



OPEN ACCESS

EDITED BY

Cornelio Yáñez-Márquez,
National Polytechnic Institute (IPN), Mexico

REVIEWED BY

Mailyn Moreno,
Centro de Investigación en Computación,
Instituto Politécnico Nacional (CIC-IPN),
Mexico
Gaohang Yu,
Hangzhou Dianzi University, China

*CORRESPONDENCE

Yunzhi Jin
✉ yzjin@ynu.edu.cn

RECEIVED 05 February 2024

ACCEPTED 30 April 2024

PUBLISHED 02 July 2024

CITATION

Wu Y and Jin Y (2024) Efficient enhancement of low-rank tensor completion via thin QR decomposition.
Front. Big Data 7:1382144.
doi: 10.3389/fdata.2024.1382144

COPYRIGHT

© 2024 Wu and Jin. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Efficient enhancement of low-rank tensor completion via thin QR decomposition

Yan Wu and Yunzhi Jin*

Yunnan Key Laboratory of Statistical Modeling and Data Analysis, Yunnan University, Kunming, China

Low-rank tensor completion (LRTC), which aims to complete missing entries from tensors with partially observed terms by utilizing the low-rank structure of tensors, has been widely used in various real-world issues. The core tensor nuclear norm minimization (CTNM) method based on Tucker decomposition is one of common LRTC methods. However, the CTNM methods based on Tucker decomposition often have a large computing cost due to the fact that the general factor matrix solving technique involves multiple singular value decompositions (SVDs) in each loop. To address this problem, this article enhances the method and proposes an effective CTNM method based on thin QR decomposition (CTNM-QR) with lower computing complexity. The proposed method extends the CTNM by introducing tensor versions of the auxiliary variables instead of matrices, while using the thin QR decomposition to solve the factor matrix rather than the SVD, which can save the computational complexity and improve the tensor completion accuracy. In addition, the CTNM-QR method's convergence and complexity are analyzed further. Numerous experiments in synthetic data, real color images, and brain MRI data at different missing rates demonstrate that the proposed method not only outperforms in terms of completion accuracy and visualization, but also conducts more efficiently than most state-of-the-art LRTC methods.

KEYWORDS

auxiliary variable tensor, tensor nuclear norm minimization, thin QR decomposition, Tucker decomposition, Tucker rank

1 Introduction

With the rapid development of human needs and Internet technology, the scale of data acquired by people has expanded. Tensor data, such as third-order color images, fourth-order video sequences, and hyperspectral images, are now ubiquitous. At present, tensors are widely used in machine learning (Bai et al., 2021; Panagakis et al., 2021), computer vision (Kajo et al., 2019), image processing (Miao et al., 2020), etc., and are better at expressing the complex internal structure of higher-order data than vectors and matrices. However, due to data transmission and storage limitations, practical applications often face missing or corrupted observation tensor entries, which directly affect the quality of tensor analysis. Consequently, utilizing the observation part for missing tensor completion is a prominent research topic.

Conventional methods of completion involve matrixization or vectorization of the tensor, which leads to dimensionality catastrophe in addition to destroying the spatial structure of high-order data. The aim of low-rank tensor completion (LRTC),

a multidimensional expansion based on low-rank matrix completion (LRMC) (Candes and Recht, 2012; Xu et al., 2021), is to estimate missing terms (Zhou et al., 2018) by utilizing spatial correlation between tensor terms. LRTC is crucial for tensor processing and analysis. Several researches (Kolda and Bader, 2009; Liu et al., 2013; Zhao et al., 2020) have demonstrated that there is a significant quantity of redundant information in natural tensor data, which is typically low-rank or virtually low-rank, such as photos and videos. Figure 1 displays the distribution of singular values for the three channels of the color image. It can be observed that the majority information of the image is stored in only a few of the bigger singular values, its low rank structure is notable. This tensor property encourages the development of low-rank tensor estimation and completion problems, and it has been proactively applied in real-world problems like multi-channel image and video completion (Liu et al., 2013; Su et al., 2022), audio source separation (Yoshii et al., 2018), etc.

In the two-dimensional case, the rank is a strong tool for capturing global information about the data, but the definition of the tensor rank is not as explicit as that of the matrix, and the fundamental issue is that the definition of the tensor rank is not unique. The CANDECOMP/PARAFAC (CP) tensor rank is defined as the minimum number of rank-1 tensors needed in the CP decomposition. Nevertheless, CP rank determination is an NP-hard problem (Hillar and Lim, 2013), and there isn't always an optimal low CP rank approximation, which may result in a poor fit in real-world applications (Qiu et al., 2022). Another is the Tucker rank derived from the Tucker decomposition, defined as a vector composed of the ranks of mode unfolding matrices, which is extensively employed in LRTC driven by its superior computability. In recent years, tensor train (TT) rank (Oseledets, 2011), tubal rank (Zhang and Aeron, 2016; Zhou et al., 2018), tensor ring (TR) rank (Yuan et al., 2019a), tensor tree rank (Liu et al., 2019b), etc. have also been successively proposed to provide an effective way to deal with the LRTC problem.

Owing to the nonconvex nature of the rank function, contemporary LRTC approaches predominantly employ the tensor nuclear norm as a convex substitute for the rank function. Liu et al. (2013) pioneered the definition of the tensor nuclear norm, modeled the LRTC as a convex optimization problem, and suggested three efficient algorithms to solve the LRTC problem. Shang et al. (2017) introduced a novel method for fuzzy dual nuclear norm minimization. In a related development, Bengua et al. (2017) leveraging the TT rank, proposed the SiLRTC-TT and TMac-TT models. These algorithms have significantly contributed to addressing the LRTC problem. Nevertheless, the majority of algorithms relying on tensor nuclear norm minimization (TNNM) encounter challenges associated with the elevated computational expense incurred by multiple SVDs. In addressing this issue, Liu et al. (2014a) formulates a novel tensor completion approach by imposing nuclear norm constraints on the factor matrices of CP decomposition. Shi et al. (2017) presented a tensor rank estimation approach based on l_1 -regularized orthogonal CP decomposition. Liu (2013) builds upon the TNNM, introduces the Tucker decomposition, and proposes a Core Tensor Nuclear Norm Minimization (CTNM) completion model.

While the CTNM model reduces the computational burden to a certain degree, the introduction of the Tucker decomposition inevitably involves the iterative solution of the factor matrix. A higher-order expansion of the matrix SVD to the tensor, the Higher Order SVD (HOSVD) (De Lathauwer et al., 2000a; Chao et al., 2021) was first proposed. De Lathauwer et al. (2000b) introduced the Higher-Order Orthogonal Iteration (HOOI) approach, which initializes the HOSVD solution, to boost approximation accuracy. These factor matrix solution techniques also involve multiple SVDs of tensor matricizations in the loop iterations, which surely places some computing strain on the CTNM model and contributes to its still-relatively-slow convergence rate. Building upon the insights from Shi et al. (2015), we adopt the thin QR decomposition, as opposed to HOOI, to address the factor matrix. We propose an efficient CTNM-QR model that circumvents the necessity for SVD in factor matrix resolution, thereby enhancing the algorithm's computational productivity. The principal contributions of this article are as follows:

- (1) An enhanced CTNM-QR model is proposed, via introducing tensor versions of the auxiliary variables instead of matrices, while using the thin QR decomposition to solve the factor matrix instead of HOOI, which avoids the computation of multiple SVDs in each loop and minimizes the computational cost to a certain extent, thus speeding up the process.
- (2) The complexity and convergence performance of the proposed method are analyzed. Compared with the state-of-the-art LRTC methods, the proposed method saves computational complexity, and is consistent with the generalized higher-order orthogonal iteration (gHOI) algorithm (Liu et al., 2014b) in terms of the algorithm's primary computational cost.
- (3) The CTNM-QR model is applied to synthetic tensors, color images, and medical MRI scans for completion, validating the accuracy and efficacy of the enhanced algorithm.

The rest of the paper is organized as follows. We give the tensor notation and review some related work in Section 2. In Section 3, the proposed CTNM-QR method is described in detail, and the convergence and complexity of the CTNM-QR method are analyzed. In Section 4, we offer experimental analysis and completion results to validate the proposed algorithm. Finally, the paper is summarized in Section 5.

2 Notations and background

In this section, we introduce some notation and background of the tensor and tensor completion. Throughout this article, we use lowercase letters to denote vectors, e.g., a, b , uppercase letters for matrices, e.g., A, B , and uppercase fancy letters for tensors of order three and above, e.g., \mathcal{A}, \mathcal{X} . The i th element of a vector a is denoted as a_i , the element of a matrix A with index (i, j) is denoted as a_{ij} , and the elements of a N -order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ are denoted as x_{i_1, i_2, \dots, i_N} .

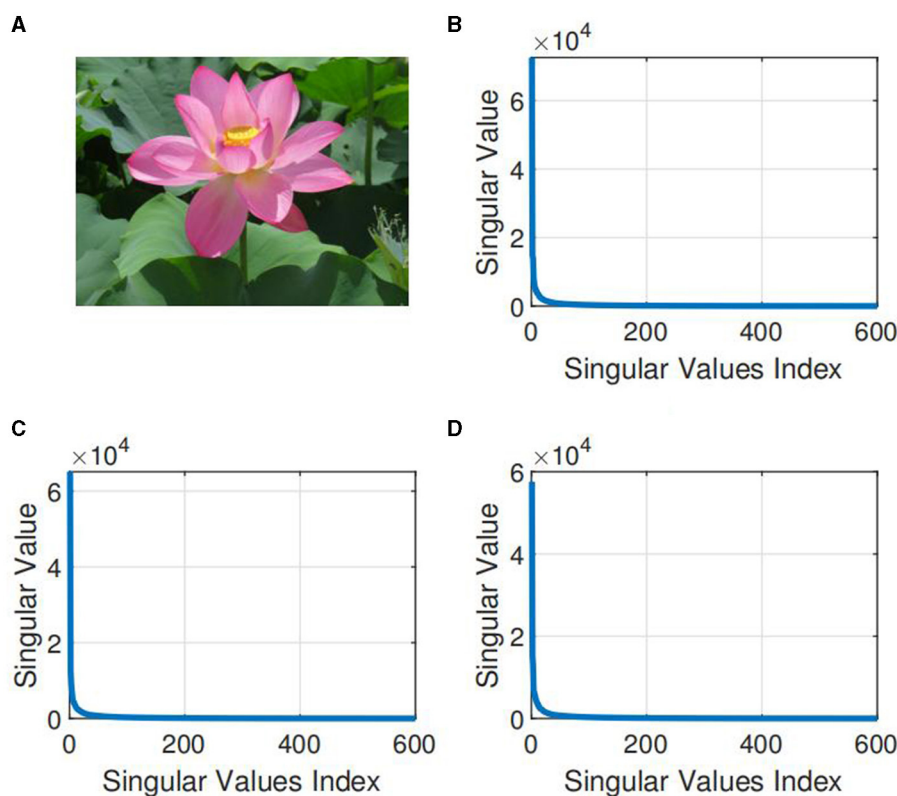


FIGURE 1 Illustration of the low-rank property of natural color image. **(A)** A three-dimensional low rank color image \mathcal{X} with dimensions $800 \times 600 \times 3$ (taken from <https://www.pexels.com/jacquemay dominique>). **(B)** Singular value distribution of the first channel $\mathcal{X}(:, :, 1)$. **(C)** Singular value distribution of the second channel $\mathcal{X}(:, :, 2)$. **(D)** Singular value distribution of the third channel $\mathcal{X}(:, :, 3)$.

