \otimes \sum_{n=0}^{\infty} \alpha_{2n}|2n\rangle \tag{6.10}$$

$$= \sum_{n=0}^{\infty} \alpha_{2n} \frac{1}{\sqrt{N_l}\sqrt{2n!}} \prod_{k=1}^{l} B_1^*(\xi^{1k}) \prod_{k=1}^{2n} B_2^*(\xi^{2k})|0_1\rangle \otimes |0_2\rangle. \tag{6.11}$$

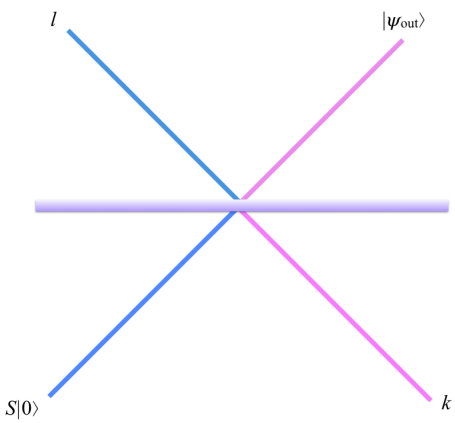


Figure 6.1: Schematic of Schrödinger cat state generation based on conditional measurement

Where, $B^*(\xi)$ is an creation operator in continuous mode, which is related to the continuous-mode Fock state $|n\rangle$ by

$$|n\rangle = \frac{1}{\sqrt{n!}}(B^*(\xi))^n|0\rangle, \quad \|\xi\|_2 = 1. \tag{6.12}$$

Let the first output channel is measured by means of the state $|k\rangle$. Setting $\ell + j - i = k$ we get $i = \ell - k + j$. As $0 \leqslant i \leqslant \ell$, we have $j \geqslant k - \ell$ and $j \leqslant k$. The *unnormalized* state at the second output channel becomes

$$|\Psi_{out,conditioned}\rangle = \sum_{n=0}^{\infty} \gamma_{n,lk}|2n + l - k\rangle \tag{6.13}$$

$$\tag{6.14}$$

Where,

$$\gamma_{n,lk} = \frac{1}{\sqrt{l!}}\frac{\alpha_{2n}}{\sqrt{(2n)!}} \sum_{j=\max(k-\ell,0)}^{\min(k,2n)} \binom{\ell}{\ell - k + j}\binom{2n}{2n - j}$$
$$\sqrt{k!(2n + \ell - k)!}(-1)^j T^{2n+k-2j} R^{l-k+2j}. \tag{6.15}$$

The fidelity between the generated state and the Schrödinger cat state is

$$F(\Psi_{out,conditioned},\psi) = |\langle\psi|\Psi_{out,conditioned}\rangle|^2. \tag{6.16}$$

When $l-k$ is odd, an odd cat state is produced, while an even cat state is generated when $l-k$ is even. Generally, $l = 0$, and $k = 1$, i.e. one-photon-subtracted squeezed vacuum state is set to generate odd cat states. However, such scheme can only generate an approximate Schrödinger's cat with the maximum amplitude of 1.2

6.3 Enlargement of Schrödinger kitten states

An approach to generate large-size Schrödinger cat states based on l-photon-added and k-photon-subtracted squeezed vacuum states is proposed as shown in Fig. 6.2.

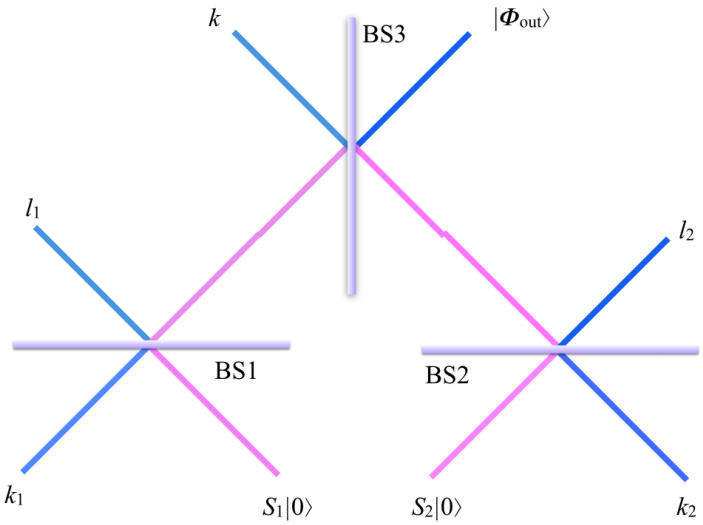


Figure 6.2: Schematic of Schrödinger cat state amplification based on conditional measurement

Two Schrödinger kitten states generated with l-photon added and k-photon-subtracted squeezed states are combined on the third beam splitter BS_3. The input state to the third beam splitter is $|\Psi^1_{\text{out,conditioned},l_1k_1}\rangle\otimes|\Psi^2_{\text{out,conditioned},l_2k_2}\rangle$, which can be written as

$$|\Psi^1_{\text{out,conditioned},l_1k_1}\rangle\otimes|\Psi^2_{\text{out,conditioned},l_2k_2}\rangle = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty}\gamma^1_{n,\ell_1k_1}\gamma^2_{m,\ell_2k_2}|2n+\ell_1-k_1\rangle\otimes|2m+\ell_2-k_2\rangle, \tag{6.17}$$

Where, γ^1_{n,ℓ_1k_1} and γ^2_{n,ℓ_1k_1} have the same format to γ_{nlk} in Eq. (6.15).

