Journal of Scientific Computing (2024) 101:1
https://doi.org/10.1007/s10915-024-02624-z

Quaternion Tensor Left Ring Decomposition and Application for Color Image Inpainting

Jifei Miao[1] · Kit Ian Kou[2] · Hongmin Cai[3] · Lizhi Liu[4]

Received: 26 September 2023 / Revised: 13 April 2024 / Accepted: 7 July 2024

Abstract

In recent years, tensor networks have emerged as powerful tools for solving large-scale optimization problems. One of the most promising tensor networks is the tensor ring (TR) decomposition, which achieves circular dimensional permutation invariance in the model through the utilization of the trace operation and equitable treatment of the latent cores. On the other hand, more recently, quaternions have gained significant attention and have been widely utilized in color image processing tasks due to their effectiveness in encoding color pixels by considering the three color channels as a unified entity. Therefore, in this paper, based on the left quaternion matrix multiplication, we propose the quaternion tensor left ring (QTLR) decomposition, which inherits the powerful and generalized representation abilities of the TR decomposition while leveraging the advantages of quaternions for color pixel representation. In addition to providing the definition of QTLR decomposition and an algorithm for learning the QTLR format, the paper further proposes a low-rank quaternion tensor completion (LRQTC) model and its algorithm for color image inpainting based on the defined QTLR decomposition. Finally, extensive experiments on color image inpainting demonstrate that the proposed LRQTC method is highly competitive.

✉ Kit Ian Kou
kikou@umac.mo

Jifei Miao
miaojifei@ynu.edu.cn

Hongmin Cai
hmcai@scut.edu.cn

Lizhi Liu
liulizh@sysucc.org.cn

[1] The School of Mathematics and Statistics, Yunnan University, Kunming 650091, Yunnan, China

[2] The Department of Mathematics, Faculty of Science and Technology, University of Macau, Macau 999078, China

[3] The School of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, Guangdong, China

[4] The Department of Radiology, State Key Laboratory of Oncology in South China, Collaborative Innovation Center for Cancer Medicine, Guangdong Key Laboratory of Nasopharyngeal Carcinoma Diagnosis and Therapy, Sun Yat-sen University Cancer Center, 651 Dongfeng Road East, Guangzhou 510060, Guangdong, China

Published online: 13 August 2024

Keywords Quaternion tensor left ring decomposition · Quaternion tensor low-rank completion · Color image inpainting

Mathematics Subject Classification 11R52 · 15A69 · 15A83 · 94A08

1 Introduction

Tensor networks have gained prominence in recent years as powerful tools for tackling large-scale optimization problems [6, 17, 27, 42, 44]. Among them, the tensor ring (TR) decomposition [42] is one of the most advanced tensor networks. The TR decomposition represents an N-th order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ by multiplying a sequence of third-order tensors $\mathcal{Z}_n$, $n = 1, 2, \ldots, N$ in a circular manner. Specifically, it can be expressed in an element-wise form given by

$$\mathcal{T}(i_1, i_2, \ldots, i_N) = \mathrm{Tr}\{\mathcal{Z}_1(i_1)\mathcal{Z}_2(i_2)\ldots\mathcal{Z}_N(i_N)\}, \tag{1}$$

where $\mathcal{T}(i_1, i_2, \ldots, i_N)$ denotes the $(i_1, i_2, \ldots, i_N)$-th element of $\mathcal{T}$, $\mathrm{Tr}\{\cdot\}$ denotes the trace operator, $\mathcal{Z}_n(i_n) \in \mathbb{R}^{r_n \times r_{n+1}}$ denotes the i_n-th lateral slice of the TR factor $\mathcal{Z}_n$, the last TR factor $\dot{\mathcal{Z}}_N$ is of size $r_N \times I_N \times r_1$, *i.e.*, $r_{N+1} = r_1$. The TR decomposition has been widely utilized in various image processing tasks due to its powerful and generalized representation ability. In particular, the TR-based low-rank tensor completion (LRTC) methods for image inpainting have been extensively studied recently [12, 29, 34, 38]. For example, Wang et al. presented a TR-based completion algorithm in [32], which involves alternately updating each TR factor. Nevertheless, the performance of the algorithm is influenced by the pre-defined TR-rank, leading to a significant increase in computational cost. In [38], Yuan et al. addressed these challenges by applying matrix nuclear norm regularization to the mode-2 unfolding of each TR factor, thereby improving the stability of the performance. In [11, 36], the authors proposed a TR nuclear norm minimization model using a tensor circular unfolding scheme for tensor completion. Notably, this approach does not rely on a pre-defined TR-rank and demonstrates superior performance compared to previous TR decomposition-based methods. However, when dealing with color pixels comprising RGB channels, real-valued third-order tensors may not fully exploit the strong correlation among the three channels. This limitation arises from the fact that real-valued third-order tensors represent color images by simply concatenating the RGB channels together, treating both the 'intra-channel relationship' (the relationship within each channel) and the 'spatial relationship' (the relationship between pixels) equally [26].

On the other hand, quaternions have gained considerable attention in the field of color image processing as a more suitable tool for representing color pixels. Concretely, the quaternion-based method encodes the RGB three-channel pixel values on the three imaginary parts of a quaternion [18]. That is,

$$\dot{i} = 0 + t_r i + t_g j + t_b k, \tag{2}$$

where $\dot{i}$ denotes a color pixel, t_r, t_g, and t_b are RGB three-channel pixel values, i, j, and k are the three imaginary units. While both real-valued third-order tensors and quaternion matrices can be utilized for representing color images, quaternion matrices, being a novel representation, possess more favorable characteristics and advantages in this context. Quaternions treat the three channels of color pixels as a cohesive entity [2, 5, 14, 22], thereby effectively preserving the intra-channel relationship. Hence, quaternion matrices, particularly their low-rank approximation models, have been extensively utilized for color image processing tasks

recently. For instance, quaternion matrix rank minimization methods [13, 15, 20, 35] and quaternion matrix factorization methods [3, 22]. These methods have achieved remarkable results in tasks such as color image inpainting and color image denoising. While there has been a significant amount of research progress on quaternion matrices recently, the study on quaternion tensors has just begun, particularly in the context of quaternion tensor networks,[1] which remains largely unexplored. The research on quaternion tensors goes beyond a mere expansion of quaternion theory; it is primarily driven by applications. Most notably, for color videos, a third-order quaternion tensor is required for representation in the most intuitive manner [10, 24]. Additionally, there has been a proliferation of techniques for data reshuffling or dimensionality enhancement, such as KA [1], OKA [40], Hankelization [43], and methods that leverage non-local data similarity to rearrange image or video data [19]. These methods typically bring certain prior information within the original data into clearer focus, leading to substantial improvements in the final processing outcomes for specific tasks. However, these techniques generally result in an increase in data dimensions. Therefore, for color data, the exploration of higher-order quaternion arrays, namely higher-order quaternion tensors, becomes particularly essential. Currently, research on quaternion tensors is primarily focused on their singular value decomposition methods [23, 28]. However, the singular value decomposition of quaternion tensors often involves a high computational workload, making it unsuitable for processing large-scale data in practical applications. Thus, it is necessary to study the theory of quaternion tensor networks, which involves representing large quaternion tensors using relatively smaller quaternion tensors. This approach helps alleviate the challenges of storage and processing of large-scale data.

Consequently, in this paper, we aim to propose the quaternion tensor left ring (QTLR)[2] decomposition, which will inherit the powerful and generalized representation capabilities of the TR decomposition while leveraging the advantages of quaternions for color pixel representation. It is important to note that QTLR diverges from a straightforward extension of TR to quaternions, primarily due to the non-commutativity of quaternion multiplication. This inherent non-commutativity gives rise to disparities in the definitions and associated properties of QTLR in comparison to TR, which is the rationale behind the introduction of QTLR as a distinct concept from TR. Furthermore, as an important application of the defined QTLR, we propose a low-rank quaternion tensor completion (LRQTC) method based on QTLR decomposition to address the inpainting task in color images. This method can mitigate the limitations of quaternion matrix-based approaches, which are not suitable for higher-dimensional quaternion data, and tensor-based methods, which may be unable to distinguish between the intra-channel relationship and spatial relationship of color pixels. Therefore, the proposed QTLR-based LRQTC method is expected to make further advancements in the inpainting of color images compared to existing methods.