2.1 Tensor concepts and terminology

In this subsection, we first introduce some fundamental tensor concepts and terminology. An extensive review of tensors are discussed in *Kolda and Bader (2009)*.

For the definition of a tensor, there are several definitions at different abstraction levels (*Boumal, 2023*). Here, an N th-order tensor is defined as a multidimensional array with N dimensions ($N > 2$), which is an extension of a matrix to higher order. Here, N represents the order of the tensor and the number of dimensions of the tensor, also known as modes. Analogous to the rows and columns of a matrix, we can create subarrays by fixing some of the given tensor's indices. A fiber of a tensor is a vector created by fixing all but one index of a particular mode (*Filipović and Jukić, 2015*). A third-order tensor \mathcal{X} has row, column and tube fibers, denoted by $x(:, j, k)$, $x(i, :, k)$, $x(i, j, :)$, respectively. A slice of a tensor is defined as a two-dimensional matrix, obtained by fixing all but the indices of two particular modes, e.g., a third-order tensor with horizontal, lateral, and frontal slices, denoted by $x(i, :, :)$, $x(:, j, :)$, and $x(:, :, k)$, respectively.

We can also transform a tensor into a matrix via reordering the elements of an N th-order tensor, known as matricization, unfolding or flattening. The mode- n unfolding of an N th-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is obtained by keeping the n th mode fixed and concatenating the slices of the rest of the modes into one

long matrix, denoted by $\mathcal{X}_{(n)}$, which is an $I_n \times \prod_{j \neq n} I_j$ matrix for $n = 1, 2, \dots, N$, where the symbol \prod is continued product. The tensor element (i_1, i_2, \dots, i_N) is mapped into the element (i_n, j) of the matrix $\mathcal{X}_{(n)}$, where

$$j = 1 + \sum_{k=1, k \neq n}^N (i_k - 1) J_k, J_k = \prod_{m=1, m \neq n}^{k-1} I_m.$$

The mode- n product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ with a matrix $U \in \mathbb{R}^{J \times I_n}$ is denoted as $\mathcal{X} \times_n U \in \mathbb{R}^{I_1 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N}$. If there exists a series of distinct modes, the order of multiplication in the mode- n product is independent, that is,

$$\mathcal{X} \times_m A \times_n B = \mathcal{X} \times_n B \times_m A (m \neq n).$$

According to the mode- n unfolding of the tensor, we have the following formula

$$\mathcal{Y} = \mathcal{X} \times_n U \Leftrightarrow \mathcal{Y}_{(n)} = U \mathcal{X}_{(n)}.$$

The inner product of two tensors $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ and $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ with the same size is denoted as a sum of the products of the corresponding elements, that is,

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N} y_{i_1 i_2 \dots i_N}.$$

By an extension of the matrix property, the Frobenius norm of the tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is defined as the square root of the sum of the squares of all elements:

$$\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle} = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N}^2}$$

The distance between the tensor \mathcal{X} and \mathcal{Y} is defined as $\|\mathcal{X} - \mathcal{Y}\|_F$.

The Kronecker product of two matrices $A \in \mathbb{R}^{I \times J}$ and $B \in \mathbb{R}^{K \times L}$ is denoted by $A \otimes B$, and yields a matrix of size $(IK) \times (JL)$, defined as:

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1J}B \\ a_{21}B & a_{22}B & \dots & a_{2J}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1}B & a_{i2}B & \dots & a_{iJ}B \end{bmatrix}$$

2.2 Tensor decomposition

Tensor decomposition refers to expressing a tensor as a sequence of elementary operations on other simple arrays. There are many tensor decompositions. We introduce two major tensor decompositions, including CANDECOMP/PARAFAC (CP) decomposition (Cattell, 1944) and Tucker decomposition (Tucker, 1963).

For a given tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, the CP decomposition is defined as the factorization of \mathcal{A} into the minimum number of linear combinations of rank-1 tensors:

$$\mathcal{A} = \sum_{r=1}^R \lambda_r \mathbf{a}_r^{(1)} \circ \dots \circ \mathbf{a}_r^{(N)},$$

where “ \circ ” denotes the outer product of vectors, and $\mathbf{a}_r^{(k)} \in \mathbb{R}^{I_k}$ ($k = 1, 2, \dots, N$) is the factor vector.

The Tucker decomposition is based on an extension of the CP decomposition, a form of higher-order principal component analysis. The Tucker decomposition of the tensor \mathcal{A} is expressed as:

$$\mathcal{A} \approx \mathcal{C} \times_1 U_1 \times \dots \times_N U_N,$$

where $\mathcal{C} \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_N}$ is the core tensor and denotes the degree of interaction between the different components, and $U_n \in \mathbb{R}^{I_n \times R_n}$ is the n th-mode factor matrix, denoting the principal components on each mode, usually orthogonal. In general, R_n is much smaller than I_n for $n = 1, \dots, N$, so the core tensor \mathcal{C} can be considered as a compressed version of the original tensor \mathcal{A} , which reduces the computational complexity to some extent.

The matricized form of the Tucker decomposition is as follows:

$$\mathcal{A}_{(n)} = U_n \mathcal{C}_{(n)} (U_{N+1} \otimes \dots \otimes U_{n+1} \otimes U_{n-1} \otimes \dots \otimes U_1)^T.$$

The Tucker-rank of the N th-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, also known as the multilinear rank (Kasai and Mishra, 2016), is

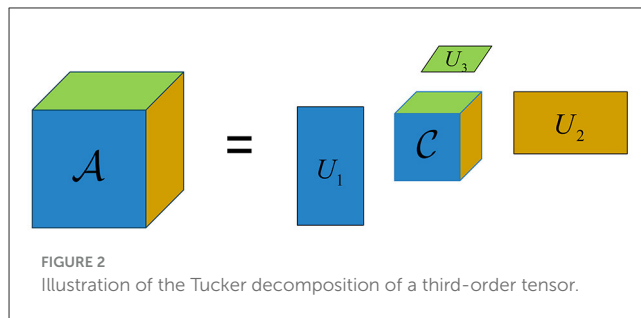


FIGURE 2 Illustration of the Tucker decomposition of a third-order tensor.

denoted as an N -dimensional vector composed of the ranks of the various mode unfolding matrices of the tensor:

$$rank_{tc}(\mathcal{A}) = (rank(\mathcal{A}_{(1)}), rank(\mathcal{A}_{(2)}), \dots, rank(\mathcal{A}_{(N)})) \quad (1)$$

where $rank(\mathcal{A}_{(n)})$ is the rank of the mode- n unfolding matrices, representing the dimension of the vector space spanned by the mode- n fiber. In contrast to the CP rank, the Tucker-rank can be easily obtained by computation, while the CP rank can only be determined empirically or by multiple experimental comparisons to search for the optimal value (Mu et al., 2014). Diagram of the Tucker decomposition of a third-order tensor is shown in Figure 2.

2.3 Low-rank tensor completion

The LRTC aims to infer and fill missing entries of the tensor from partially observed values via utilizing the low-rank structure of the tensor, which is an extension of the low-rank matrix completion.

Let $M \in \mathbb{R}^{p \times q}$ be the observed incomplete matrix and Ω be the index set, i.e., the observed positions of the matrix elements. Recovering the matrix M based on the observed partial elements is referred to as low-rank matrix completion, that is,

$$\min_X rank(X), \text{ s.t. } X_{\Omega} = M_{\Omega}, \quad (2)$$

where the complete matrix $X \in \mathbb{R}^{p \times q}$ is obtained via the above optimization problem with constraints.

As a broadening of the low-rank matrix completion (Equation 2), the LRTC (Ji et al., 2009; Liu et al., 2013; Yuan et al., 2019a) is formulated as the following optimization problem:

$$\min_{\mathcal{X}} rank(\mathcal{X}), \text{ s.t. } \mathcal{X}_{\Omega} = \mathcal{T}_{\Omega},$$

where $\mathcal{X}, \mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, \mathcal{T} is the incomplete tensor containing the missing elements, and \mathcal{X} is the completed tensor. The optimization problem is built with a rank minimization constraint, which ensures that the output tensor has the lowest rank possible for tensor completion. In the LRTC, there is also a category of optimization based on tensor decomposition, formulated as minimizing the discrepancy between the real observations and the predicted values to complete the missing entries under the given tensor rank. That is,

$$\min_{\mathcal{X}} \|\mathcal{X}_{\Omega} - \mathcal{T}_{\Omega}\|_F^2, \text{ s.t. } rank(\mathcal{X}) = R,$$

where R is a bounded constraint on the rank of the \mathcal{X} and the tensor \mathcal{X} fulfills a specific tensor decomposition model. The traditional models are the completion models based on the CP and Tucker decompositions, e.g., the optimization model for tensor completion using CP decomposition is stated as follows:

$$\min_{\mathcal{X}} \|\mathcal{X}_{\Omega} - \mathcal{T}_{\Omega}\|_F^2, \text{ s.t. } \mathcal{X} = \sum_{r=1}^R \lambda_r \mathbf{a}_r^{(1)} \circ \dots \circ \mathbf{a}_r^{(N)}.$$

The LRTC algorithm minimizing tensor rank is more widely used because the low-rank decomposition-based completion algorithm is impractical in some situations in reality, especially it is difficult to select an appropriate tensor rank for use in the algorithm when there are few observed elements. However, since the tensor rank is not unique, this makes the LRTC algorithm very challenging (Liu et al., 2013). Besides the CP rank and the Tucker rank in Equation (1), there are various forms of tensor rank, such as TT rank, tubal rank, TR rank, etc. Selecting one of these forms will result in different LRTC models.

2.4 Tensor nuclear norm minimization

In the LRTC, since the computation of the tensor CP rank is an NP-hard problem (Hillar and Lim, 2013) and the Tucker rank has the excellent property of being easy to compute, optimal completion models based on the Tucker rank of the tensor are widely used (Romera-Paredes and Pontil, 2013; Liu et al., 2019a; Yu et al., 2023). That is,

$$\min_{\mathcal{X}} \sum_{i=1}^N \alpha_i \text{rank}(\mathcal{X}_{(i)}), \text{ s.t. } \mathcal{X}_{\Omega} = \mathcal{T}_{\Omega}, \tag{3}$$

where the parameters α_i are pre-specified weights and satisfy $\alpha_i \geq 0, \sum \alpha_i = 1$. The Tucker rank of the tensor is represented in the form of a weighted sum. Gandy et al. (2011) suggests the unweighted model of $\alpha_i = 1 (i = 1, 2, \dots, N)$.