The total input state is a linear combination of products $|2n+\ell_1-k_1\rangle\otimes|2m+\ell_2-k_2\rangle$, where $n, m = 0, \ldots, \infty$. Identifying n_1 with $2n+\ell_1-k_1$ and n_2 with $2m+\ell_2-k_2$ respectively. After the third beamsplitter, the product will turn to

$$\begin{aligned}
&|2n+\ell_1-k_1\rangle\otimes|2m+\ell_2-k_2\rangle \\
\rightarrow\ &\frac{1}{\sqrt{(2n+\ell_1-k_1)!(2m+\ell_2-k_2)!}}\sum_{j=0}^{2n+\ell_1-k_1}\sum_{k=0}^{2m+\ell_2-k_2}\binom{2n+\ell_1-k_1}{j}\binom{2m+\ell_2-k_2}{k}(-1)^k \\
&T_3^{2m+\ell_2-k_2+j-k}R_3^{2n+\ell_1-k_1-j+k}\sqrt{(j+k)!(2n+\ell_1-k_1+2m+\ell_2-k_2-j-k)!} \\
&|j+k\rangle|2n+\ell_1-k_1+2m+\ell_2-k_2-j-k\rangle. \\
=\ &\sum_{j=0}^{2n+\ell_1-k_1}\sum_{k=0}^{2m+\ell_2-k_2}\gamma_{n,m,j,k}|j+k\rangle|2n+\ell_1-k_1+2m+\ell_2-k_2-j-k\rangle
\end{aligned} \tag{6.18}$$

where

$$\begin{aligned}
\gamma_{n,m,j,k} \triangleq\ &\frac{1}{\sqrt{(2n+\ell_1-k_1)!(2m+\ell_2-k_2)!}}\binom{2n+\ell_1-k_1}{j}\binom{2m+\ell_2-k_2}{k}(-1)^k \\
&\times T_3^{2m+\ell_2-k_2+j-k}R_3^{2n+\ell_1-k_1-j+k}\sqrt{(j+k)!(2n+\ell_1-k_1+2m+\ell_2-k_2-j-k)!}
\end{aligned} \tag{6.19}$$

is the corresponding coefficient. Consequently, the final output state of the third beamsplitter is

$$|\Psi\rangle = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty}\gamma^1_{n,\ell_1k_1}\gamma^2_{m,\ell_2k_2}\sum_{j=0}^{2n+\ell_1-k_1}\sum_{k=0}^{2m+\ell_2-k_2}\gamma_{n,m,j,k}|j+k\rangle|2n+\ell_1-k_1+2m+\ell_2-k_2-j-k\rangle. \tag{6.20}$$

n particular, if $\ell_1 - k_1 = 1$ and $\ell_2 - k_2 = 1$, then Eq.(6.19) and Eq.(6.20) become

$$\begin{aligned}\gamma_{n,m,j,k} &= \frac{1}{\sqrt{(2n+1)!(2m+1)!}}\binom{2n+1}{j}\binom{2m+1}{k}(-1)^k \\ &\quad \times T_3^{2m+1+j-k} R_3^{2n+1-j+k}\sqrt{(j+k)!(2(n+m+1)-j-k)!}, \quad (6.21)\end{aligned}$$

and

$$|\Psi\rangle = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty}\gamma^1_{n,\ell_1 k_1}\gamma^2_{m,\ell_2 k_2}\sum_{j=0}^{2n+1}\sum_{k=0}^{2m+1}\gamma_{n,m,j,k}|j+k\rangle|2(n+m+1)-j-k\rangle, \quad (6.22)$$

respectively.

In particular, if $\ell_1 - k_1 = 1$ and $\ell_2 - k_2 = 1$, we rewrite Eq.(6.19) and Eq.(6.20) below

$$\begin{aligned}\gamma_{n,m,j,k} &= \frac{1}{\sqrt{(2n+1)!(2m+1)!}}\binom{2n+1}{j}\binom{2m+1}{k}(-1)^k \\ &\quad \times T_3^{2m+1+j-k} R_3^{2n+1-j+k}\sqrt{(j+k)!(2(n+m+1)-j-k)!}, \quad (6.23)\end{aligned}$$

$$|\Psi\rangle = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty}\gamma^1_{n,\ell_1 k_1}\gamma^2_{m,\ell_2 k_2}\sum_{j=0}^{2n+1}\sum_{k=0}^{2m+1}\gamma_{n,m,j,k}|j+k\rangle|2(n+m+1)-j-k\rangle. \quad (6.24)$$

In what follows, we discuss two special case: one and three photons are measured, i.e. $k = 1$, and $k = 3$ in Fig. 6.2.

6.3.1 Amplification of kitten states with $k = 1$

l_1=l_2=2 and k_1=k_2=1

When l_1=l_2=2 and k_1=k_2=1, when $\xi = 0.6$ and the reflectivity of BS3, R_3, is optimized, the variation of β with fidelity is shown in Fig. 6.3. From Fig. 6.3, it can be seen that the size of the kitten state is amplified from 1.225 to 1.241 by 1.01 times.