We outline the primary contributions of this paper as follows:

– We define the QTLR decomposition for quaternion tensors and prove its cyclic permutation property. It is worth noting that when quaternion tensors degenerate into real tensors, the definition of QTLR decomposition and the cyclic permutation property will degenerate into their corresponding real counterparts as presented in [42]. Furthermore,

[1] The term 'tensor network' refers to the representation of higher-order tensors through interconnected lower-order tensors [7]. Within the tensor networks family, CP, Tucker, tensor train (TT), and TR decompositions are all included.

[2] The term 'left' originates from our use of left quaternion matrix multiplication (*see* Definition 5) to define quaternion tensor ring decomposition.

inspired by the TR-SVD algorithm introduced in [42], we also present the QTLR-QSVD algorithm for learning the QTLR format.
– We generalize the tensor circular unfolding scheme from [11, 36] to quaternion tensors and define the quaternion tensor circular unfolding. For the circular unfolding quaternion matrices that satisfy a certain condition, we prove the relationship between their rank and the defined QTLR-rank. Based on this, we propose a LRQTC model along with its corresponding computational method, which can be considered as an important application of QTLR decomposition.
– We extend a tensor augmentation technique called OKA [40] to quaternion matrices, enabling the transformation of quaternion matrices into higher-order quaternion tensors. Subsequently, we apply the proposed LRQTC method to color image inpainting tasks. Experimental results validate the competitiveness of it.

The reminder of this paper is organized as follows. In Sect. 2, we present certain notations and foundational concepts pertaining to quaternion algebra. This encompasses quaternion matrices and quaternion tensors. In Sect. 3, we provide the definition of QTLR along with its associated properties. Additionally, within this section, we present a learning algorithm for the QTLR format, termed QTLR-QSVD. In Sect. 4, we propose an LRQTC model along with its corresponding algorithm. Section 5 outlines the concrete process of color image inpainting and presents the experimental results. The conclusion is ultimately provided in Sect. 6.

2 Preliminary

Within this section, we introduce specific notations and fundamental principles related to the realm of quaternion algebra, quaternion matrices, and quaternion tensors.

2.1 Notations

In this paper, $\mathbb{R}$, $\mathbb{C}$, and $\mathbb{H}$ respectively denote the real space, complex space, and quaternion space. A scalar, a vector, a matrix, and a tensor are written as a, $\mathbf{a}$, $\mathbf{A}$, and $\mathcal{A}$ respectively. $\dot{a}$, $\dot{\mathbf{a}}$, $\dot{\mathbf{A}}$, and $\dot{\mathcal{A}}$ respectively represent a quaternion scalar, a quaternion vector, a quaternion matrix, and a quaternion tensor. The $(i_1, i_2, \ldots, i_N)$-th element of $\dot{\mathcal{A}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ is denoted as $\dot{\mathcal{A}}(i_1, i_2, \ldots, i_N)$. $(\cdot)^*$, $(\cdot)^T$, and $(\cdot)^H$ denote the conjugate, transpose, and conjugate transpose, respectively. $\text{rank}(\cdot)$ and $\text{Tr}\{\cdot\}$ respectively denote the rank and trace operators. $\mathfrak{R}(\cdot)$ denotes the real part of quaternion (scalar, vector, matrix, and tensor). $\text{diag}(\cdot)$, $\text{reshape}(\cdot)$, and $\text{permute}(\cdot)$ are command operations in *MATLAB* that respectively represent the generation of a diagonal matrix, reshaping of arrays, and rearrangement of array dimensions. In addition, $\|\cdot\|_F$, $\|\cdot\|_{w,*}$, and $\langle\cdot,\cdot\rangle$ are respectively the Frobenius norm, the weighted nuclear norm [37], and the inner product operation.

2.2 Introduction to Quaternions

Quaternion was introduced by Hamilton [9]. A quaternion $\dot{q} \in \mathbb{H}$ has a Cartesian form given by:

$$\dot{q} = q_0 + q_1 i + q_2 j + q_3 k, \tag{3}$$

where $q_l \in \mathbb{R}$ $(l = 0, 1, 2, 3)$ are called its components, i, j, and k are the three imaginary units related through the famous relations:

$$\begin{cases} i^2 = j^2 = k^2 = ijk = -1, \\ ij = -ji = k, jk = -kj = i, ki = -ik = j. \end{cases} \tag{4}$$

Quaternions have similar rules for addition, subtraction, multiplication, and division as complex numbers, as well as similar definitions for conjugation and modulus. However, the difference lies in the non-commutativity property of quaternion multiplication. That is, in general $\dot{p}\dot{q} \neq \dot{q}\dot{p}$.

A multidimensional array or an N-th order tensor is named a quaternion tensor if its elements are quaternions (quaternion matrices can be regarded as second-order quaternion tensors), i.e., $\dot{\mathcal{T}} = (\dot{t}_{i_1 i_2 \ldots i_N}) \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N} = \mathcal{T}_0 + \mathcal{T}_1 i + \mathcal{T}_2 j + \mathcal{T}_3 k$, where $\mathcal{T}_l \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ $(l = 0, 1, 2, 3)$, $\dot{\mathcal{T}}$ is pure if $\mathcal{T}_0$ is a zero tensor [24]. The definition of the inner product between two quaternion tensors, $\dot{\mathcal{X}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ and $\dot{\mathcal{Y}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$, is given by: $\langle \dot{\mathcal{X}}, \dot{\mathcal{Y}} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \dot{x}^*_{i_1 i_2 \ldots i_N} \dot{y}_{i_1 i_2 \ldots i_N}$. The Frobenius norm of quaternion tensor $\dot{\mathcal{X}}$ is $\|\dot{\mathcal{X}}\|_F = \sqrt{\langle \dot{\mathcal{X}}, \dot{\mathcal{X}} \rangle}$.

The most common approach in studying higher-order tensors is to unfold them into matrices. Thus, we extend three unfolding methods for real tensors in [42] to quaternion tensors.

Definition 1 (*Multi-Index Operation* [29]) The multi-index operation is defined as follows:

$$\overline{i_1 i_2 \ldots i_N} = i_1 + (i_2 - 1) I_1 + (i_3 - 1) I_1 I_2 + \ldots + (i_N - 1) \prod_{n=1}^{N-1} I_n,$$

where $i_n \in [I_n]$.

Definition 2 (**k-Unfolding**) Let $\dot{\mathcal{T}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ be an N-th order quaternion tensor, the k-unfolding of $\dot{\mathcal{T}}$ is a quaternion matrix, denoted by $\dot{\mathbf{T}}_{\langle k \rangle}$ of size $\prod_{n=1}^{k} I_n \times \prod_{n=k+1}^{N} I_n$, whose elements are defined by

$$\dot{\mathbf{T}}_{\langle k \rangle}(m, n) = \dot{\mathcal{T}}(i_1, i_2, \ldots, i_N),$$

where $m = \overline{i_1 i_2 \ldots i_k}$, $n = \overline{i_{k+1} i_{k+2} \ldots i_N}$.

Definition 3 (**Mode-k Unfolding**) Let $\dot{\mathcal{T}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ be an N-th order quaternion tensor, the mode-k unfolding of $\dot{\mathcal{T}}$ is a quaternion matrix, denoted by $\dot{\mathbf{T}}_{[k]}$ of size $I_k \times \prod_{n \neq k} I_n$, whose elements are defined by

$$\dot{\mathbf{T}}_{[k]}(i_k, t) = \dot{\mathcal{T}}(i_1, i_2, \ldots, i_N),$$

where $t = \overline{i_{k+1} \ldots i_N i_1 \ldots i_{k-1}}$.

Definition 4 (**Classical Mode-k Unfolding**) Let $\dot{\mathcal{T}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ be an N-th order quaternion tensor, the classical mode-k unfolding of $\dot{\mathcal{T}}$ is a quaternion matrix, denoted by $\dot{\mathbf{T}}_{(k)}$ of size $I_k \times \prod_{n \neq k} I_n$, whose elements are defined by

$$\dot{\mathbf{T}}_{(k)}(i_k, t) = \dot{\mathcal{T}}(i_1, i_2, \ldots, i_N),$$

where $t = \overline{i_1 \ldots i_{k-1} i_{k+1} \ldots i_N}$.