Since the rank of the matrix is non-convex, it is difficult to guarantee to finding a globally optimal solution in the optimization (3). Since the nuclear norm of the matrix is shown to be the tightest convex envelop of the matrix rank (Candes and Recht, 2012), the completion model based on the nuclear norm minimization is proposed as follows:

$$\min_{\mathcal{X}} \sum_{i=1}^N \alpha_i \|\mathcal{X}_{(i)}\|_*, \text{ s.t. } \mathcal{X}_{\Omega} = \mathcal{T}_{\Omega}, \tag{4}$$

where the nuclear norm $\|\mathcal{X}_{(i)}\|_* = \sum_j \sigma_j(\mathcal{X}_{(i)})$, and $\sigma_j(\mathcal{X}_{(i)})$ denotes the j -th largest singular value of the matricization $\mathcal{X}_{(i)}$. The $\sum_{i=1}^N \alpha_i \|\mathcal{X}_{(i)}\|_*$ is defined as the nuclear norm of the tensor \mathcal{X} , denoted as $\|\mathcal{X}\|_*$. The tensor nuclear norm is also known as the tensor Schatten-q norm (Ji et al., 2009; Signoretto et al., 2010). Based on the nuclear norm, the concept of low-rank matrix completion is generalized to high-dimensional data, where the low-rank property of the tensor is often used as a

necessary assumption to limit the degrees of freedom of missing entries (Kressner et al., 2014).

The tensor nuclear norm minimization completion model (TNNM) (4) can be expressed as the following:

$$\min_{\mathcal{X}} \sum_{i=1}^N \alpha_i \|\mathcal{X}_{(i)}\|_* + \frac{\lambda}{2} \|\mathcal{X}_{\Omega} - \mathcal{T}_{\Omega}\|_F^2, \tag{5}$$

where $\lambda > 0$ is a regularization parameter.

Since the matrix nuclear norm in the model (5) is interdependent and cannot be optimized independently, auxiliary variables for splitting need to be introduced for simplifying solution algorithm. Liu et al. (2013) presented three solution algorithms, which are simple LRTC, fast LRTC, and high-accuracy LRTC. Based on tensor decomposition, Liu (2013); Liu et al. (2014a) proposed completion models for the factor matrix nuclear norm minimization and the core tensor nuclear norm minimization. Nevertheless, the majority of TNNM algorithms involve SVDs of several large matrices in each iteration, resulting in high computing costs.

3 The proposed method

3.1 Core tensor nuclear norm minimization based on thin QR decomposition

In the subsection, we describe the proposed CTNM-QR algorithm and the solution procedure in detail. Specifically, we adopt the ideas of the thin QR decomposition to update the factor matrix on the framework of the core tensor nuclear norm minimization (CTNM) and propose an efficient low-rank tensor completion algorithm.

In the CTNM model based on the Tucker decomposition (Liu, 2013; Liu et al., 2014b), the following assumption is stated:

Assumption 1. Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ with $\text{rank}_{tc} = (R_1, R_2, \dots, R_N)$ and satisfy Tucker decomposition $\mathcal{X} = \mathcal{C} \times_1 U_1 \times_2 U_2 \times \dots \times_N U_N$, where the core tensor $\mathcal{C} \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_N}$, and the orthogonal factor matrix $U_i \in \mathbb{R}^{I_i \times R_i}$. Then

$$\|\mathcal{X}\|_* = \|\mathcal{C}\|_*$$

where $\|\mathcal{X}\|_*$ and $\|\mathcal{C}\|_*$, denotes the nuclear norm of the original tensor and the core tensor, respectively.

According to the Assumption 1, the completion model for the CTNM is:

$$\min_{\mathcal{C}, \{U_i\}, \mathcal{X}} \sum_{i=1}^N \alpha_i \|\mathcal{C}_{(i)}\|_* + \frac{\lambda}{2} \|\mathcal{X} - \mathcal{C} \times_1 U_1 \times \dots \times_N U_N\|_F^2, \tag{6}$$

s.t., $\mathcal{X}_{\Omega} = \mathcal{T}_{\Omega}$, and $(U_i)^T U_i = I_i, i = 1, 2, \dots, N$,

where I_i is an $I_i \times I_i$ Identity matrix. Owing to the interdependence of the unfolding matrices, we introduce auxiliary tensors $\mathcal{V}_i \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_N} (i = 1, 2, \dots, N)$ to facilitate variable separation.

Then, the model (6) is transformed into the following equivalent form:

$$\min_{\mathcal{C}, \mathcal{V}_i, U_i, \mathcal{X}} \sum_{i=1}^N \alpha_i \|\mathcal{V}_{i(i)}\|_* + \frac{\lambda}{2} \|\mathcal{X} - \mathcal{C} \times_1 U_1 \times \cdots \times_N U_N\|_F^2, \quad (7)$$

s.t., $\mathcal{X}_\Omega = \mathcal{T}_\Omega, \mathcal{C} = \mathcal{V}_i$, and $U_i^T U_i = I_i, i = 1, \dots, N$.

For the optimization problem Equation (7), we propose an efficient CTNM-QR algorithm based on alternating direction multiplication method (ADMM) to address it. The ADMM is a commonly employed optimization method that decomposes a problem into a sequence of sub-problems to calculate the optimal solution, and is adept at proficiently addressing optimization problems featuring multiple non-smooth terms in the objective function (Glowinski, 2014; Boţ and Nguyen, 2020; Han, 2022). The partial augmented Lagrangian function for the model (7) is as follows:

$$\begin{aligned} \mathcal{L}(\mathcal{X}, \mathcal{C}, U_1, \dots, U_N, \mathcal{V}_1, \dots, \mathcal{V}_N, \mathcal{Y}_1, \dots, \mathcal{Y}_N) \\ = \sum_{i=1}^N \left(\alpha_i \|\mathcal{V}_{i(i)}\|_* + \langle \mathcal{Y}_i, \mathcal{C} - \mathcal{V}_i \rangle + \frac{\mu}{2} \|\mathcal{C} - \mathcal{V}_i\|_F^2 \right) \\ + \frac{\lambda}{2} \|\mathcal{X} - \mathcal{C} \times_1 U_1 \times \cdots \times_N U_N\|_F^2, \end{aligned} \quad (8)$$

where $\mathcal{Y}_i(i = 1, \dots, N)$ is the Lagrange multiplier tensor and $\mu > 0$ is the penalty parameter.

The ADMM converts the original tensor completion problem (7) into solving the five subproblems of (9a–9e) by minimizing the augmented Lagrangian function (8), which iteratively updates the corresponding parameters. Specifically, we update $\mathcal{V}_i^{k+1}, \mathcal{C}^{k+1}, U_i^{k+1}, \mathcal{X}^{k+1}$ and \mathcal{Y}_i^{k+1} sequentially in the order of (9a–9e) with the fixed other parameters. The steps of the algorithm solution are as follows:

$$\mathcal{V}_i^{k+1} = \arg \min_{\mathcal{V}_i} \mathcal{L}(\mathcal{X}^k, \mathcal{C}^k, \{U_j^k, \mathcal{Y}_j^k\}_{j=1}^N, \{\mathcal{V}_j^{k+1}\}_{j=1}^{i-1}, \{\mathcal{V}_j^k\}_{j=i+1}^N), \quad (9a)$$

$$\mathcal{C}^{k+1} = \arg \min_{\mathcal{C}} \mathcal{L}(\mathcal{X}^k, \mathcal{C}, \{U_j^k, \mathcal{V}_j^{k+1}, \mathcal{Y}_j^k\}_{j=1}^N), \quad (9b)$$

$$U_i^{k+1} = \arg \min_{U_i} \mathcal{L}(\mathcal{X}^k, \mathcal{C}^{k+1}, \{U_j^{k+1}\}_{j=1}^{i-1}, \{U_j^k\}_{j=i+1}^N, \{\mathcal{V}_j^{k+1}, \mathcal{Y}_j^k\}_{j=1}^N), \text{ s.t. } U_i^T U_i = I, \quad (9c)$$

$$\mathcal{X}^{k+1} = \arg \min_{\mathcal{X}} \mathcal{L}(\mathcal{X}, \mathcal{C}^{k+1}, \{U_j^{k+1}, \mathcal{V}_j^{k+1}, \mathcal{Y}_j^k\}_{j=1}^N), \text{ s.t. } \mathcal{X}_\Omega = \mathcal{T}_\Omega, \quad (9d)$$

$$\mathcal{Y}_i^{k+1} = \mathcal{Y}_i^k + \mu^k (\mathcal{C}^{k+1} - \mathcal{V}_i^{k+1}). \quad (9e)$$

That is, for \mathcal{V}_i^{k+1} in the $k + 1$ th iteration, (9a) can be written as the following optimization subproblem:

$$\mathcal{V}_i^{k+1} = \arg \min_{\mathcal{V}_i} \frac{\alpha_i}{\mu^k} \|\mathcal{V}_{i(i)}\|_* + \frac{1}{2} \left\| \mathcal{C}^k - \mathcal{V}_i + \frac{\mathcal{Y}_i^k}{\mu^k} \right\|_F^2 \text{ for } i = 1, 2, \dots, N. \quad (10)$$

The iteration for \mathcal{V}_i^{k+1} for $i = 1, 2, \dots, N$ can be effectively addressed through the application of the singular value

thresholding (SVT) operator (Cai et al., 2010). It is easy to obtain a closed solution to the problem (10) as:

$$\mathcal{V}_i^{k+1} = \text{fold}_i(\text{SVT}_{\alpha_i/\mu^k}(\mathcal{C}^k + \frac{\mathcal{Y}_i^k}{\mu^k})_{(i)}), \quad (11)$$

where fold_i denotes the transformation of the matrix into a tensor along the i th mode, if the SVD of a matrix Z is $Z = U \text{diag}(\sigma) V^T$, $\text{SVT}_\tau(Z) = U \text{diag}(\max\{\sigma - \tau, 0\}) V^T$, $\max(\cdot, \cdot)$ is the element-based maximization operator.

For \mathcal{C}^{k+1} in the $k + 1$ th iteration, the optimization problem (9b) with respect to \mathcal{C} can be rewritten as:

$$\begin{aligned} \mathcal{C}^{k+1} = \arg \min_{\mathcal{C}} \sum_{i=1}^N \frac{\mu^k}{2} \left\| \mathcal{C} - \mathcal{V}_i^{k+1} + \frac{\mathcal{Y}_i^k}{\mu^k} \right\|_F^2 \\ + \frac{\lambda}{2} \left\| \mathcal{X}^k - \mathcal{C} \times_1 U_1^k \times \cdots \times_N U_N^k \right\|_F^2. \end{aligned} \quad (12)$$

With other variables fixed, it is easy to see that Equation (12) is a smooth and differentiable optimization problem that can be derived by first-order optimality conditions for \mathcal{C} (Liu, 2013; Liu et al., 2014b).