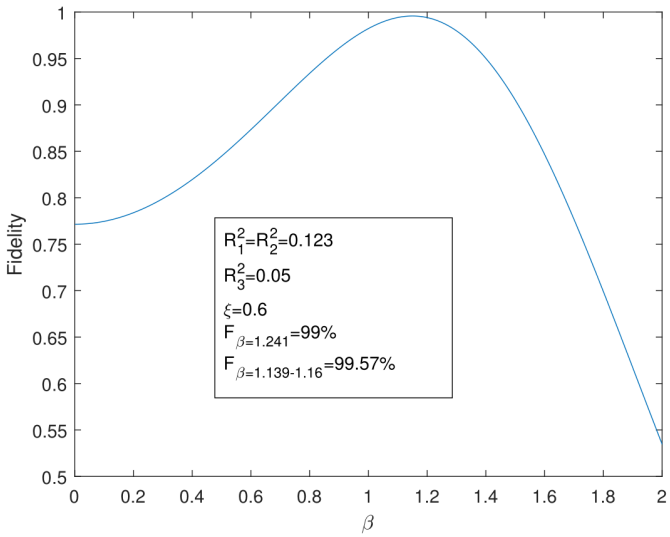


Figure 6.3: Variation of fidelity with β when $\xi = 0.6$ in the case of kitten state amplification

l_1=l_2=3 and k_1=k_2=2

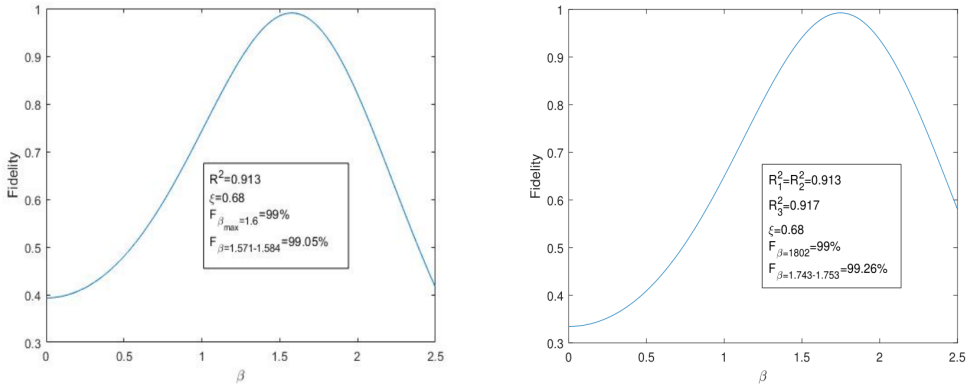
When $\xi = 0.68$, Fig. 6.4(a) shows the variation between β and fidelity of an odd Schrödinger cat state generated from the setup shown in Fig. 6.1 when $l = 3$ and $k = 2$. The maximum β can reach 1.6 to ensure that the fidelity is over 0.99. While when two odd Schrödinger cat states with $\beta = 1.6$ are combined with BS3 as shown in Fig. 6.2, the amplitude is enlarged to 1.802 with an amplification of 1.13 as indicated in Fig. 6.4(b).

According to the results shown above, the amplification of odd Schrödinger cat states is realized. Particularly, when l_1=l_2=3 and k_1=k_2=2, the amplification as large as 1.13 is achievable in the proposed scheme shown in Fig. 6.2.

6.3.2 Large-size Schrödinger cat states generation

l=3, and k=2

By optimizing the reflectivity of BS1, BS2 and BS3 in Fig. 6.2, an odd Schrödinger cat state with an amplitude as large as 1.991 and fidelity over 0.99 is obtained as



(a) (b)

Figure 6.4: Variation of fidelity with β when $l = 3$ and $k = 2$ (a) One beam splitter (b) Amplified kitten state with three beam splitters.

shown in Fig. 6.5(a), which is 1.5 times of the odd cat generated from one-photon subtracted squeezed vacuum state. When the squeezing parameter, ξ, of two input squeezed vacuum states is enhanced to 0.7, a maximum odd cat state with $\beta = 2.001$ can be obtained with $F = 0.99$ as implied in Fig. 6.5(b). When the squeezing parameter increases to 0.8, the amplitude of the odd cat state could be as large as 2.266 and 2.322 while keeping the fidelity as 0.99 and 0.98, respectively, as shown in Fig. 6.5(c). Fig. 6.7 show the photon number distribution and the Wigner function of generated cat state and the real odd cat state with $\beta = 2.266$.

Therefore, large-size Schrödinger cat states are possible to be generated based on the proposed scheme shown in Fig. 6.2.

l=0, and k=1

When two kitten states generated with the traditional approach, i.e. $l_1 = l_2 = 0$ and $k_1 = k_2 = 1$, are input into the scheme. Large-size cat states with amplitudes of $\beta = 2.31$, $\beta = 2.42$, and $\beta = 2.58$ and fidelity of 0.99 corresponding to $\xi = 0.6$, $\xi = 0.68$, and $\xi = 0.8$ are obtained by optimizing the reflectivity of three beam

splitters as shown in Fig. 6.7. Kitten states with an amplitude of 1.39 and 1.51 and fidelity of 0.975 and 0.9634 are amplified to a cat state with an amplitude of 2.42 and fidelity of 0.99 when $\xi = 0.68$ as shown in Fig. 6.6.