In order to provide the definition of QTLR decomposition, we first introduce the left and right quaternion matrix multiplications.

Definition 5 (*Left and Right Quaternion Matrix Multiplications* [30]) Given two quaternion matrices $\dot{\mathbf{A}} \in \mathbb{H}^{M\times N}$ and $\dot{\mathbf{B}} \in \mathbb{H}^{N\times P}$, the left and right multiplications are respectively defined as

$$(\dot{\mathbf{A}} \cdot_L \dot{\mathbf{B}})_{mp} = \sum_{n=1}^{N} \dot{a}_{mn}\dot{b}_{np} \quad \text{and} \quad (\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}})_{mp} = \sum_{n=1}^{N} \dot{b}_{np}\dot{a}_{mn}. \tag{5}$$

It is easy to verify the following relationship between the left and right quaternion matrix multiplications:

$$\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}} = (\dot{\mathbf{B}}^T \cdot_L \dot{\mathbf{A}}^T)^T, \tag{6}$$

For the sake of brevity in expression, we use $\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}$ to denote the right quaternion matrix multiplication of $\dot{\mathbf{A}}$ and $\dot{\mathbf{B}}$ instead of using $(\dot{\mathbf{B}}^T \cdot_L \dot{\mathbf{A}}^T)^T$. Note that due to the non-commutativity of quaternion multiplication, generally $\dot{\mathbf{A}} \cdot_L \dot{\mathbf{B}} \neq \dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}$.[3] For simplicity, we also define $\dot{a} \cdot_L \dot{b} = \dot{a}\dot{b}$ and $\dot{a} \cdot_R \dot{b} = \dot{b}\dot{a}$ for quaternion scalars $\dot{a}$ and $\dot{b}$. Additionally, if we do not specify whether it is left multiplication or right multiplication, it is assumed to be left multiplication, *i.e.,* $\dot{\mathbf{A}}\dot{\mathbf{B}} = \dot{\mathbf{A}} \cdot_L \dot{\mathbf{B}}$. The defined left and right quaternion matrix multiplications have the following associativity property [31]:

$$(\dot{\mathbf{A}} \cdot_L \dot{\mathbf{B}}) \cdot_L \dot{\mathbf{C}} = \dot{\mathbf{A}} \cdot_L (\dot{\mathbf{B}} \cdot_L \dot{\mathbf{C}}) \quad \text{and} \quad (\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}) \cdot_R \dot{\mathbf{C}} = \dot{\mathbf{A}} \cdot_R (\dot{\mathbf{B}} \cdot_R \dot{\mathbf{C}}). \tag{7}$$

However, in general

$$(\dot{\mathbf{A}} \cdot_L \dot{\mathbf{B}}) \cdot_R \dot{\mathbf{C}} \neq \dot{\mathbf{A}} \cdot_L (\dot{\mathbf{B}} \cdot_R \dot{\mathbf{C}}) \quad \text{and} \quad (\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}) \cdot_L \dot{\mathbf{C}} \neq \dot{\mathbf{A}} \cdot_R (\dot{\mathbf{B}} \cdot_L \dot{\mathbf{C}}). \tag{8}$$

In addition, the property that $\text{rank}(\dot{\mathbf{A}} \cdot_L \dot{\mathbf{B}}) \leq \min(\text{rank}(\dot{\mathbf{A}}), \text{rank}(\dot{\mathbf{B}}))$ can be obtained from [4, 39], where the rank of a quaternion matrix $\dot{\mathbf{A}}$ is defined to be the maximum number of columns of $\dot{\mathbf{A}}$ which are right linearly independent [39]. In the following lemma, we demonstrate the same property for the right multiplication of two quaternion matrices.

Lemma 1 *For any two quaternion matrices $\dot{\mathbf{A}} \in \mathbb{H}^{M\times N}$ and $\dot{\mathbf{B}} \in \mathbb{H}^{N\times P}$, we have*

$$\text{rank}(\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}) \leq \min(\text{rank}(\dot{\mathbf{A}}), \text{rank}(\dot{\mathbf{B}})). \tag{9}$$

Proof Denote the right row null spaces of $\dot{\mathbf{B}}$ and $\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}$ as $\mathcal{RRN}(\dot{\mathbf{B}}) = \{\dot{\mathbf{x}} \in \mathbb{H}^P : \dot{\mathbf{B}} \cdot_R \dot{\mathbf{x}} = \mathbf{0}\}$ and $\mathcal{RRN}(\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}) = \{\dot{\mathbf{x}} \in \mathbb{H}^P : (\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}) \cdot_R \dot{\mathbf{x}} = \mathbf{0}\}$ [31]. One can easily find that $\mathcal{RRN}(\dot{\mathbf{B}}) \subseteq \mathcal{RRN}(\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}})$. Thus, $\dim \mathcal{RRN}(\dot{\mathbf{B}}) \leq \dim \mathcal{RRN}(\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}})$ and $\text{rank}(\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}) \leq \text{rank}(\dot{\mathbf{B}})$. Similarly, Denote the right column null spaces of $\dot{\mathbf{A}}$ and $\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}$ as $\mathcal{RCN}(\dot{\mathbf{A}}) = \{\dot{\mathbf{x}} \in \mathbb{H}^M : \dot{\mathbf{x}}^T \cdot_R \dot{\mathbf{A}} = \mathbf{0}^T\}$ and $\mathcal{RCN}(\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}) = \{\dot{\mathbf{x}} \in \mathbb{H}^M : \dot{\mathbf{x}}^T \cdot_R (\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}) = \mathbf{0}^T\}$ [31]. One can also find that $\mathcal{RCN}(\dot{\mathbf{A}}) \subseteq \mathcal{RCN}(\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}})$. Thus, $\dim \mathcal{RCN}(\dot{\mathbf{A}}) \leq \dim \mathcal{RCN}(\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}})$ and $\text{rank}(\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}) \leq \text{rank}(\dot{\mathbf{A}})$. In all, $\text{rank}(\dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}) \leq \min(\text{rank}(\dot{\mathbf{A}}), \text{rank}(\dot{\mathbf{B}}))$.

□

3 Quaternion Tensor Left Ring Decomposition

In this section, we first define the QTLR decomposition. Following the definition, we present an important property of the QTLR decomposition, and finally propose an algorithm for learning the QTLR format.

[3] From $\dot{\mathbf{A}} \cdot_L \dot{\mathbf{B}} \neq \dot{\mathbf{A}} \cdot_R \dot{\mathbf{B}}$ and (6), we can directly obtain another quaternion matrix property distinct from real matrices, namely, in general $(\dot{\mathbf{B}}^T \cdot_L \dot{\mathbf{A}}^T)^T \neq \dot{\mathbf{A}} \cdot_L \dot{\mathbf{B}}$.

3.1 The Definition of QTLR Decomposition

Definition 6 (*QTLR Decomposition*) Let $\dot{\mathcal{T}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ be an N-th order quaternion tensor with I_n-dimension along the n-th mode, then QTLR representation is to decompose it into a sequence of third-order quaternion tensors $\dot{\mathcal{Z}}_n \in \mathbb{H}^{r_n \times I_n \times r_{n+1}}$ (which can be also called the n-th core of $\dot{\mathcal{T}}$), $n = 1, 2, \ldots, N$, which can be represented using an element-wise formulation as[4]

$$\dot{\mathcal{T}}(i_1, i_2, \ldots, i_N) = \mathrm{Tr}\{\dot{\mathcal{Z}}_1(i_1) \cdot_L \dot{\mathcal{Z}}_2(i_2) \cdot_L \ldots \cdot_L \dot{\mathcal{Z}}_N(i_N)\}, \tag{10}$$

where $\dot{\mathcal{T}}(i_1, i_2, \ldots, i_N)$ denotes the $(i_1, i_2, \ldots, i_N)$-th element of $\dot{\mathcal{T}}$, $\dot{\mathcal{Z}}_n(i_n) = \dot{\mathcal{Z}}_n(:, i_n, :) \in \mathbb{H}^{r_n \times r_{n+1}}$ denotes the i_n-th lateral slice quaternion matrix of the third-order quaternion tensor $\dot{\mathcal{Z}}_n$, the last third-order quaternion tensor $\dot{\mathcal{Z}}_N$ is of size $r_N \times I_N \times r_1$, *i.e.*, $r_{N+1} = r_1$, which ensures the product of these quaternion matrices is a square quaternion matrix. In addition, the vector $\mathbf{r} = [r_1, r_2, \ldots, r_N]$ is defined as the QTLR-rank of the quaternion tensor $\dot{\mathcal{T}}$.