For \mathcal{X}^{k+1} in the $k + 1$ th iteration, solving \mathcal{X} by the following subproblem:

$$\begin{aligned} \mathcal{X}^{k+1} = \arg \min_{\mathcal{X}} \left\| \mathcal{X} - \mathcal{C}^{k+1} \times_1 U_1^{k+1} \times \cdots \times_N U_N^{k+1} \right\|_F^2 \\ , \text{ s.t. } \mathcal{X}_\Omega = \mathcal{T}_\Omega. \end{aligned} \quad (13)$$

By deriving the Karush-Kuhn-Tucker (KKT) condition for Equation (13), it is easy to obtain the update form of \mathcal{X} as $\mathcal{X}_\Omega^{k+1} = \mathcal{T}_\Omega$ and $\mathcal{X}_{\Omega^c}^{k+1} = \left(\mathcal{C}^{k+1} \times_1 U_1^{k+1} \times \cdots \times_N U_N^{k+1} \right)_{\Omega^c}$, where Ω is the index set of the observed elements, Ω^c is the complement of Ω .

For the factor matrix $U_i^{k+1} \in \mathbb{R}^{I_i \times R_i} (i = 1, 2, \dots, N)$ in the $k + 1$ th iteration, the optimization problem (9c) is equivalent to the following optimization problem:

$$\begin{aligned} U_i^{k+1} = \arg \min \left\| \mathcal{X}^k - \mathcal{C}^{k+1} \times_1 U_1^{k+1} \cdots \times_{i-1} U_{i-1}^{k+1} \right. \\ \left. \times_{i+1} U_{i+1}^k \cdots \times_N U_N^k \right\|_F^2 \text{ s.t. } (U_i)^T U_i = I_i, \end{aligned} \quad (14)$$

This is a peculiar Tucker decomposition model in which $U_i, i = 1, 2, \dots, N$ is an orthogonal matrix. HOSVD (De Lathauwer et al., 2000a; Chao et al., 2021) and HOOI (De Lathauwer et al., 2000b) are frequently used in the computation of Equation (14). Nevertheless, these approaches need alternating iterations of solving Equation (14) and involve SVDs of multiple tensor matricizations in each loop iteration, which causes the algorithms to converge quite slowly. To enhance the approaches, we adopt thin QR decomposition instead of HOOI for solving U_i , which essentially avoid the SVDs during factorization and improve the computing efficiency.

Specifically, we first isolate the i th factor matrix U_i from the Tucker decomposition since the order of multiplications in the tensor mode- i product is not relevant. Then the equivalent form of the optimization model (14) for the solution of the i th factor matrix U_i is as follows:

$$\min_{U_i} \frac{1}{2} \left\| \mathcal{Z}_i^k \times_i U_i - \mathcal{X}^k \right\|_F^2, \text{ s.t. } (U_i)^T U_i = I_i, \quad (15)$$

where the given tensor $\mathcal{Z}_i^k = \mathcal{C}^{k+1} \times_1 U_1^{k+1} \times_2 \cdots \times_{i-1} U_{i-1}^{k+1} \times_{i+1} U_{i+1}^k \cdots \times_N U_N^k$. The problem (15) is nonconvex for all variables, yet it is convex for each block tensor. When the variables $\mathcal{C}, \mathcal{X}, \{U_j\}_{j=1, j \neq i}^N$ are fixed, the optimization (15) with respect to the factor matrix U_n is a smooth and differentiable optimization problem and can be solved via the first-order optimality condition. Hence the following equation holds:

$$\left(\mathcal{Z}_i^k \times_i U_i - \mathcal{X}^k \right)_{(i)} \mathcal{Z}_{i(i)}^{kT} = 0,$$

where $\left(\mathcal{Z}_i^k \times_i U_i - \mathcal{X}^k \right)_{(i)}$ and $\mathcal{Z}_{i(i)}^k$ are the mode- i unfolding matrices of the tensors $\mathcal{Z}_i^k \times_i U_i - \mathcal{X}^k$ and \mathcal{Z}_i^k , respectively. Then the least-squares solution without taking into account the orthogonality constraints on the factor matrices can be obtained by

$$\tilde{U}_i^{k+1} = \mathcal{X}_{(i)}^k \mathcal{Z}_{i(i)}^{kT} \left(\mathcal{Z}_{i(i)}^k \mathcal{Z}_{i(i)}^{kT} \right)^{-1}.$$

Motivated by the work of Shi et al. (2015), we adopt the thin QR decomposition for the orthogonality constraints of the factor matrices, that is,

$$U_i^{k+1} = QR \left(\tilde{U}_i^{k+1} \right), \text{ for } i = 1, 2, \dots, N, \quad (16)$$

where $QR(\cdot)$ denotes the thin QR decomposition of the matrix. At this step, the proposed CTNM-QR algorithm merely needs to carry out QR decomposition of a matrix with the computational complexity $\mathcal{O} \left(\sum_{i=1}^N I_i \times R_i^2 \right)$. While the computational complexity of the HOSVD algorithm in terms of U_i iterations is $\mathcal{O} \left(\sum_{i=1}^N I_i^2 \times \prod_{j \neq i} I_j \right)$, the HOOI algorithm is $\mathcal{O} \left(\sum_{i=1}^N I_i^2 \times \prod_{j \neq i} R_j \right)$, where $R_i \ll I_i$. This indicates that the significant computational complexity associated with the SVD can be effectively mitigated by employing QR decomposition to resolve the factor matrix.

3.2 Implementation

In the following, we discuss several implementations issues. To effectively solve the objective function (8), the penalty parameter μ is adaptively varied via setting the initialization of μ to μ_0 , and iteratively increasing μ^k by $\mu^{k+1} = \rho \mu^k$, where $\rho \in (1.0, 1.1]$ and μ_0 is usually a smaller constant. For the iteration among the problems (9a–9e), we principally use two convergence criteria. One is the relative change satisfaction between two neighboring recovery tensors \mathcal{X}^{k+1} and \mathcal{X}^k :

$$\frac{\left\| \mathcal{X}^{k+1} - \mathcal{X}^k \right\|_F}{\left\| \mathcal{X}^k \right\|_F} \leq tol,$$

where k is the number of iterations and tol is the tolerance error. The alternative halting condition is $k \geq K$, where K is the maximum number of iterations. The iteration ends if any of the two conditions is met. We summarize the implementation of the CTNM-QR as Algorithm 1.

```

Input: Missing tensor data  $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , the index set of the observed elements  $\Omega$ , complementary set  $\bar{\Omega}$ , Tucker-rank  $(R_1, R_2, \dots, R_N)$ ,  $\alpha_i, \lambda, \mu_0, \mu_{max}, \rho, K, tol$ ;
Output: Completed tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ 
1 Initialization:  $\mathcal{X}_\Omega = \mathcal{T}_\Omega, \mathcal{X}_{\bar{\Omega}} = 0; \mathcal{Y}_i = 0, \mathcal{C} = rand(R_1, \dots, R_N), \mathcal{V}_i = \mathcal{C}$ , and  $U_i = QR(rand(I_i, R_i))$  for  $i = 1, 2, \dots, N$ , where  $rand(R_1, \dots, R_N)$  and  $rand(I_i, R_i)$  represent the  $R_1 \times \dots \times R_N$  tensor and the  $I_i \times R_i$  matrix randomly from the standard normal distribution;
2 while not converged do
3   for  $i = 1:N$  do
4     Update  $\mathcal{V}_i^{k+1}$  by Equation (11);
5   end
6   Update  $\mathcal{C}^{k+1}$  by solving Equation (12);
7   for  $i = 1:N$  do
8     Update  $U_i^{k+1}$  by Equation (16);
9   end
10  Update  $\mathcal{X}^{k+1}$  by solving Equation (13);
11  for  $i = 1:N$  do
12    Update  $\mathcal{Y}_i^{k+1}$  by  $\mathcal{Y}_i^{k+1} = \mathcal{Y}_i^k + \mu^k (\mathcal{C}^{k+1} - \mathcal{V}_i^{k+1})$ ;
13  end
14  Update  $\mu^{k+1}$  by  $\mu^{k+1} = \min(\rho \mu^k, \mu_{max})$ ;
15  Check the convergence conditions  $\left\| \mathcal{X}^{k+1} - \mathcal{X}^k \right\|_F / \left\| \mathcal{X}^k \right\|_F \leq tol$  or  $k \geq K$ .
16 end

```

Algorithm 1. Core tensor nuclear-norm minimization with QR decomposition (CTNM-QR).

3.3 Convergence analysis

Let $\left\{ \mathcal{C}^k, U_1^k, \dots, U_N^k, \mathcal{V}_1^k, \dots, \mathcal{V}_N^k, \mathcal{X}^k \right\}$ be a sequence generated by the Algorithm 1. Although the CTNM-QR is a nonconvex ADMM algorithm, the subsequence are convergent and converge to the KKT point of the problem (6) proved by Liu et al. (2014b). In the subsection, we only illustrate the convergence of the improved part of the algorithm for the subproblem (9c).

Denote the objective function in Equation (15) as $f(U_n) = \frac{1}{2} \left\| \mathcal{Z}_n \times_n U_n - \mathcal{X} \right\|_F^2$, where $(U_n)^T U_n = I_n (n = 1, 2, \dots, N)$. The function $f(U_n)$ is convex on U_n without considering orthogonality constraints, so minimizing $f(U_n)$ yields $\tilde{U}_n^k = \mathcal{X}_{(n)}^{k-1} \left(\mathcal{Z}_{n(n)}^{k-1} \right)^T \left(\mathcal{Z}_{n(n)}^{k-1} \left(\mathcal{Z}_{n(n)}^{k-1} \right)^T \right)^{-1}$. Then the following equation holds:

$$f(\tilde{U}_n^k) \leq f(U_n^{k-1}).$$

Denote $g(\mathcal{C}^k, U_n) = \frac{1}{2} \left\| \mathcal{C}^k \times_1 U_1^k \cdots \times_{n-1} U_{n-1}^k \times_{n+1} U_{n+1}^{k-1} \cdots \times_N U_N^{k-1} \times_n \tilde{U}_n - \mathcal{X} \right\|_F^2$. By carrying out a thin QR decomposition of \tilde{U}_n^k , we have $\tilde{U}_n^k = U_n^k Q^k$, where U_n^k is an orthogonal matrix. Then we have

$$f(\tilde{U}_n^k) = g(\mathcal{C}^k, \tilde{U}_n^k) = g(\tilde{\mathcal{C}}^k, U_n^k) = f(U_n^k) \leq f(U_n^{k-1})$$

TABLE 1 Computational complexity of each iteration in tensor completion algorithms.