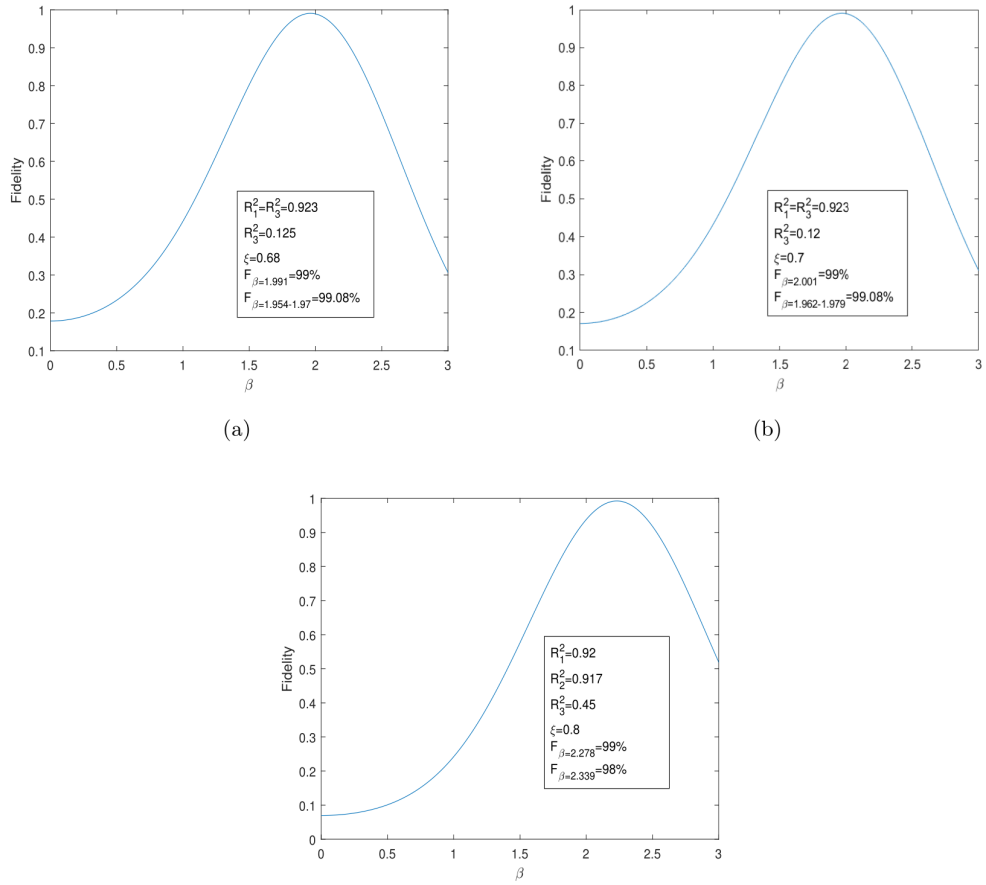


Figure 6.5: Large-size Schrödinger cat state generated from the three-beam-splitters scheme (a) $\beta = 1.991$ when $\xi = 0.68$ (b) $\beta = 2.001$ when $\xi = 0.7$ (c) $\beta = 2.266$ when $\xi = 0.8$