Note that formula (10) can also be expressed in index form as follows:

$$\dot{\mathcal{T}}(i_1, i_2, \ldots, i_N) = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_N=1}^{r_N} \dot{\mathcal{Z}}_1(\alpha_1, i_1, \alpha_2) \cdot_L \mathcal{Z}_2(\alpha_2, i_2, \alpha_3) \cdot_L \ldots \cdot_L \mathcal{Z}_N(\alpha_N, i_N, \alpha_{N+1}), \tag{11}$$

where $\alpha_{N+1} = \alpha_1$. Thus, one can easily find that quaternion tensor train (QTT) decomposition [25] is a special case of the defined QTLR decomposition when $r_1 = 1$.

For an efficient representation of QTLR decomposition, we introduce two quaternion tensor multiplications for third-order quaternion tensors, namely the quaternion tensor left connection multiplication and the quaternion tensor right connection multiplication.

Definition 7 (*Quaternion Tensor Left and Right Connection Multiplications*) Let $\dot{\mathcal{Z}}_n \in \mathbb{H}^{r_n \times I_n \times r_{n+1}}$, $n = 1, 2, \ldots, N$, be N third-order quaternion tensors, the quaternion tensor left and right connection multiplications between $\dot{\mathcal{Z}}_n$ and $\dot{\mathcal{Z}}_{n+1}$ are respectively defined as

$$\dot{\mathcal{Z}}_n \cdot_L \dot{\mathcal{Z}}_{n+1} \in \mathbb{H}^{r_n \times I_n I_{n+1} \times r_{n+2}} = \mathrm{reshape}(\dot{\mathbf{Z}}_n^L \cdot_L \dot{\mathbf{Z}}_{n+1}^R, [r_n, I_n I_{n+1}, r_{n+2}]) \tag{12}$$

and

$$\dot{\mathcal{Z}}_n \cdot_R \dot{\mathcal{Z}}_{n+1} \in \mathbb{H}^{r_n \times I_n I_{n+1} \times r_{n+2}} = \mathrm{reshape}(\dot{\mathbf{Z}}_n^L \cdot_R \dot{\mathbf{Z}}_{n+1}^R, [r_n, I_n I_{n+1}, r_{n+2}]), \tag{13}$$

where $\dot{\mathbf{Z}}_n^L \in \mathbb{H}^{r_n I_n \times r_{n+1}} = (\dot{\mathbf{Z}}_n)_{\langle 2 \rangle}$ and $\dot{\mathbf{Z}}_{n+1}^R \in \mathbb{H}^{r_{n+1} \times I_{n+1} r_{n+2}} = (\dot{\mathbf{Z}}_{n+1})_{\langle 1 \rangle}$.

Then, following the definition (12), the QTLR decomposition (10) can be represented as

$$\dot{\mathcal{T}} = f(\dot{\mathcal{Z}}) = f(\dot{\mathcal{Z}}_1 \cdot_L \dot{\mathcal{Z}}_2 \cdot_L \ldots \cdot_L \dot{\mathcal{Z}}_N), \tag{14}$$

where function f is a trace operation on $\dot{\mathcal{Z}}(:, k, :)$, $k = 1, 2, \ldots, \prod_{i=1}^N I_i$, followed by a reshaping operation from vector of the length $\prod_{i=1}^N I_i$ to quaternion tensor of the size $I_1 \times I_2 \times \ldots \times I_N$.

Definition 8 (*Quaternion Tensor Permutation*) For any N-th order quaternion tensor $\dot{\mathcal{T}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$, the n-th quaternion tensor permutation is defined as $\dot{\mathcal{T}}^{P_n} \in \mathbb{H}^{I_n \times \ldots \times I_N \times I_1 \times \ldots \times I_{n-1}}$:

$$\dot{\mathcal{T}}^{P_n}(i_n, \ldots, i_N, i_1, \ldots, i_{n-1}) = \dot{\mathcal{T}}(i_1, i_2, \ldots, i_N). \tag{15}$$

[4] One can also use the right quaternion matrix multiplication to define the QTRR decomposition, which is $\dot{\mathcal{T}}(i_1, i_2, \ldots, i_N) = \mathrm{Tr}\{\dot{\mathcal{Z}}_1(i_1) \cdot_R \dot{\mathcal{Z}}_2(i_2) \cdot_R \ldots \cdot_R \dot{\mathcal{Z}}_N(i_N)\}$. Due to the analogous analysis process and application effects between QTRR and QTLR, we will exclusively consider QTLR throughout this paper.

In the following theorem, based on the definition of quaternion tensor permutation and QTLR decomposition, we present the cyclic permutation property of QTLR decomposition.

Theorem 1 *(Cyclic Permutation Property of QTLR Decomposition) The quaternion tensor permutation of $\dot{\mathcal{T}}$ is equivalent to its cores circularly shifting, as follows:*

$$\dot{\mathcal{T}}^{P_n} = f\big((\dot{\mathcal{Z}}_n \cdot_L \cdots \cdot_L \dot{\mathcal{Z}}_N) \cdot_R (\dot{\mathcal{Z}}_1 \cdot_L \cdots \cdot_L \dot{\mathcal{Z}}_{n-1})\big), \tag{16}$$

with elements

$$\dot{\mathcal{T}}^{P_n}(i_n, \ldots, i_N, i_1, \ldots, i_{n-1}) = \mathrm{Tr}\{\big(\dot{\mathcal{Z}}_n(i_n) \cdot_L \cdots \cdot_L \dot{\mathcal{Z}}_N(i_N)\big) \cdot_R \big(\dot{\mathcal{Z}}_1(i_1) \cdot_L \cdots \cdot_L \dot{\mathcal{Z}}_{n-1}(i_{n-1})\big)\}. \tag{17}$$

Proof

$$\begin{aligned}
&\dot{\mathcal{T}}^{P_n}(i_n, \ldots, i_N, i_1, \ldots, i_{n-1}) \\
&= \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_N=1}^{r_N} \dot{\mathcal{Z}}_1(\alpha_1, i_1, \alpha_2) \cdot_L \mathcal{Z}_2(\alpha_2, i_2, \alpha_3) \cdot_L \cdots \cdot_L \mathcal{Z}_N(\alpha_N, i_N, \alpha_{N+1}) \\
&= \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_N=1}^{r_N} \big(\dot{\mathcal{Z}}_1(\alpha_1, i_1, \alpha_2) \cdot_L \cdots \cdot_L \mathcal{Z}_{n-1}(\alpha_{n-1}, i_{n-1}, \alpha_n)\big) \cdot_L \\
&\quad \big(\mathcal{Z}_n(\alpha_n, i_n, \alpha_{n+1}) \cdot_L \cdots \cdot_L \mathcal{Z}_N(\alpha_N, i_N, \alpha_{N+1})\big) \\
&= \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_N=1}^{r_N} \big(\mathcal{Z}_n(\alpha_n, i_n, \alpha_{n+1}) \cdot_L \cdots \cdot_L \mathcal{Z}_N(\alpha_N, i_N, \alpha_{N+1})\big) \cdot_R \\
&\quad \big(\dot{\mathcal{Z}}_1(\alpha_1, i_1, \alpha_2) \cdot_L \cdots \cdot_L \mathcal{Z}_{n-1}(\alpha_{n-1}, i_{n-1}, \alpha_n)\big) \\
&= \mathrm{Tr}\{\big(\dot{\mathcal{Z}}_n(i_n) \cdot_L \cdots \cdot_L \dot{\mathcal{Z}}_N(i_N)\big) \cdot_R \big(\dot{\mathcal{Z}}_1(i_1) \cdot_L \cdots \cdot_L \dot{\mathcal{Z}}_{n-1}(i_{n-1})\big)\},
\end{aligned} \tag{18}$$

where the first equality holds due to (15) and (11), the second and third equalities hold directly as a result of the definitions of left and right multiplications between quaternion scalars. □

Note that, when quaternion tensors degenerate into real tensors, the cyclic permutation property will degenerate into its real counterpart as presented in [42]. Although, due to the non-commutativity of quaternion multiplication, there are significant differences in the cyclic permutation property between our defined QTLR decomposition and the TR decomposition in [42], they exhibit a similar form, which is why we refer to the decomposition of (10) as quaternion tensor left 'ring'.