Algorithms	Complexity
TT-WOPT (Yuan et al., 2017)	$\mathcal{O}\left(N\prod_{i=1}^N I_i + R^2 \sum_{i=1}^N \prod_{j \neq i} I_j\right)$
TT-SGD (Yuan et al., 2019b)	$\mathcal{O}(N^2 R^3)$
CP-WOPT (Acar et al., 2011)	$\mathcal{O}\left(2(N+1)R\prod_{i=1}^N I_i\right)$
SiLRTC, HaLRTC (Liu et al., 2013)	$\mathcal{O}\left(\sum_{i=1}^N I_i \prod_{i=1}^N I_i\right)$
CTNM (Liu, 2013)	$\mathcal{O}\left(2(N+1)R\prod_{i=1}^N I_i\right)$
gHOI (Liu et al., 2014b)	$\mathcal{O}\left((N+1)R\prod_{i=1}^N I_i\right)$
CTNM-QR	$\mathcal{O}\left((N+1)R\prod_{i=1}^N I_i\right)$

where $\tilde{\mathcal{C}}^k = \mathcal{C}^k \times_n Q^k$. Therefore, when $\mathcal{C}, \mathcal{X}, \{U_i\}_{i=1, i \neq n}^N$ is fixed, the objective function $f(U_n)$ tends to be decreasing during the iteration process, and the sequence $\{U_1^k, \dots, U_N^k\}$ is the convergence sequence.

3.4 Complexity analysis

In the subsection, we analyze the complexity of the proposed CTNM-QR algorithm. The running time of the CTNM-QR algorithm is primarily consumed in solving the auxiliary variables with the SVT operator and some multiplicative computations. Suppose the size of the input tensor is $I_1 \times I_2 \times \dots \times I_N$ and Tucker rank is $R_1 = \dots = R_N = R$, the time complexity of solving Equation (9a) is $\mathcal{O}(NR^{N+1})$, the time complexity of solving \mathcal{X} in (9d) is $\mathcal{O}\left(R\prod_{i=1}^N I_i\right)$, and the time complexity of computing U_n in Equation (16) is $\mathcal{O}\left(NR\prod_{i=1}^N I_i + R^2 \sum_i I_i\right)$. Consequently, the total time complexity of the Algorithm 1 is $\mathcal{O}\left(T(N+1)R\prod_{i=1}^N I_i\right)$, where T is the total number of iterations. When the N and $I_i, i = 1, 2, \dots, N$ are fixed, the time complexity of the CTNM-QR algorithm mainly dominates on R . Fortunately, the R generally takes small values due to the low-rank assumption of the tensor.

For comparison, we list the major computational complexity of several classical tensor-completion algorithms in Table 1. Although the per-iteration complexity of TT-SGD (Yuan et al., 2019b) is incredibly low, the total number of iterations required is higher and more time-consuming than the proposed CTNM-QR algorithm in practice. In addition, the Polak-Ribiere nonlinear conjugate gradient algorithm of CP-WOPT (Acar et al., 2011) also suffers from the problem of being time-consuming. Compared with the nuclear norm minimization algorithms such as SiLRTC, HaLRTC, CTNM, etc., the proposed CTNM-QR algorithm saves some computational complexity. The CTNM-QR is consistent with the generalized higher-order orthogonal iteration (gHOI) algorithm (Liu et al., 2014b) in terms of the algorithm's primary computational cost.

4 Experiments

In this section, we conduct experiments to evaluate the performance and efficiency of the CTNM-QR algorithm in the

LRTC problem using synthetic tensor data, color images, and medical MRI scans. The experiment compares the CTNM-QR with five other typical completion algorithms: gHOI (Liu et al., 2014b), CTNM (Liu, 2013), FaLRTC (Liu et al., 2013), CP-WOPT (Acar et al., 2011), and TT-SGD (Yuan et al., 2019b), where the parameters of the comparison algorithms are optimized. In addition, by means of studies, we analyze the convergence of the CTNM-QR algorithm intuitively. All the experiments are implemented on Windows 10 and MATLAB (R2018b) with an Intel(R) Core(TM) i5-7200U CPU at 2.70 GHz and 4 GB RAM.

To construct the incomplete tensor artificially, we utilize random uniform sampling from the complete tensor based on the given missing rate (MR), so the tensor missing form is randomized uniform missing. We set up varying MR for the test tensor, where the MR is defined as:

$$MR = 1 - \frac{M}{\prod_{n=1}^N I_n}$$

where M is the number of entries observed in the N order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$. We adopt relative squared error (RSE) and CPU time as evaluation metrics to measure the performance of the tensor completion. The RSE is calculated as follows:

$$RSE = \frac{\|\mathcal{X} - \mathcal{T}\|_F}{\|\mathcal{T}\|_F}$$

where $\|\cdot\|_F$ indicates the Frobenius norm, the \mathcal{X} and \mathcal{T} denote the tensor recovered by the algorithm and the original true tensor, respectively. The algorithm's accuracy in recovering the tensor increases as the RSE decreases, and computational complexity decreases as the CPU time decreases.

4.1 Synthetic tensor completion

In the simulation studies, we generate the low rank synthetic tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ by the Tucker decomposition model, i.e., $\mathcal{T} = \mathcal{C} \times_1 U_1 \times_2 U_2 \times \dots \times_N U_N$, where $\mathcal{C} \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_N}$, $U_i \in \mathbb{R}^{I_i \times R_i} (i = 1, 2, \dots, N)$ and their entries independently follow the standard normal distribution. This implies that the Tucker rank of the tensor \mathcal{T} is (R_1, R_2, \dots, R_N) .

In the concrete implementation, we set the tolerance to $tol = 10^{-5}$ for all algorithms, the maximum number of iterations to $K = 6 \times 10^5$ for the TT-SGD, and $K = 500$ for the rest of the algorithms. For the FaLRTC, gHOI, CTNM, and CTNM-QR, we set the weights $\alpha_i = 1/N, i = 1, 2, \dots, N$. The smoothing parameter of the FaLRTC is set to $\alpha_i/I_i, i = 1, \dots, N$, and the rest of the parameters are maintained at their default values. For the CP-WOPT, the tensor rank R is set to 15. For the TT-SGD, the TT rank is set to $R_1 = \dots = R_{N-1} = 15$. For the gHOI, CTNM, and CTNM-QR, we set $\mu_0 = 10^{-3}, \mu_{\max} = 10^{10}, \rho = 1.05$.

We consider two cases including the fourth-order tensor with the size $40 \times 40 \times 40 \times 40$ and the fifth-order tensor with the size $20 \times 20 \times 20 \times 20 \times 20$. The Tucker-rank of the tensor is set to $R_1 = \dots = R_N = 5$, where the tensor orders N are 4 and 5, respectively. We consider four distinct MRs 30, 50, 70, 80% for the test tensor and replicate 100 simulations. Table 2 presents the average results (RSE and time cost) for 100

TABLE 2 Average RSE and time cost (seconds) of all approaches on various tensor sizes.

MR	CP-WOPT		FaLRTC		TT-SGD		CTNM		gHOI		CTNM-QR	
	RSE	Time	RSE	Time	RSE	Time	RSE	Time	RSE	Time	RSE	Time
$40 \times 40 \times 40 \times 40$												
30%	4.62e-1	73.96	1.47e-2	44.12	5.99e-2	138.25	2.40e-3	9.75	6.73e-4	2.75	3.71e-4	3.06
50%	5.54e-1	100.21	2.95e-2	38.67	9.77e-2	136.57	8.96e-3	20.15	2.36e-3	4.97	1.91e-3	5.56
70%	5.73e-1	144.19	5.61e-2	63.78	1.29e-1	139.18	5.95e-2	40.92	3.96e-3	12.03	3.26e-3	9.95
80%	5.99e-1	238.06	9.19e-2	80.37	2.90e-1	135.12	1.26e-1	106.91	9.10e-3	17.48	6.27e-3	13.79
$20 \times 20 \times 20 \times 20 \times 20$												
30%	4.63e-1	66.18	1.22e-2	60.79	4.27e-2	161.67	2.89e-3	18.63	7.91e-4	3.70	3.07e-4	4.92
50%	6.00e-1	58.37	5.98e-2	68.96	8.18e-2	160.00	6.58e-3	29.25	7.15e-4	6.96	6.00e-4	8.92
70%	7.06e-1	50.39	8.59e-2	80.89	1.68e-1	174.45	9.58e-2	65.25	1.40e-3	18.37	1.20e-3	13.28
80%	7.40e-1	96.97	1.05e-1	94.42	1.78e-1	175.31	8.18e-2	101.26	5.91e-2	23.08	1.40e-2	20.32

Bold values indicate the best results obtained from the experiments.

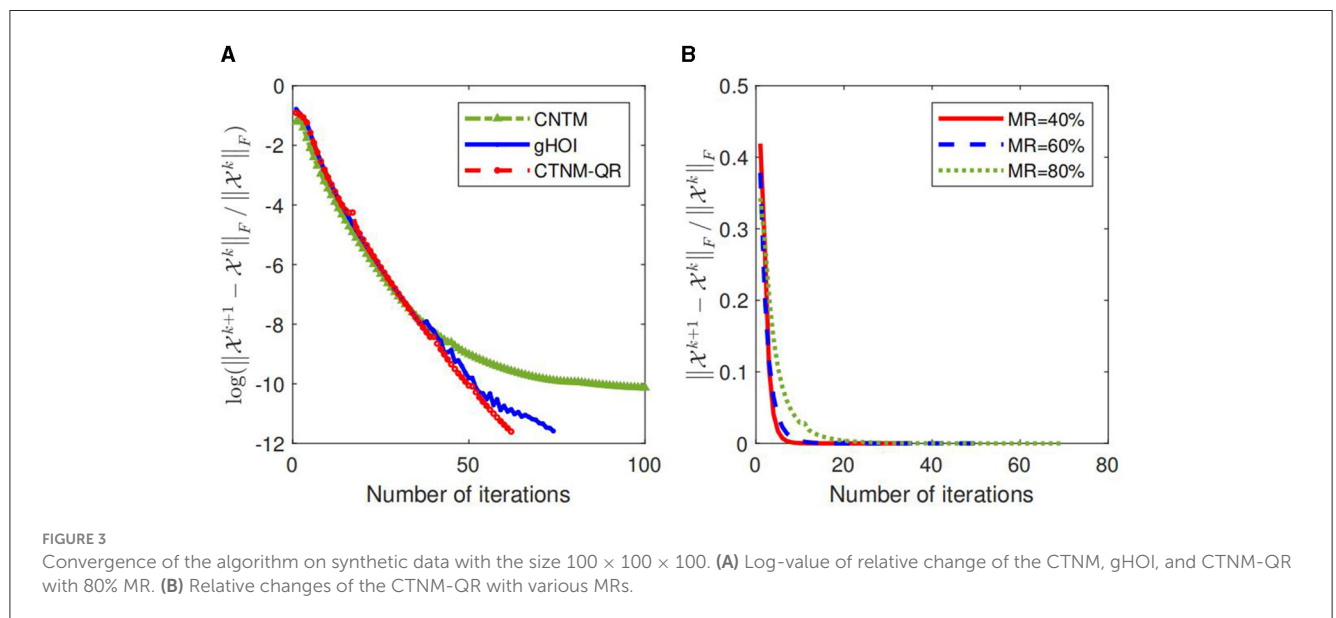


FIGURE 3 Convergence of the algorithm on synthetic data with the size $100 \times 100 \times 100$. (A) Log-value of relative change of the CTNM, gHOI, and CTNM-QR with 80% MR. (B) Relative changes of the CTNM-QR with various MRs.