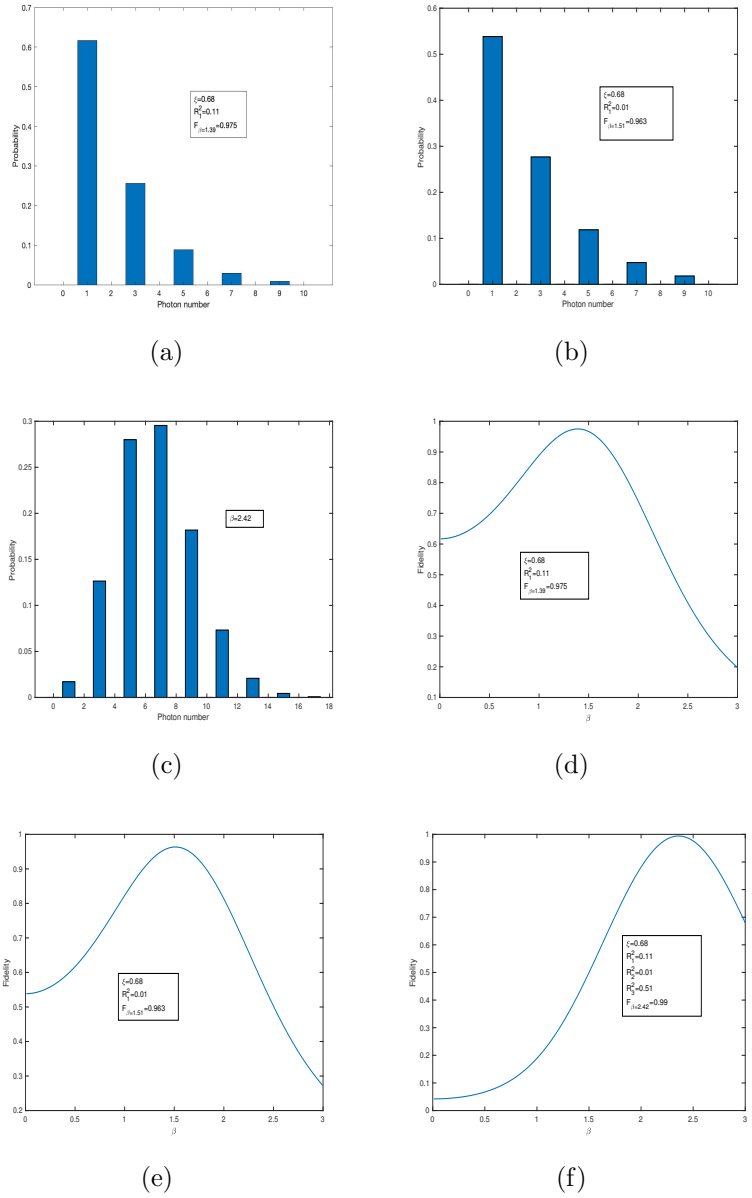
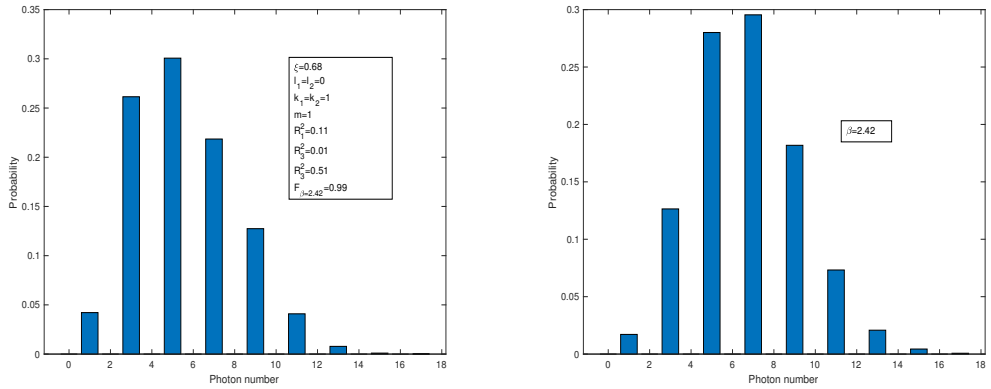
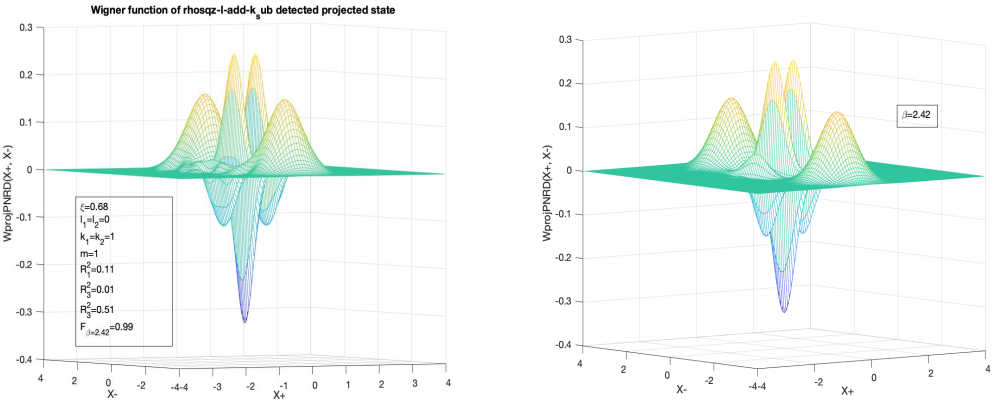


Figure 6.6: Photon number distribution and fidelity variation with β when l=0 and k=1 (a)(d) Input kitten state 1 (b)(e) Input kitten state 2 (c)(f) Amplified cat state



(a) (b)



(c) (d)

Figure 6.7: Photon number distribution and Wigner function when l=0 and k=1 (a) and (c) for the generated odd cat states; (b) and (d) for an ideal odd cat states with $\beta = 2.42$ in theory

6.4 Conclusion

In this chapter, we propose an effective approach to produce a large-size Schrödinger cat states with high fidelity based on conditioned measurement. Through photon subtraction from a squeezed vacuum state and photon adding, an amplification on the amplitude of Schrödinger cat is obtained with high fidelity. The simulation results shows that odd cat states with an amplitude of 2.001 with the fidelity of 0.99 could be obtained.

Chapter 7

Conclusions

In this thesis, we propose a quantum t-svd algorithm for third-order tensors and then extend it to order-p tensors. It can be proved that our quantum t-svd algorithm is polynomial faster than its classical counterpart if we do not recover classical information from the output state. Based on this algorithm, we next present a quantum context-aware recommendation systems algorithm with time complexity $\mathcal{O}(\sqrt{k}N\text{polylog}(N))$, compared to the classical counterpart with complexity $\mathcal{O}(kN^3)$. In fact, it extends Kerenidis and Prakash's matrix recommendation system algorithm to third-order tensors, so our algorithm can incorporate users' contextual information into recommendations. Our algorithm obtains the recommended product by measuring the output quantum state corresponding to an approximation of a user's dynamic preferences in stead of reconstructing the whole tensor. Also, we present a classical Monte-Carlo tensor approximation algorithm based on the truncated t-svd and the randomized low-rank matrix approximations algorithm. At last, we provide two schemes for the effective generation of large-size Schrödinger's cat states. The schemes are based on the linear operation of Fock states and squeezed vacuum states. Through photon subtraction from a squeezed vacuum state and photon adding, an amplification on the amplitude of Schrödinger cat is obtained with high fidelity.