In the following, we develop an algorithm to learn the QTLR format.

3.2 QTLR-QSVD Algorithm

Considering that exact quaternion tensor decompositions often demand extensive computational resources and storage, our focus shifts towards low-rank quaternion tensor approximation within the QTLR format. Inspired by the TR-SVD algorithm for TR decomposition in [42], we propose QTLR-QSVD algorithm for learning the QTLR format in this section. Before deriving the QTLR-QSVD algorithm, we first present a required definition and a theorem.

Definition 9 (*Quaternion Subchain Tensors*) Four quaternion subchain tensors are defined and denoted by

$$\begin{aligned}
\dot{\mathcal{Z}}^{<k} \in \mathbb{H}^{r_1 \times \prod_{n=1}^{k-1} I_n \times r_k} &= \dot{\mathcal{Z}}_1 \cdot_L \dot{\mathcal{Z}}_2 \cdot_L \cdots \cdot_L \dot{\mathcal{Z}}_{k-1}, \\
\dot{\mathcal{Z}}^{\leq k} \in \mathbb{H}^{r_1 \times \prod_{n=1}^{k} I_n \times r_{k+1}} &= \dot{\mathcal{Z}}_1 \cdot_L \dot{\mathcal{Z}}_2 \cdot_L \cdots \cdot_L \dot{\mathcal{Z}}_k, \\
\dot{\mathcal{Z}}^{>k} \in \mathbb{H}^{r_{k+1} \times \prod_{n=k+1}^{N} I_n \times r_1} &= \dot{\mathcal{Z}}_{k+1} \cdot_L \dot{\mathcal{Z}}_{k+2} \cdot_L \cdots \cdot_L \dot{\mathcal{Z}}_N, \\
\dot{\mathcal{Z}}^{\geq k} \in \mathbb{H}^{r_k \times \prod_{n=k}^{N} I_n \times r_1} &= \dot{\mathcal{Z}}_k \cdot_L \dot{\mathcal{Z}}_{k+1} \cdot_L \cdots \cdot_L \dot{\mathcal{Z}}_N.
\end{aligned} \tag{19}$$

Note that the lateral slice matrices of $\dot{\mathcal{Z}}^{<k}$ are $\dot{\mathcal{Z}}^{<k}(:,t,:) = \prod_{n=1}^{k-1} \dot{\mathcal{Z}}_n(i_n)$, where $t = \overline{i_1 i_2 \ldots i_{k-1}}$. Similar results can be obtained for $\dot{\mathcal{Z}}^{\leq k}$, $\dot{\mathcal{Z}}^{>k}$, and $\dot{\mathcal{Z}}^{\geq k}$.

Theorem 2 *Assume $\dot{\mathcal{T}}$ can be represented by a QTLR decomposition. Then,*

$$\dot{\mathbf{T}}_{\langle k \rangle} = \dot{\mathbf{Z}}_{(2)}^{\leq k} \cdot_L (\dot{\mathbf{Z}}_{[2]}^{>k})^T.$$

Proof Based on the definition of k-unfolding of quaternion tensor $\dot{\mathcal{T}}$, we can express the QTLR decomposition in the following form:

$$\begin{aligned}
\dot{\mathbf{T}}_{\langle k \rangle}(t_1, t_2) &= \mathrm{Tr}\{\dot{\mathcal{Z}}_1(i_1) \cdot_L \dot{\mathcal{Z}}_2(i_2) \cdot_L \cdots \cdot_L \dot{\mathcal{Z}}_N(i_N)\} \\
&= \mathrm{Tr}\left\{ \prod_{n=1}^{k} \dot{\mathcal{Z}}_n(i_n) \prod_{n=k+1}^{N} \dot{\mathcal{Z}}_n(i_n) \right\} \\
&= \mathrm{Tr}\left\{ \dot{\mathcal{Z}}^{\leq k}(:, t_1, :) \dot{\mathcal{Z}}^{>k}(:, t_2, :) \right\} \\
&= \mathrm{reshape}(\dot{\mathcal{Z}}^{\leq k}(:, t_1, :), [1, r_1 r_{k+1}]) \cdot_L \mathrm{reshape}((\dot{\mathcal{Z}}^{>k}(:, t_2, :))^T, [r_1 r_{k+1}, 1]) \\
&= \sum_{p=1}^{r_1 r_{k+1}} \dot{\mathbf{Z}}_{(2)}^{\leq k}(t_1, p) \cdot_L ((\dot{\mathbf{Z}}_{[2]}^{>k})^T)(p, t_2),
\end{aligned} \tag{20}$$

where $t_1 = \overline{i_1 i_2 \ldots i_k}$ and $t_2 = \overline{i_{k+1} i_{k+2} \ldots i_N}$. Thus, we have $\dot{\mathbf{T}}_{\langle k \rangle} = \dot{\mathbf{Z}}_{(2)}^{\leq k} \cdot_L (\dot{\mathbf{Z}}_{[2]}^{>k})^T$. □

Now, we present an algorithm that utilizes N sequential quaternion singular value decompositions (QSVDs) [39] for computing the QTLR decomposition. From Theorem 2, we have $\dot{\mathbf{T}}_{\langle 1 \rangle} = \dot{\mathbf{Z}}_{(2)}^{\leq 1} \cdot_L (\dot{\mathbf{Z}}_{[2]}^{>1})^T$, then we truncate the QSVD[5] of $\dot{\mathbf{T}}_{\langle 1 \rangle}$ to obtain its low-rank approximation, i.e., such that

$$\dot{\mathbf{T}}_{\langle 1 \rangle} = \dot{\mathbf{U}}_1 \Sigma_1 \dot{\mathbf{V}}_1^H + \dot{\varepsilon}_1. \tag{21}$$

Let $\dot{\mathbf{Z}}_{(2)}^{\leq 1} = \dot{\mathbf{U}}_1$ and $(\dot{\mathbf{Z}}_{[2]}^{>1})^T = \Sigma_1 \dot{\mathbf{V}}_1^H$, then the first core $\dot{\mathcal{Z}}_1$ and quaternion subchain tensor $\dot{\mathcal{Z}}^{>1}$ can be obtained by the proper reshaping and permutation of $\dot{\mathbf{U}}_1$ and $\Sigma_1 \dot{\mathbf{V}}_1^H$, respectively. Afterwards, let $\dot{\mathbf{Z}}^{>1} = \mathrm{reshape}(\dot{\mathcal{Z}}^{>1}, [r_2 I_2, \prod_{n=3}^{N} I_n r_1])$, then truncate the QSVD of $\dot{\mathbf{Z}}^{>1}$ to obtain its low-rank approximation, i.e., such that

$$\dot{\mathbf{Z}}^{>1} = \dot{\mathbf{U}}_2 \Sigma_2 \dot{\mathbf{V}}_2^H + \dot{\varepsilon}_2. \tag{22}$$

Then, the second core $\dot{\mathcal{Z}}_2$ and quaternion subchain tensor $\dot{\mathcal{Z}}^{>2}$ can be obtained by the proper reshaping of $\dot{\mathbf{U}}_2$ and $\Sigma_2 \dot{\mathbf{V}}_2^H$, respectively. This procedure can be carried out in a sequential

[5] The δ-truncated QSVD of a quaternion matrix $\dot{\mathbf{T}}$ means that we use a truncation threshold δ to truncate the singular values of $\dot{\mathbf{T}}$, retaining only those that are greater than or equal to δ^2. The symbol $\mathrm{rank}_\delta(\dot{\mathbf{T}})$ represents the number of singular values in $\dot{\mathbf{T}}$ that are greater than or equal to δ^2.

manner to acquire all N cores $\dot{\mathcal{Z}}_n, n = 1, 2, \ldots, N$. Similar to the TR-SVD algorithm [42], for QTLR-QSVD algorithm, we set the truncation threshold as

$$\delta_n = \begin{cases} \sqrt{2}\epsilon_p \|\dot{\mathcal{T}}\|_F / \sqrt{N}, & n = 1, \\ \epsilon_p \|\dot{\mathcal{T}}\|_F / \sqrt{N}, & n > 1, \end{cases} \tag{23}$$

where ϵ_p is a given prescribed relative error. The detailed procedure of the QTLR-QSVD algorithm is listed in Algorithm 1.