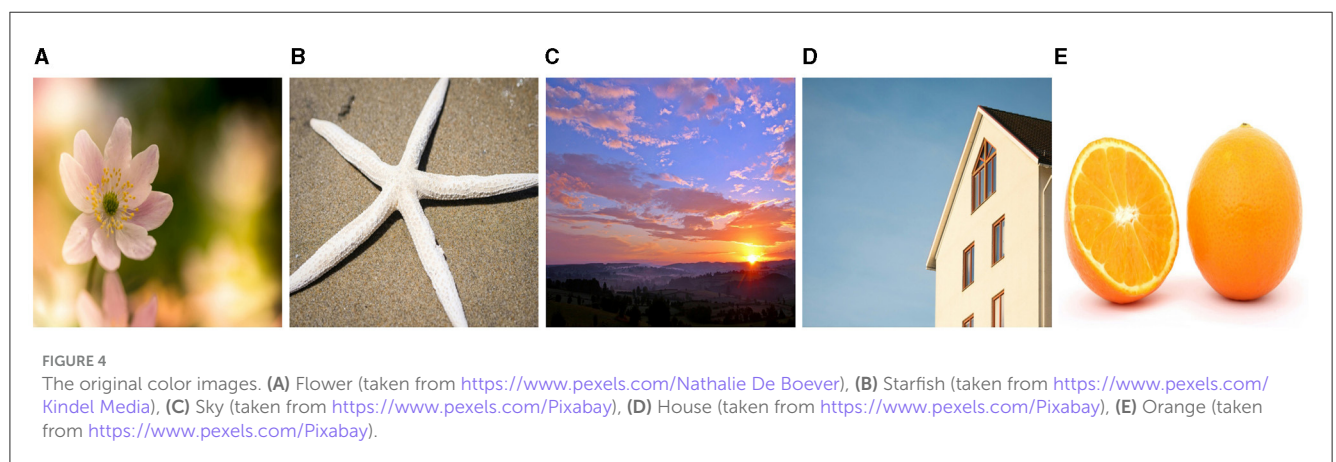


FIGURE 4 The original color images. (A) Flower (taken from <https://www.pexels.com/Nathalie-De-Boever/>), (B) Starfish (taken from <https://www.pexels.com/Kindel-Media/>), (C) Sky (taken from <https://www.pexels.com/Pixabay/>), (D) House (taken from <https://www.pexels.com/Pixabay/>), (E) Orange (taken from <https://www.pexels.com/Pixabay/>).

independent experiments. The outcomes reveal that the CTNM-QR method yields more accurate solutions with the less time costs, and outperforms other algorithms in the term of the RSE regardless

of whether it's a higher order tensor or a lower order tensor. Although the gHOT method has a somewhat shorter running time than the CTNM-QR at the low MRs, the time costs of both methods

TABLE 3 RSE and time cost (seconds) comparison on true color images at different MRs.

MR	Method	Flower		Starfish		Sky		Lena		Apple	
		RSE	Time	RSE	Time	RSE	Time	RSE	Time	RSE	Time
40%	CP-WOPT	7.46e-2	210.25	1.88e-1	186.98	6.58e-2	40.26	1.79e-1	68.70	6.93e-2	100.02
	FaLRTC	1.26e-1	38.50	1.74e-1	44.21	4.47e-2	34.69	1.19e-1	38.79	6.69e-2	48.69
	TT-SGD	1.12e-1	75.05	1.76e-1	77.73	5.56e-2	61.59	1.11e-1	61.10	6.90e-2	69.15
	gHOI	5.32e-2	17.91	1.20e-1	12.83	3.24e-2	9.78	9.59e-2	13.48	3.18e-2	9.43
	CTNM	8.24e-2	19.35	1.69e-1	34.58	2.95e-2	8.60	1.13e-1	26.39	3.66e-2	11.27
	CTNM-QR	4.16e-2	11.31	1.15e-1	14.54	2.56e-2	4.19	8.88e-2	7.84	3.02e-2	3.29
60%	CP-WOPT	1.22e-1	204.18	1.97e-1	350.43	9.03e-2	52.39	1.96e-1	159.59	1.16e-1	115.27
	FaLRTC	1.93e-1	37.74	2.51e-1	53.79	6.50e-2	43.14	1.76e-1	45.32	9.14e-2	66.90
	TT-SGD	1.35e-1	69.01	2.19e-1	79.79	6.55e-2	64.15	1.37e-1	74.22	8.54e-2	79.98
	gHOI	7.97e-2	20.46	1.64e-1	15.34	3.68e-2	10.52	1.11e-1	16.48	4.24e-2	12.37
	CTNM	1.45e-1	35.46	2.79e-1	45.18	3.51e-2	14.60	1.54e-1	37.47	4.05e-2	14.02
	CTNM-QR	5.67e-2	18.26	1.43e-1	17.79	3.08e-2	8.02	1.02e-1	14.39	3.78e-2	7.30
80%	CP-WOPT	1.91e-1	185.96	2.83e-1	181.91	1.37e-1	61.59	2.41e-1	80.00	1.86e-1	130.56
	FaLRTC	3.06e-1	48.92	3.71e-1	59.68	1.58e-1	48.67	2.70e-1	57.96	1.06e-1	57.57
	TT-SGD	1.57e-1	77.56	2.66e-1	69.92	9.20e-2	68.32	1.67e-1	71.20	9.41e-2	73.87
	gHOI	1.41e-1	27.82	2.12e-1	22.37	7.53e-2	14.53	1.34e-1	18.25	7.45e-2	21.52
	CTNM	1.99e-1	50.89	3.36e-1	50.87	1.10e-1	47.72	1.88e-1	49.71	7.78e-2	28.82
	CTNM-QR	9.13e-2	30.45	1.79e-1	19.91	3.90e-2	12.64	1.10e-1	20.58	4.59e-2	15.12

Bold values indicate the best results obtained from the experiments.

are quite close and the RSE of the CTNM-QR is less than one of the gHOI.

We also investigate the convergence of the CTNM, gHOI, and CTNM-QR methods on synthetic tensor data with the Tucker-rank $(R_1, R_2, R_3) = (10, 10, 10)$ and the size $100 \times 100 \times 100$, as shown in Figure 3A, where the abscissa indicates the number of iterations and the vertical indicates the log-value of the relative change of the \mathcal{A}^k . The results display that the relative change of the convergence of the CTNM-QR drops more quickly and smoothly than that of CTNM and gHOI. Figure 3B illustrates the relative changes of the convergence of the CTNM-QR approach at 40, 60, and 80% MRs. From the Figure 3B, we find that even at the high MR, the CTNM-QR converges fast in fewer than 100 iterations.

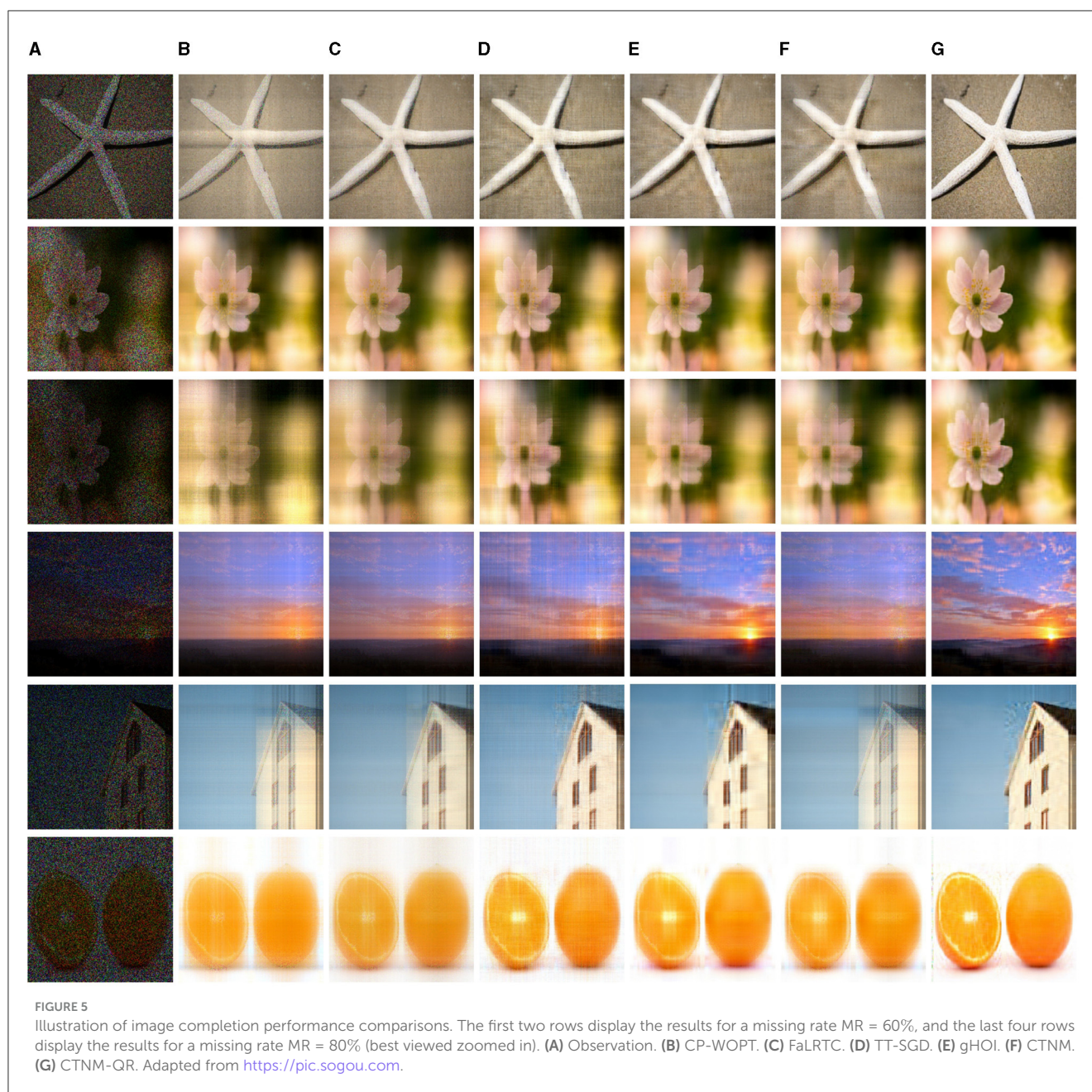
4.2 Color image completion

In this subsection, we confirm the efficacy of the CTNM-QR by performing recovery on RGB real-color images, where each image is represented as a third-order tensor. The low-rank nature of color images has been demonstrated in Figure 1, which allows excellent recovery of incomplete images utilizing the low-rank structure of the tensor. The low-rank color pictures chosen for this experiment include Flower, Starfish, Sky, Lena, and Apple. The original images are portrayed in Figure 4. The images are $500 \times 500 \times 3$ tensors and encompass portraits and various natural landscapes. The image pixels are normalized to $[0, 1]$. We carry out multiple experimental comparisons for every image in order to determine the optimal rank value.