Bibliography

[1] D. Achlioptas and F. McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM (JACM)*, 54(2):9, 2007.

[2] J. M. Arrazola, A. Delgado, B. R. Bardhan, and S. Lloyd. Quantum-inspired algorithms in practice. *Quantum*, 4:307, 2020.

[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[4] A. K. Biswas, M. M. Hasan, A. R. Chowdhury, and H. M. H. Babu. Efficient approaches for designing reversible binary coded decimal adders. *Microelectronics Journal*, 39(12):1693–1703, 2008.

[5] A. M. Brańczyk and T. Ralph. Teleportation using squeezed single photons. *Physical Review A*, 78(5):052304, 2008.

[6] J. B. Brask, I. Rigas, E. S. Polzik, U. L. Andersen, and A. S. Sørensen. Hybrid long-distance entanglement distribution protocol. *Physical review letters*, 105(16):160501, 2010.

[7] C.-F. Chiang. Quantum phase estimation with an arbitrary number of qubits. *International Journal of Quantum Information*, 11(01):1350008, 2013.

[8] B. D. Clader, B. C. Jacobs, and C. R. Sprouse. Preconditioned quantum linear system algorithm. *Physical Review Letters*, 110(25):250504, 2013.

[9] P. Comon. Tensor decompositions. *Mathematics in Signal Processing V*, pages 1–24, 2002.

[10] M. Dakna, T. Anhut, T. Opatrnỳ, L. Knöll, and D.-G. Welsch. Generating schrödinger-cat-like states by means of conditional measurements on a beam splitter. *Physical Review A*, 55(4):3184, 1997.

[11] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[12] P. Drineas and M. W. Mahoney. A randomized algorithm for a tensor-based generalization of the singular value decomposition. *Linear algebra and its applications*, 420(2-3):553–571, 2007.

[13] M. Eaton, R. Nehra, and O. Pfister. Non-gaussian and gottesman–kitaev–preskill state preparation by photon catalysis. *New Journal of Physics*, 21(11):113034, 2019.

[14] G. Ely, S. Aeron, N. Hao, and M. E. Kilmer. 5d seismic data completion and denoising using a novel class of tensor decompositions. *Geophysics*, 80(4):V83–V95, 2015.

[15] N. B. Erichson, K. Manohar, S. L. Brunton, and J. N. Kutz. Randomized cp tensor decomposition. *Machine Learning: Science and Technology*, 1(2):025012, 2020.

[16] A. Facon, E.-K. Dietsche, D. Grosso, S. Haroche, J.-M. Raimond, M. Brune, and S. Gleyzes. A sensitive electrometer based on a rydberg atom in a schrödinger-cat state. *Nature*, 535(7611):262–265, 2016.

[17] A. Frieze, R. Kannan, and S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM (JACM)*, 51(6):1025–1041, 2004.

[18] T. Gerrits, S. Glancy, T. S. Clement, B. Calkins, A. E. Lita, A. J. Miller, A. L. Migdall, S. W. Nam, R. P. Mirin, and E. Knill. Generation of optical coherent-state superpositions by number-resolved photon subtraction from the squeezed vacuum. *Physical Review A*, 82(3):031802, 2010.

[19] A. Gilchrist, K. Nemoto, W. J. Munro, T. C. Ralph, S. Glancy, S. L. Braunstein, and G. J. Milburn. Schrödinger cats and their power for quantum information processing. *Journal of Optics B: Quantum and Semiclassical Optics*, 6(8):S828, 2004.

[20] V. Giovannetti, S. Lloyd, and L. Maccone. Quantum random access memory. *Physical review letters*, 100(16):160501, 2008.

[21] L. Gu, X. Wang, and G. Zhang. Quantum higher order singular value decomposition. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 1166–1171, Oct 2019.

[22] N. Hao, M. E. Kilmer, K. Braman, and R. C. Hoover. Facial recognition using tensor-tensor decompositions. *SIAM Journal on Imaging Sciences*, 6(1):437–463, 2013.

[23] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009.

[24] J. Hastrup, J. S. Neergaard-Nielsen, and U. L. Andersen. Deterministic generation of a four-component optical cat state. *Optics letters*, 45(3):640–643, 2020.

[25] S. Hu, L. Qi, and G. Zhang. Computing the geometric measure of entanglement of multipartite pure states by means of non-negative tensors. *Physical Review A*, 93(1):012304, 2016.

[26] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire. Towards quantum machine learning with tensor networks. *Quantum Science and technology*, 4(2):024001, 2019.

[27] H. Jeong, W. Son, M. Kim, D. Ahn, and Č. Brukner. Quantum nonlocality test for continuous-variable states with dichotomic observables. *Physical Review A*, 67(1):012106, 2003.

[28] J. Joo, W. J. Munro, and T. P. Spiller. Quantum metrology with entangled coherent states. *Physical review letters*, 107(8):083601, 2011.

[29] R. Kannan and S. Vempala. *Spectral algorithms*. Now Publishers Inc, 2009.

[30] R. Kannan and S. Vempala. Randomized algorithms in numerical linear algebra. *Acta Numerica*, 26:95, 2017.

[31] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86, 2010.