Algorithm 1 QTLR-QSVD

Input: An N-th order quaternion tensor $\dot{\mathcal{T}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ and the prescribed relative error ϵ_p.
Step1: Compute truncation threshold δ_n for $n = 1$ and $n > 1$ via (23).
Step2: Choose the first mode as the start point and obtain 1-unfolding quaternion matrix $\dot{\mathbf{T}}_{\langle 1 \rangle}$.
Step3: Low-rank approximation by applying δ_1-truncated QSVD: $\dot{\mathbf{T}}_{\langle 1 \rangle} = \dot{\mathbf{U}}_1 \Sigma_1 \dot{\mathbf{V}}_1^H + \dot{\varepsilon}_1$.
Step4: Split ranks r_1 and r_2 by: $\min\limits_{r_1, r_2} |r_1 - r_2|$, *s.t.* $r_1 r_2 = \text{rank}_{\delta_1}(\dot{\mathbf{T}}_{\langle 1 \rangle})$.
Step5: Obtain $\dot{\mathcal{Z}}_1$ via $\dot{\mathcal{Z}}_1 = \text{permute}(\text{reshape}(\dot{\mathbf{U}}_1, [I_1, r_1, r_2]), [2, 1, 3])$.
Step6: Obtain $\dot{\mathcal{Z}}^{>1}$ via $\dot{\mathcal{Z}}^{>1} = \text{permute}(\text{reshape}(\Sigma_1 \dot{\mathbf{V}}_1^H, [r_1, r_2, \prod_{n=2}^{N} I_n]), [2, 3, 1])$.
Step7: Perform the following iterative procedure:
for $n = 2$ to $N - 1$ **do**
  $\dot{\mathbf{Z}}^{>n-1} = \text{reshape}(\dot{\mathcal{Z}}^{>n-1}, [r_n I_n, \prod_{p=n+1}^{N} I_p r_1])$.
  Compute δ_n-truncated QSVD: $\dot{\mathbf{Z}}^{>n-1} = \dot{\mathbf{U}}_n \Sigma_n \dot{\mathbf{V}}_n^H + \dot{\varepsilon}_n$.
  $r_{n+1} = \text{rank}_{\delta_n}(\dot{\mathbf{Z}}^{>n-1})$.
  $\dot{\mathcal{Z}}_n = \text{reshape}(\dot{\mathbf{U}}_n, [r_n, I_n, r_{n+1}])$.
  $\dot{\mathcal{Z}}^{>n} = \text{reshape}(\Sigma_n \dot{\mathbf{V}}_n^H, [r_{n+1}, \prod_{p=n+1}^{N} I_p, r_1])$.
end for
Output: N cores $\dot{\mathcal{Z}}_n, n = 1, 2, \ldots, N$ of QTLR decomposition.

The QTLR-QSVD algorithm possesses inherent computational efficiency as a result of its non-recursive nature, enabling it to achieve a high degree of approximation for any given quaternion tensor. We know that the most intuitive application of rank truncation (or singular value truncation) is low-rank reconstruction of images. For matrices or quaternion matrices, methods such as truncated SVD or truncated QSVD [39] can be applied. However, for tensors (including quaternion tensors), since the definition of rank is not unique (depending on the tensor decomposition), the truncation of tensor singular values varies in form under different rank definitions. For example, low-rank reconstruction of images can be based on truncated higher-order singular value decomposition (HOSVD) [8, 23], truncated t-SVD decomposition [16, 23, 28], and truncated TT decomposition [25, 27], among others. In this section, to validate the feasibility of low-rank approximation using our proposed QTLR-QSVD algorithm and the superiority of introducing quaternions, we compared the effects of QTLR-QSVD and TR-SVD on low-rank reconstruction of color images (as shown in Fig. 1), that is, comparing their reconstruction effects under the same truncation threshold to determine which performs better.

Based on the comparison results, we can conclude that the incorporation of quaternions enables QTLR-QSVD to achieve better performance in the reconstruction of color images compared to TR-SVD. Similar results can be obtained for other color images as well.

Note that the non-commutativity of quaternion multiplication prevents the learning algorithm for the QTLR model from achieving the same level of richness as the algorithm for the TR model, which is essentially a degenerate version of the QTLR model in the real number

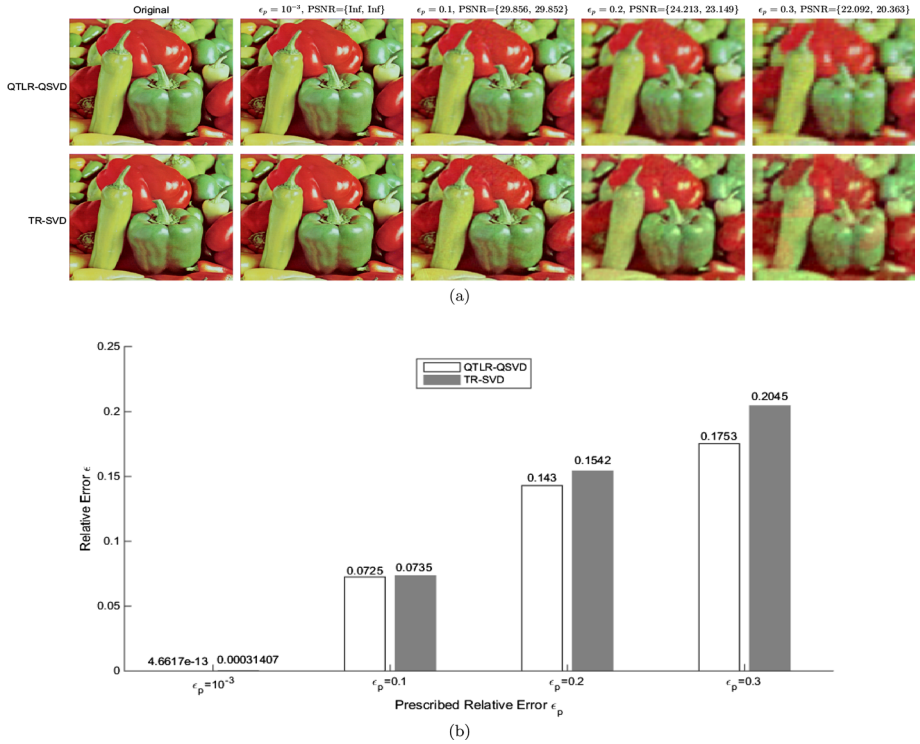

Fig. 1 The reconstruction of one color image 'peppers' by using QTLR-QSVD and TR-SVD. The color image is tensorized to 9th-order quaternion tensor $4 \times 4 \times \ldots \times 4$ (10th-order tensor $4 \times 4 \times \ldots \times 4 \times 3$ for TR-SVD) by the OKA procedure (see Sect. 5.1). **a** The reconstruction results are displayed for different prescribed relative errors (the first and second terms of PSNR correspond to QTLR-QSVD and TR-SVD, respectively). **b** The reconstruction relative errors are shown for different prescribed relative errors

domain. For instance, the TR-ALS series algorithms used for learning the TR model in [42] cannot be applied to the learning of the QTLR model.[6]

As mentioned in the introduction section, the introduction of the QTLR model primarily aims to combine the structural advantages of the TR model with the benefits of quaternion representation for color pixels. It is anticipated that methods based on the QTLR decomposition will yield improved results in tasks related to color image processing. In this paper, we will use the example of color image inpainting based on an LRQTC model as an application case to validate this assertion.