In the concrete implementation, the tolerance error of all algorithms is set to $tol = 10^{-4}$, the maximum number of iterations $K = 5 \times 10^5$ for TT-SGD and $K = 500$ for the rest of the algorithms. For the FaLRTC, gHOI, CTNM and CTNM-QR, we set the weights $\alpha_i = [1, 1, 10^{-3}]$, and the smoothing parameter for the former to $\alpha_i/\sqrt{I_i}$ for $i = 1, 2, 3$. For the CTNM-QR, we set $\mu_0 = 10^{-3}$, $\mu_{\max} = 10^{10}$, $\rho = 1.05$. The rest of the algorithm parameters are preserved at the default settings.

For the color images, we utilize random uniform sampling to make them missing, and the average reconstruction results from 100 independent experiments, including RSE and time cost (seconds), are reported in Table 3, where MR is set to 40, 60, and 80%. The best results are highlighted in bold in the table, and it is clear that the proposed CTNM-QR method achieves the best RSE values in all images and outperforms the other competing methods in terms of CPU time in a substantial number of cases. For example, when recovering Sky with an MR of 80%, the RSE of the CTNM-QR is improved by 0.036 and 0.071 over gHOI and the original algorithm CTNM, respectively, and by 0.119 over the FaLRTC. In terms of running speed, it outperforms the CTNM and the TT-SGD by around 4 and 7 times, respectively. Furthermore, in the majority of image completions, the CP-WOPT method takes the longest. It is obvious that the CTNM-QR can efficiently mine the low-rank image spatial correlation to precisely recover the missing tensor in a reduced iteration time.

Figure 5 depicts visual comparisons of the observed images with the images following the various completion algorithms, where the missing rate of incomplete images is 60 and 80%, and the missing entries are displayed in black. From Figure 5,



it can be seen that the best visibility outcomes are obtained by the CTNM-QR method (Figure 5G), and the recovered image is very clear and close to the original image. In most instances, the results of the CP-WOPT contain some fairly ambiguous parts, and this is the most acute, especially for the Lena and Apple. In the completion of missing data for images with an 80% MR, the FaLRTC and CTNM exhibit suboptimal performance, particularly on the Flower and Lena datasets. Although the results of the gHOI and TT-SGD appear to be more discernible in comparison, the finer details of the images have not been fully recovered, indicating a certain degree of information loss. It implies that the proposed CTNM-QR method can recover the images accurately and efficiently.

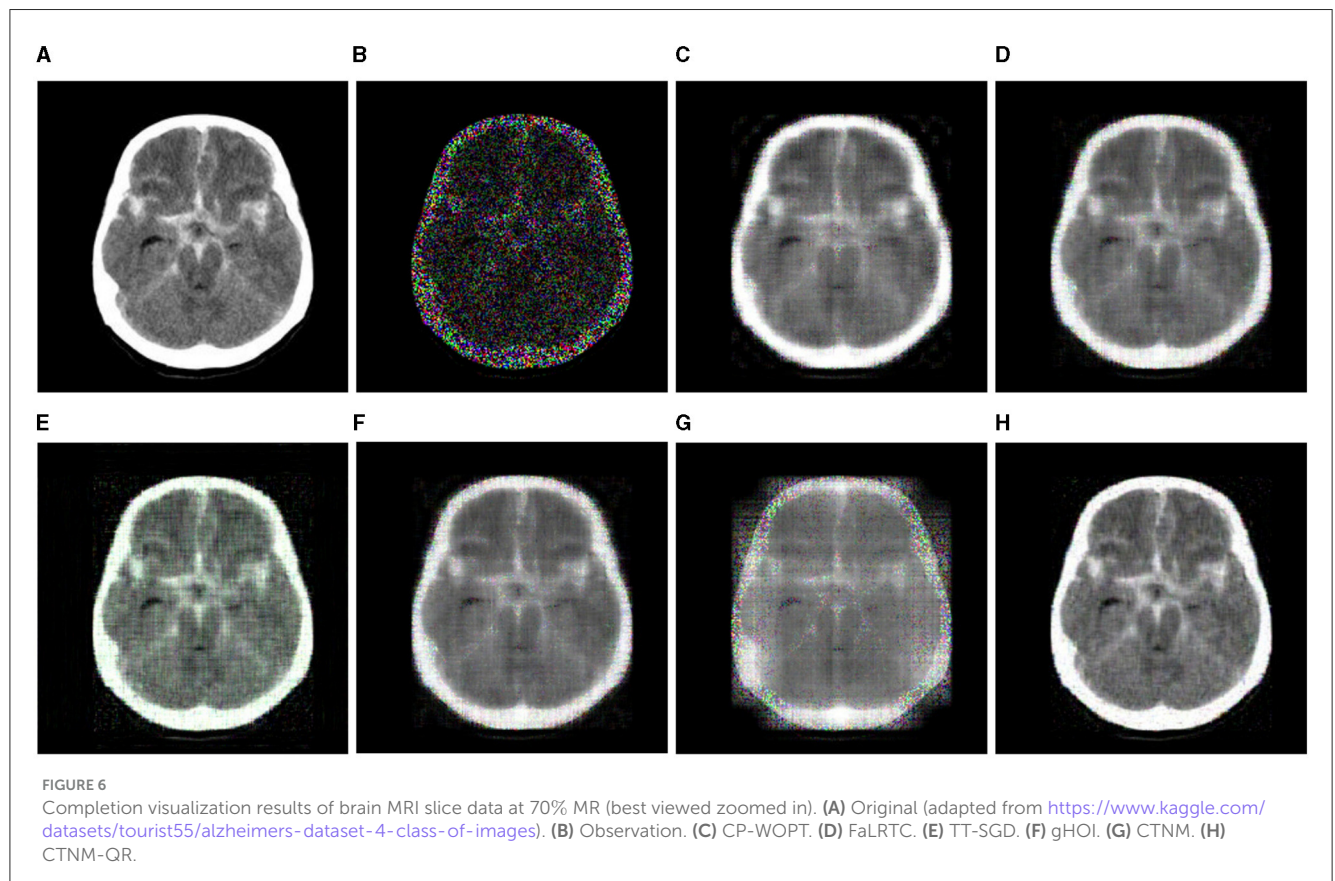
4.3 Brain MRI completion

In the subsection, we evaluate the performance of the CTNM-QR algorithm via the MRI images completion of the brain. MRI images of the brain are acquired through the use of magnetic resonance imaging, which utilizes a strong magnetic field and harmless radio waves. It has an influential role in the fields of medicine and neuroscience, being a key tool used by physicians to diagnose diseases, study brain activity, and guide treatment programs. Here, we analyze brain MRI data with the size $236 \times 236 \times 180$. Similarly, medical MRI data is approximated as a low-rank tensor and thus can be recovered with low-rank completion algorithms. For comparison, the tolerance value for all algorithms

TABLE 4 RSE and time cost (seconds) comparison on brain MRI scanning slice.

MR	CP-WOPT		FaLRTC		TT-SGD		CTNM		gHOI		CTNM-QR	
	RSE	Time	RSE	Time	RSE	Time	RSE	Time	RSE	Time	RSE	Time
30%	9.63e-2	13.69	7.25e-2	8.29	9.75e-2	74.36	4.90e-2	4.59	4.90e-2	4.10	4.89e-2	3.34
50%	1.19e-1	16.56	1.11e-1	9.98	1.29e-1	73.36	6.83e-2	7.07	6.84e-2	8.86	6.80e-2	5.60
70%	1.89e-1	25.27	1.70e-1	13.30	1.57e-1	69.38	3.79e-1	22.92	1.97e-1	11.59	1.21e-1	7.16

Bold values indicate the best results obtained from the experiments.



is set to $tol = 10^{-3}$, the tensor Tucker-rank is set to $R_1 = R_2 = R_3 = 30$ for the gHOI, CTNM, and CTNM-QR, the tensor CP rank is set to 40 for the CP-WOPT, and the TT rank is set to $R_1 = R_2 = 20$ for the TT-SGD. We choose a slice of the MRI images for completion due to their enormous computation cost.

Table 4 presents the recovery accuracy (RSE) and time cost (seconds) of every algorithm at varying MRs, where the MRs are set to 30, 50, and 70%, respectively. The findings show that the RSE of the CTNM-QR, gHOI, and CTNM are nearly equal at MR values of 30 and 50%. However, when the MR increases to 70%, the RSE of the CTNM and gHOI dramatically rose. In terms of the runtime, the proposed CTNM-QR method typically runs faster than others, especially more noticeable at the higher MR. In addition, Figure 6 displays the recovery results of the brain MRI slice data at the 70% MR, where the CTNM-QR outperforms the other algorithms in terms of both recovered vision and efficiency.

5 Conclusion

In this paper, we propose a CTNM-QR method, which is based on an enhancement of CTNM and aims at addressing the problem of completing the low-rank missing tensor. Firstly, we review the CTNM algorithm briefly and discuss our changes, which can be categorized into two major points. One difference is that the model's auxiliary variables are introduced as tensors rather than matrix versions, and the tensor form is able to utilize multidimensional information to better retain the spatial structure of high-dimensional data. The other is that the solution of the factor matrix in the model flexibly adopts the thin QR decomposition instead of HOOI, which avoids the computation of multiple SVDs in each loop and minimizes the computational cost to a certain extent, thus speeding up the process. Secondly, we further investigate the convergence and complexity of the CTNM-QR method. Ultimately, extensive experiments on synthetic data of various orders, real color images, and brain MRI data indicate that

our method not only beats most state-of-the-art LRTC algorithms in terms of completion accuracy and visualization, but also runs faster.