[32] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash. q-means: A quantum algorithm for unsupervised machine learning. In *Advances in Neural Information Processing Systems*, pages 4134–4144, 2019.

[33] I. Kerenidis and A. Prakash. Quantum Recommendation Systems. In C. H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:21, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[34] M. E. Kilmer and C. D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011.

[35] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[36] S.-W. Lee and H. Jeong. Near-deterministic quantum teleportation and resource-efficient quantum computation using linear optics and hybrid qubits. *Physical Review A*, 87(2):022326, 2013.

[37] C. Liu, J. Zhou, and K. He. Image compression based on truncated hosvd. In *2009 International Conference on Information Engineering and Computer Science*, pages 1–4. IEEE, 2009.

[38] S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631, 2014.

[39] G. Long. General quantum interference principle and duality computer. *Communications in Theoretical Physics*, 45(5):825, 2006.

[40] G. Long. Duality quantum computing and duality quantum information processing. *International Journal of Theoretical Physics*, 50(4):1305–1318, 2011.

[41] A. Lund, H. Jeong, T. Ralph, and M. Kim. Conditional production of superpositions of coherent states with inefficient photon detection. *Physical Review A*, 70(2):020101, 2004.

[42] A. P. Lund, T. C. Ralph, and H. L. Haselgrove. Fault-tolerant linear optical quantum computing with small-amplitude coherent states. *Physical review letters*, 100(3):030503, 2008.

[43] Y. Ma, Y. Wang, and V. Tresp. Quantum machine learning algorithm for knowledge graphs. *arXiv preprint arXiv:2001.01077*, 2020.

[44] M. W. Mahoney, M. Maggioni, and P. Drineas. Tensor-cur decompositions for tensor-based data. *SIAM Journal on Matrix Analysis and Applications*, 30(3):957–987, 2008.

[45] C. D. Martin, R. Shafer, and B. LaRue. An order-p tensor factorization with applications in imaging. *SIAM Journal on Scientific Computing*, 35(1):A474–A490, 2013.

[46] E. V. Mikheev, A. S. Pugin, D. A. Kuts, S. A. Podoshvedov, and N. B. An. Efficient production of large-size optical schrödinger cat states. *Scientific reports*, 9(1):1–15, 2019.

[47] R. Minster, A. K. Saibaba, and M. E. Kilmer. Randomized algorithms for low-rank tensor decompositions in the tucker format. *SIAM Journal on Mathematics of Data Science*, 2(1):189–215, 2020.

[48] W. J. Munro, K. Nemoto, G. J. Milburn, and S. L. Braunstein. Weak-force detection with superposed coherent states. *Physical Review A*, 66(2):023819, 2002.

[49] N. Namekata, Y. Takahashi, G. Fujii, D. Fukuda, S. Kurimura, and S. Inoue. Non-gaussian operation based on photon subtraction using a photon-number-resolving detector at a telecommunications wavelength. *Nature Photonics*, 4(9):655–660, 2010.

[50] J. S. Neergaard-Nielsen, Y. Eto, C.-W. Lee, H. Jeong, and M. Sasaki. Quantum tele-amplification with a continuous-variable superposition state. *Nature Photonics*, 7(6):439–443, 2013.

[51] J. S. Neergaard-Nielsen, B. M. Nielsen, C. Hettich, K. Mølmer, and E. S. Polzik. Generation of a superposition of odd photon number states for quantum information networks. *Physical review letters*, 97(8):083604, 2006.

[52] M. Nielsen and I. Chuang. *Quantum Computation and Information*. Cambridge University Press, London, 2010.

[53] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[54] A. Ourjoumtsev, H. Jeong, R. Tualle-Brouri, and P. Grangier. Generation of optical 'schrödinger cats' from photon number states. *Nature*, 448(7155):784–786, 2007.

[55] A. Ourjoumtsev, R. Tualle-Brouri, J. Laurat, and P. Grangier. Generating optical schrödinger kittens for quantum information processing. *Science*, 312(5770):83–86, 2006.

[56] J. Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.

[57] L. Qi, H. Chen, and Y. Chen. *Tensor Eigenvalues and Their Applications*, volume 39. Springer, 2018.

[58] L. Qi and Z. Luo. *Tensor Analysis: Spectral Theory and Special Tensors*, volume 151. Siam, 2017.

[59] L. Qi, G. Zhang, D. Braun, F. Bohnet-Waldraff, and O. Giraud. Regularly decomposable tensors and classical spin states. *Communications in mathematical sciences*, 2017.

[60] L. Qi, G. Zhang, and G. Ni. How entangled can a multi-party system possibly be? *Physics Letters A*, 382(22):1465–1471, 2018.

[61] D. Rafailidis and A. Nanopoulos. Modeling users preference dynamics and side information in recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(6):782–792, 2015.

[62] T. C. Ralph, A. Gilchrist, G. J. Milburn, W. J. Munro, and S. Glancy. Quantum computation with optical coherent states. *Physical Review A*, 68(4):042319, 2003.

[63] P. Rebentrost, M. Mohseni, and S. Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(13):130503, 2014.

[64] P. Rebentrost, M. Schuld, L. Wossnig, F. Petruccione, and S. Lloyd. Quantum gradient descent and Newton's method for constrained polynomial optimization. *New Journal of Physics*, 21(7):073023, 2019.