4 Low-Rank Quaternion Tensor Completion

In this section, we will propose an LRQTC model and its corresponding optimization algorithm based on the previously defined QTLR decomposition and QTLR-rank.

[6] In fact, one can verify that based on our definition of QTLR and its satisfied cyclic permutation property, the inability of the TR-ALS algorithms [42] to be applied to learning the QTLR format primarily stems from property (8).

4.1 LRQTC Model Based on QTLR Weighted Nuclear Norm Minimization

In order to introduce the QTLR weighted nuclear norm formulation, we first define the circular unfolding of a quaternion tensor and then theoretically establish its connection to the QTLR-rank. This method does not rely on a pre-specified QTLR-rank, thus transforming the problem of approximating higher-order quaternion tensors with a low QTLR-rank into a low-rank approximation problem of quaternion matrices.

Definition 10 (*Quaternion Tensor Circular Unfolding*) Let $\dot{\mathcal{T}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ be an N-th order quaternion tensor, its circular unfolding is a quaternion matrix, denoted by $\dot{\mathbf{T}}_{\{k,l\}}$, which first permutes $\dot{\mathcal{T}}$ with order $[k, \ldots, N, 1 \ldots, k-1]$ and then performs matricization along the first l modes, i.e., l-unfolding. The indices of $\dot{\mathbf{T}}_{\{k,l\}}(p, q)$ are formulated as

$$\dot{\mathbf{T}}_{\{k,l\}}(p, q) = \dot{\mathcal{T}}(i_1, i_2, \ldots, i_N), \tag{24}$$

where $p = 1 + \sum_{s=k}^{k+l-1} (i_s - 1) \prod_{t=k}^{s-1} I_t$ and $q = 1 + \sum_{s=k+l}^{k-1} (i_s - 1) \prod_{t=k+l}^{s-1} I_t$. Additionally, we use $\text{fold}_{\{k,l\}}(\dot{\mathbf{T}}_{\{k,l\}})$ to denote the inverse process of quaternion tensor circular unfolding.

Note that when $l = 1$, the quaternion tensor circular unfolding is reduced to the quaternion tensor mode-k unfolding of $\dot{\mathcal{T}}$, i.e., $\dot{\mathbf{T}}_{\{k,1\}} = \dot{\mathbf{T}}_{[k]}$.

Theorem 3 *Assume that $\dot{\mathcal{T}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ is an N-th order quaternion tensor with QTLR-rank $\mathbf{r} = [r_1, r_2, \ldots, r_N]$, and then when $l = N - k + 1$, for each $\dot{\mathbf{T}}_{\{k,l\}}$, we have*

$$\text{rank}(\dot{\mathbf{T}}_{\{k,l\}}) \leq r_k r_{k+l}. \tag{25}$$

Proof From (10) and (24), $\dot{\mathbf{T}}_{\{k,l\}}(p, q)$ can be represented in the index form, that is,

$$\dot{\mathbf{T}}_{\{k,l\}}(p, q) = \dot{\mathcal{T}}(i_1, i_2, \ldots, i_N) = \text{Tr}\{\dot{\mathcal{Z}}_1(i_1) \cdot_L \dot{\mathcal{Z}}_2(i_2) \cdot_L \ldots \cdot_L \dot{\mathcal{Z}}_N(i_N)\}. \tag{26}$$

Based on Definition 8 and Theorem 1, (26) can be rewritten as

$$\dot{\mathbf{T}}_{\{k,l\}}(p, q) = \text{Tr}\{\left(\dot{\mathcal{Z}}_k(i_k) \cdot_L \ldots \cdot_L \dot{\mathcal{Z}}_N(i_N)\right) \cdot_R \left(\dot{\mathcal{Z}}_1(i_1) \cdot_L \ldots \cdot_L \dot{\mathcal{Z}}_{k-1}(i_{k-1})\right)\}. \tag{27}$$

since $l = N - k + 1$, (27) can be further rewritten as

$$\begin{aligned}
\dot{\mathbf{T}}_{\{k,l\}}(p, q) &= \text{Tr}\{\left(\dot{\mathcal{Z}}_k(i_k) \cdot_L \ldots \cdot_L \dot{\mathcal{Z}}_{k+l-1}(i_{k+l-1})\right) \cdot_R \left(\dot{\mathcal{Z}}_1(i_1) \cdot_L \ldots \cdot_L \dot{\mathcal{Z}}_{k-1}(i_{k-1})\right)\} \\
&= \text{Tr}\{\dot{\mathcal{W}}(:, p, :) \cdot_R \dot{\mathcal{H}}(:, q, :)\} \\
&= \sum_{\alpha_1=1}^{r_k} \sum_{\alpha_2=1}^{r_{l+k}} \dot{\mathcal{W}}(\alpha_1, p, \alpha_2) \cdot_R \dot{\mathcal{H}}(\alpha_2, q, \alpha_1) \\
&= \sum_{\beta=1}^{r_k r_{l+k}} \dot{\mathbf{W}}_{(2)}(p, \beta) \cdot_R \dot{\mathbf{H}}_{[2]}^T(\beta, q) \\
&= \sum_{\beta=1}^{r_k r_{l+k}} \dot{\mathbf{H}}_{[2]}^T(\beta, q) \dot{\mathbf{W}}_{(2)}(p, \beta),
\end{aligned} \tag{28}$$

where $\dot{\mathcal{W}} \in \mathbb{H}^{r_k \times \prod_{n=k}^{N} I_n \times r_{l+k}} = \dot{\mathcal{Z}}_k \cdot_L \ldots \cdot_L \dot{\mathcal{Z}}_{k+l-1}$, $\dot{\mathcal{H}} \in \mathbb{H}^{r_1 \times \prod_{n=1}^{k-1} I_n \times r_k} = \dot{\mathcal{Z}}_1 \cdot_L \ldots \cdot_L \dot{\mathcal{Z}}_{k-1}$. According to (28), we can get that $\dot{\mathbf{T}}_{\{k,l\}} = \dot{\mathbf{W}}_{(2)} \cdot_R \dot{\mathbf{H}}_{[2]}^T$, which combining Lemma 1 means $\text{rank}(\dot{\mathbf{T}}_{\{k,l\}}) \leq \min(\text{rank}(\dot{\mathbf{W}}_{(2)}, \text{rank}(\dot{\mathbf{H}}_{[2]}^T)) \leq \min(r_k r_{k+l}, \prod_{n=k}^{N} I_n, \prod_{n=1}^{k-1} I_n)$. Note that $r_{k+l} = r_{N+1} = r_1$. □

4.1.1 The Proposed Model

From Theorem 3, we can observe that for an arbitrary N-th order quaternion tensor with QTLR-rank $\mathbf{r} = [r_1, r_2, \ldots, r_N]$, the rank of each circular unfolding quaternion matrix $\dot{\mathbf{T}}_{\{k,l\}}$ with $l = N - k + 1$ is bounded by $r_k r_{k+l}$. This indicates that the low QTLR-rank property of quaternion tensor $\dot{\mathcal{T}}$ is maintained in a small subspace of $\dot{\mathbf{T}}_{\{k,l\}}$ for $k = 1, 2, \ldots, N$. In other words, if $\dot{\mathcal{T}}$ has a low QTLR-rank, meaning that all the components of $\mathbf{r} = [r_1, r_2, \ldots, r_N]$ are small, then the rank of $\dot{\mathbf{T}}_{\{k,l\}}$ will also be small, indicating that $\dot{\mathbf{T}}_{\{k,l\}}$ is also of low rank. Hence, to minimize QTLR-rank, a natural option is to consider the sum of rank of circular unfolding quaternion matrices:

$$\min_{\dot{\mathcal{T}}} \sum_{k=2}^{N} \alpha_k \text{rank}(\dot{\mathbf{T}}_{\{k,l\}}), \tag{29}$$

where α_k for $k = 2, 3, \ldots, N$ are positive parameters satisfying $\sum_{k=2}^{N} \alpha_k = 1$. Note that k starts from 2, because when $k = 1$, there is no permutation for $\dot{\mathcal{T}}$. Nevertheless, the general computational complexity of problem (29) makes it intractable. To address the solvability of (29), a convex surrogate, the sum of weighted nuclear norm, has been adopted. The definition of this surrogate is provided as follows.