In the future, we will continue to explore how to automatically determine the optimal rank of a tensor in CTNM-QR to settle the rank selection challenge. In addition, combining sophisticated techniques such as deep learning will boost the algorithm's ability to adapt to more complicated data. To widen the method's application scenarios, we are interested in adapting CTNM-QR to additional practical applications such as recommender systems, video denoising, and so on.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

YW: Writing – original draft. YJ: Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This project

was supported from NSF grants of China (12231017, 12001479, 202101AU070073, and 202201AT070101).

Acknowledgments

We would like to thank Kaggle Data for making the data available.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Acar, E., Dunlavy, D. M., Kolda, T. G., and Mørup, M. (2011). Scalable tensor factorizations for incomplete data. *Chemomet. Intell. Lab. Syst.* 106, 41–56. doi: 10.1016/j.chemolab.2010.08.004
- Bai, Z., Li, Y., Woźniak, M., Zhou, M., and Li, D. (2021). Decomvqnet: decomposing visual question answering deep network via tensor decomposition and regression. *Pattern Recognit.* 110:107538. doi: 10.1016/j.patcog.2020.107538
- Bengua, J. A., Phien, H. N., Tuan, H. D., and Do, M. N. (2017). Efficient tensor completion for color image and video recovery: low-rank tensor train. *IEEE Transact. Image Process.* 26, 2466–2479. doi: 10.1109/TIP.2017.2672439
- Boţ, R. I., and Nguyen, D.-K. (2020). The proximal alternating direction method of multipliers in the nonconvex setting: convergence analysis and rates. *Math. Operat. Res.* 45, 682–712. doi: 10.1287/moor.2019.1008
- Boumal, N. (2023). *An Introduction to Optimization on Smooth Manifolds*. Cambridge: Cambridge University Press.
- Cai, J.-F., Candès, E. J., and Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM J. Optimiz.* 20, 1956–1982. doi: 10.1137/080738970
- Candès, E., and Recht, B. (2012). Exact matrix completion via convex optimization. *Commun. ACM* 55, 111–119. doi: 10.1145/2184319.2184343
- Cattell, R. B. (1944). Parallel proportional profiles and other principles for determining the choice of factors by rotation. *Psychometrika* 9, 267–283. doi: 10.1007/BF02288739
- Chao, Z., Huang, L., and Needell, D. (2021). Hosvd-based algorithm for weighted tensor completion. *J. Imaging* 7:110. doi: 10.3390/jimaging7070110
- De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000a). A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.* 21, 1253–1278. doi: 10.1137/S0895479896305696
- De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000b). On the best rank-1 and rank-(r_1, r_2, \dots, r_m) approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.* 21, 1324–1342. doi: 10.1137/S0895479896346995
- Filipović, M., and Jukić, A. (2015). Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimens. Syst. Signal Process.* 26, 677–692. doi: 10.1007/s11045-013-0269-9
- Gandy, S., Recht, B., and Yamada, I. (2011). Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Probl.* 27:025010. doi: 10.1088/0266-5611/27/2/025010
- Glowinski, R. (2014). *On Alternating Direction Methods of Multipliers: A Historical Perspective. Modeling, Simulation and Optimization for Science and Technology*. Berlin: Springer, 59–82.
- Han, D.-R. (2022). A survey on some recent developments of alternating direction method of multipliers. *J. Operat. Res. Soc. China* 10, 1–52. doi: 10.1007/s40305-021-00368-3
- Hillar, C. J., and Lim, L.-H. (2013). Most tensor problems are np-hard. *J. ACM* 60, 1–39. doi: 10.1145/2512329
- Ji, L., Musialski, P., Wonka, P., and Jieping, Y. (2009). “Tensor completion for estimating missing values in visual data,” in *International Conference on Computer Vision* (Kyoto: IEEE), 2114–2121.
- Kajo, I., Kamel, N., and Ruichek, Y. (2019). Incremental tensor-based completion method for detection of stationary foreground objects. *IEEE Transact. Circ. Syst. Video Technol.* 29, 1325–1338. doi: 10.1109/TCSVT.2018.2841825
- Kasai, H., and Mishra, B. (2016). “Low-rank tensor completion: a riemannian manifold preconditioning approach,” in *Proceedings of the 33rd International Conference on Machine Learning, Vol. 48* (New York, NY: PMLR), 1012–1021.
- Kolda, T. G., and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Rev.* 51, 455–500. doi: 10.1137/07070111X
- Kressner, D., Steinlechner, M., and Vandereycken, B. (2014). Low-rank tensor completion by riemannian optimization. *BIT Numer. Math.* 54, 447–468. doi: 10.1007/s10543-013-0455-z

- Liu, J., Musialski, P., Wonka, P., and Ye, J. (2013). Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 208–220. doi: 10.1109/TPAMI.2012.39
- Liu, Y. (2013). *Algorithm Research of Fast Low-Rank Matrix and Tensor Recovery* (PhD thesis). Wanfang Data Resource System.
- Liu, Y., Long, Z., Huang, H., and Zhu, C. (2019a). Low cp rank and Tucker rank tensor completion for estimating missing components in image data. *IEEE Transact. Circ. Syst. Video Technol.* 30, 944–954. doi: 10.1109/TCSVT.2019.2901311
- Liu, Y., Long, Z., and Zhu, C. (2019b). Image completion using low tensor tree rank and total variation minimization. *IEEE Transact. Multimed.* 21, 338–350. doi: 10.1109/TMM.2018.2859026
- Liu, Y., Shang, F., Cheng, H., Cheng, J., and Tong, H. (2014a). “Factor matrix trace norm minimization for low-rank tensor completion,” in *Proceedings of the 2014 SIAM International Conference on Data Mining (SDM)* (Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM)), 866–874.
- Liu, Y., Shang, F., Fan, W., Cheng, J., and Cheng, H. (2014b). Generalized higher-order orthogonal iteration for tensor decomposition and completion. *Adv. Neural Inf. Process. Syst.* 27, 1–9. doi: 10.5555/2968826.2969023
- Miao, J., Kou, K. I., and Liu, W. (2020). Low-rank quaternion tensor completion for recovering color videos and images. *Pattern Recognit.* 107:107505. doi: 10.1016/j.patcog.2020.107505
- Mu, C., Huang, B., Wright, J., and Goldfarb, D. (2014). “Square deal: lower bounds and improved relaxations for tensor recovery,” in *Proceedings of the 31st International Conference on Machine Learning, Vol. 32* (Beijing: PMLR), 73–81.
- Oseledets, I. V. (2011). Tensor-train decomposition. *SIAM J. Sci. Comp.* 33, 2295–2317. doi: 10.1137/090752286
- Panagakis, Y., Kossaifi, J., Chrysos, G. G., Oldfield, J., Nicolaou, M. A., Anandkumar, A., et al. (2021). Tensor methods in computer vision and deep learning. *Proc. IEEE* 109, 863–890. doi: 10.1109/JPROC.2021.3074329
- Qiu, Y., Zhou, G., Zhao, Q., and Xie, S. (2022). Noisy tensor completion via low-rank tensor ring. *IEEE Transact. Neural Netw. Learn. Syst.* 35, 1127–1141. doi: 10.1109/TNNLS.2022.3181378
- Romera-Paredes, B., and Pontil, M. (2013). “A new convex relaxation for tensor completion,” in *Proceedings of the 26th International Conference on Neural Information Processing Systems, Vol. 2* (New York, NY: Curran Associates Inc.), 2967–2975.
- Shang, F., Liu, Y., Cheng, J., and Yan, D. (2017). Fuzzy double trace norm minimization for recommendation systems. *IEEE Transact. Fuzzy Syst.* 26, 2039–2049. doi: 10.1109/TFUZZ.2017.2760287
- Shi, J., Yang, W., Yong, L., and Zheng, X. (2015). Low-rank tensor completion via Tucker decompositions. *J. Comp. Inf. Syst.* 11, 3759–3768. doi: 10.12733/jcis14329
- Shi, Q., Lu, H., and Cheung, Y.-m. (2017). “Tensor rank estimation and completion via cp-based nuclear norm,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM)* (New York, NY: Association for Computing Machinery), 949–958.
- Signoretto, M., De Lathauwer, L., and Suykens, J. A. (2010). *Nuclear Norms for Tensors and Their Use for Convex Multilinear Estimation. Submitted to Linear Algebra and Its Applications*. Amsterdam: Elsevier, 43.
- Su, L., Liu, J., Tian, X., Huang, K., and Tan, S. (2022). Iterative tensor eigen rank minimization for low-rank tensor completion. *Inf. Sci.* 616, 303–329. doi: 10.1016/j.ins.2022.10.061
- Tucker, L. R. (1963). Implications of factor analysis of three-way matrices for measurement of change. *Probl. Meas. Change* 15, 122–137. doi: 10.1108/09534810210423008
- Xu, K., Zhang, Y., and Xiong, Z. (2021). Iterative rank-one matrix completion via singular value decomposition and nuclear norm regularization. *Inf. Sci.* 578, 574–591. doi: 10.1016/j.ins.2021.07.035
- Yoshii, K., Kitamura, K., Bando, Y., Nakamura, E., and Kawahara, T. (2018). “Independent low-rank tensor analysis for audio source separation,” in *2018 26th European Signal Processing Conference (EUSIPCO)* (IEEE), 1657–1661.
- Yu, Q., Zhang, X., Chen, Y., and Qi, L. (2023). Low Tucker rank tensor completion using a symmetric block coordinate descent method. *Numer. Linear Algebra Appl.* 30:e2464. doi: 10.1002/nla.2464
- Yuan, L., Li, C., Mandic, D., Cao, J., and Zhao, Q. (2019a). “Tensor ring decomposition with rank minimization on latent space: an efficient approach for tensor completion,” in *Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33* (New York, NY: AAAI Press), 9151–9158.
- Yuan, L., Zhao, Q., and Cao, J. (2017). “Completion of high order tensor data with missing entries via tensor-train decomposition,” in *International Conference on Neural Information Processing* (Berlin: Springer), 222–229.
- Yuan, L., Zhao, Q., Gui, L., and Cao, J. (2019b). High-order tensor completion via gradient-based optimization under tensor train format. *Signal Process. Image Commun.* 73, 53–61. doi: 10.1016/j.image.2018.11.012
- Zhang, Z., and Aeron, S. (2016). Exact tensor completion using t-svd. *IEEE Transact. Signal Process.* 65, 1511–1526. doi: 10.1109/TSP.2016.2639466
- Zhao, X.-L., Xu, W.-H., Jiang, T.-X., Wang, Y., and Ng, M. K. (2020). Deep plug-and-play prior for low-rank tensor completion. *Neurocomputing* 400, 137–149. doi: 10.1016/j.neucom.2020.03.018
- Zhou, P., Lu, C., Lin, Z., and Zhang, C. (2018). Tensor factorization for low-rank tensor completion. *IEEE Transact. Image Process.* 27, 1152–1163. doi: 10.1109/TIP.2017.2762595