[65] P. Rebentrost, A. Steffens, I. Marvian, and S. Lloyd. Quantum singular-value decomposition of nonsparse low-rank matrices. *Physical review A*, 97(1):012327, 2018.

[66] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 727–736, 2009.

[67] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 81–90, 2010.

[68] S. M. Ross. *Introduction to Probability Models, ISE*. Academic press, 2006.

[69] N. Sangouard, C. Simon, N. Gisin, J. Laurat, R. Tualle-Brouri, and P. Grangier. Quantum repeaters with entangled coherent states. *JOSA B*, 27(6):A137–A145, 2010.

[70] R. Sarma and R. Jain. Quantum gate implementation of a novel reversible half adder and subtractor circuit. In *2018 International Conference on Intelligent Circuits and Systems (ICICS)*, pages 72–76. IEEE, 2018.

[71] T. Serikawa, J.-i. Yoshikawa, S. Takeda, H. Yonezawa, T. C. Ralph, E. H. Huntington, and A. Furusawa. Generation of a cat state in an optical sideband. *Physical review letters*, 121(14):143602, 2018.

[72] C. Shao. From linear combination of quantum states to grover's searching algorithm. *arXiv preprint arXiv:1807.09693*, 2018.

[73] C. Shao, Y. Li, and H. Li. Quantum algorithm design: Techniques and applications. *Journal of Systems Science and Complexity*, 32(1):375–452, 2019.

[74] J. Shen. On the singular values of gaussian random matrices. *Linear Algebra and its Applications*, 326(1-3):1–14, 2001.

[75] G. Song, M. K. Ng, and X. Zhang. Robust tensor completion using transformed tensor singular value decomposition. *Numerical Linear Algebra with Applications*, 27(3):e2299, 2020.

[76] D. V. Sychev, A. E. Ulanov, A. A. Pushkina, M. W. Richards, I. A. Fedorov, and A. I. Lvovsky. Enlargement of optical schrödinger's cat states. *Nature Photonics*, 11(6):379, 2017.

[77] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 43–50, 2008.

[78] K. Takase, J.-i. Yoshikawa, W. Asavanant, M. Endo, and A. Furusawa. Generation of optical schrödinger cat states by generalized photon subtraction. *Physical Review A*, 103(1):013710, 2021.

[79] E. Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 217–228, 2019.

[80] D. A. Tarzanagh and G. Michailidis. Fast randomized algorithms for t-product based tensor operations and decompositions with applications to imaging data. *SIAM Journal on Imaging Sciences*, 11(4):2629–2664, 2018.

[81] M. Teixeira and D. Rodriguez. A class of fast cyclic convolution algorithms based on block pseudocirculants. *IEEE Signal Processing Letters*, 2(5):92–94, 1995.

[82] H. Thapliyal and N. Ranganathan. Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 6(4):1–31, 2010.

[83] H. Thapliyal and N. Ranganathan. A new design of the reversible subtractor circuit. In *2011 11th IEEE International Conference on Nanotechnology*, pages 1430–1435. IEEE, 2011.

[84] L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[85] K. Wakui, H. Takahashi, A. Furusawa, and M. Sasaki. Photon subtracted squeezed states generated with periodically poled ktiopo 4. *Optics Express*, 15(6):3568–3574, 2007.

[86] C. Wang and L. Wossnig. A quantum algorithm for simulating non-sparse hamiltonians. *arXiv preprint arXiv:1803.08273*, 2018.

[87] M. Wu, S. He, Y. Zhang, J. Chen, Y. Sun, Y.-Y. Liu, J. Zhang, and H. V. Poor. A tensor-based framework for studying eigenvector multicentrality in multilayer networks. *Proceedings of the National Academy of Sciences*, 116(31):15407–15413, 2019.

[88] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with Bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM international conference on data mining*, pages 211–222. SIAM, 2010.

[89] G. Zhang. Dynamical analysis of quantum linear systems driven by multichannel multi-photon states. *Automatica*, 83:186–198, 2017.

[90] J. Zhang, A. K. Saibaba, M. E. Kilmer, and S. Aeron. A randomized tensor singular value decomposition based on the t-product. *Numerical Linear Algebra with Applications*, 25(5):e2179, 2018.

[91] M. Zhang, G. Ni, and G. Zhang. Iterative methods for computing u-eigenvalues of non-symmetric complex tensors with application in quantum entanglement. *Computational Optimization and Applications*, 75:779–798, 2020.

[92] Z. Zhang and S. Aeron. Exact tensor completion using t-svd. *IEEE Transactions on Signal Processing*, 65(6):1511–1526, 2016.

[93] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-svd. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3842–3849, 2014.

[94] G. Zhou, A. Cichocki, and S. Xie. Decomposition of big tensors with low multilinear rank. *arXiv preprint arXiv:1412.1885*, 2014.

[95] P. Zhou, C. Lu, Z. Lin, and C. Zhang. Tensor factorization for low-rank tensor completion. *IEEE Transactions on Image Processing*, 27(3):1152–1163, 2017.

[96] X.-Q. Zhou, T. C. Ralph, P. Kalasuwan, M. Zhang, A. Peruzzo, B. P. Lanyon, and J. L. O'Brien. Adding control to arbitrary unknown quantum operations. *Nature Communications*, 2(1):1–8, 2011.