Definition 11 (*QTLR Weighted Nuclear Norm*) Assume the quaternion tensor $\dot{\mathcal{T}}$ with QTLR decomposition, its QTLR weighted nuclear norm is defined as

$$\sum_{k=2}^{N} \alpha_k \|\dot{\mathbf{T}}_{\{k,l\}}\|_{w,*}, \tag{30}$$

where $l = N - k + 1$.

Folowing Theorem 3, we constrain $l = N - k + 1$ in our defined QTLR weighted nuclear norm, and we perform permutations on $\dot{\mathcal{T}}$ for $k = 2$ to $k = N$. Therefore, these $N - 1$ differently sized and permuted quaternion matrices allow for a more comprehensive capture of the low-rank structure of $\dot{\mathcal{T}}$ and the global information of the quaternion data.

Based on the defined QTLR weighted nuclear norm (30), we propose the following LRQTC model:

$$\begin{aligned} &\min_{\dot{\mathcal{T}}} \sum_{k=2}^{N} \alpha_k \|\dot{\mathbf{T}}_{\{k,l\}}\|_{w,*} \\ &\text{s.t. } P_\Omega(\dot{\mathcal{T}}) = P_\Omega(\dot{\mathcal{X}}), \end{aligned} \tag{31}$$

where $\dot{\mathcal{T}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ is a completed output N-th order quaternion tensor, $\dot{\mathcal{X}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ is the observed N-th order quaternion tensor, and $P_\Omega(\cdot)$ is the projection operator on Ω which is the index of observed elements. Specifically,

$$\mathcal{P}_\Omega(\dot{\mathcal{T}}) = \begin{cases} \dot{\mathcal{T}}(i_1, i_2, \ldots, i_N), & (i_1, i_2, \ldots, i_N) \in \Omega, \\ 0, & \text{otherwise.} \end{cases}$$

4.1.2 Numercial Scheme to Solve the LRQTC Model

To enable the solution of (31), we use the variable-splitting technique and introduce auxiliary quaternion tensors $\{\dot{\mathcal{M}}^{(k)}\}_{k=2}^{N} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ in (31). Consequently, (31) is finally transformed into the following solvable model:

$$\begin{aligned} &\min_{\dot{\mathcal{T}},\{\dot{\mathcal{M}}^{(k)}\}} \sum_{k=2}^{N} \alpha_k \|\dot{\mathbf{M}}_{\{k,l\}}^{(k)}\|_{w,*} \\ &\text{s.t. } \dot{\mathcal{T}} = \dot{\mathcal{M}}^{(k)},\ k = 2, 3, \ldots, N, \\ &\quad P_\Omega(\dot{\mathcal{T}}) = P_\Omega(\dot{\mathcal{X}}). \end{aligned} \tag{32}$$

Based on the ADMM framework in the quaternion domain [21], the augmented Lagrangian function of (32) is defined as

$$\begin{aligned} \mathcal{L}_\mu(\dot{\mathcal{X}}, \{\dot{\mathcal{M}}^{(k)}\}_{k=2}^N, \{\dot{\mathcal{Y}}^{(k)}\}_{k=2}^N) = &\sum_{k=2}^{N} \alpha_k \|\dot{\mathbf{M}}_{\{k,l\}}^{(k)}\|_{w,*} + \Re(\langle \dot{\mathcal{Y}}^{(k)}, \dot{\mathcal{T}} - \dot{\mathcal{M}}^{(k)} \rangle) \\ &+ \frac{\mu_k}{2} \|\dot{\mathcal{T}} - \dot{\mathcal{M}}^{(k)}\|_F^2 \\ \text{s.t. } P_\Omega(\dot{\mathcal{T}}) = &P_\Omega(\dot{\mathcal{X}}), \end{aligned} \tag{33}$$

where $\dot{\mathcal{Y}}^{(k)} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ for $k = 2, 3, \ldots, N$ are Lagrange Multipliers, $\mu_k > 0$ for $k = 2, 3, \ldots, N$ are penalty parameters. Then, we use an iterative scheme to solve the problem (33).

Update $\dot{\mathcal{M}}^{(k)}$: To optimize $\dot{\mathcal{M}}^{(k)}$ is equivalent to solve the subproblem:

$$\begin{aligned} \dot{\mathcal{M}}^{(k)} &= \arg\min_{\dot{\mathcal{M}}^{(k)}} \ \alpha_k \|\dot{\mathbf{M}}_{\{k,l\}}^{(k)}\|_{w,*} + \Re(\langle \dot{\mathcal{Y}}^{(k)}, \dot{\mathcal{T}} - \dot{\mathcal{M}}^{(k)} \rangle) + \frac{\mu_k}{2} \|\dot{\mathcal{T}} - \dot{\mathcal{M}}^{(k)}\|_F^2 \\ &= \arg\min_{\dot{\mathcal{M}}^{(k)}} \ \frac{\alpha_k}{\mu_k} \|\dot{\mathbf{M}}_{\{k,l\}}^{(k)}\|_{w,*} + \frac{1}{2} \|\dot{\mathcal{M}}^{(k)} - (\dot{\mathcal{T}} + \frac{\dot{\mathcal{Y}}^{(k)}}{\mu_k})\|_F^2. \end{aligned} \tag{34}$$

Denote $\dot{\Gamma} = \dot{\mathcal{T}} + \frac{\dot{\mathcal{Y}}^{(k)}}{\mu_k}$ and let $\dot{\Gamma} = \dot{\mathbf{U}} \Sigma \dot{\mathbf{V}}^H$ be the QSVD of $\dot{\Gamma}$, where

$$\Sigma = \begin{bmatrix} \operatorname{diag}\big(\sigma_1(\dot{\Gamma}), \ldots, \sigma_s(\dot{\Gamma})\big) \\ \mathbf{0} \end{bmatrix},$$

and $\sigma_n(\dot{\Gamma})$ is the n-th singular value of $\dot{\Gamma}$, s denotes the number of nonzero singular values of $\dot{\Gamma}$. From [37], the problem (34) has the following closed-form solution:

$$\dot{\mathcal{M}}^{(k)} = \operatorname{fold}_{\{k,l\}}(\dot{\mathbf{U}} \hat{\Sigma} \dot{\mathbf{V}}^H), \tag{35}$$

where

$$\hat{\Sigma} = \begin{bmatrix} \operatorname{diag}\big(\sigma_1(\dot{\mathbf{M}}_{\{k,l\}}^{(k)}), \ldots, \sigma_s(\dot{\mathbf{M}}_{\{k,l\}}^{(k)})\big) \\ \mathbf{0} \end{bmatrix},$$

and $\sigma_n(\dot{\mathbf{M}}_{\{k,l\}}^{(k)}) = \begin{cases} 0, & \text{if } c_2 < 0 \\ \frac{c_1 + \sqrt{c_2}}{2}, & \text{if } c_2 \geq 0 \end{cases}$, with $c_1 = \sigma_n(\dot{\Gamma}) - \epsilon$, $c_2 = (\sigma_n(\dot{\Gamma}) + \epsilon)^2 - 4C$, and C is a compromising constant.

Update $\dot{\mathcal{T}}$: To optimize $\dot{\mathcal{T}}$ is equivalent to solve the subproblem:

$$\begin{aligned} \dot{\mathcal{T}} &= \arg\min_{P_\Omega(\dot{\mathcal{T}}) = P_\Omega(\dot{\mathcal{X}})} \ \sum_{k=2}^{N} \Re(\langle \dot{\mathcal{Y}}_k, \dot{\mathcal{T}} - \dot{\mathcal{M}}^{(k)} \rangle) + \frac{\mu_k}{2} \|\dot{\mathcal{T}} - \dot{\mathcal{M}}^{(k)}\|_F^2 \\ &= \arg\min_{P_\Omega(\dot{\mathcal{X}}) = P_\Omega(\dot{\mathcal{T}})} \ \sum_{k=2}^{N} \frac{\mu_k}{2} \|\dot{\mathcal{T}} - \dot{\mathcal{M}}^{(k)} + \frac{\dot{\mathcal{Y}}^{(k)}}{\mu_k}\|_F^2 \end{aligned} \tag{36}$$

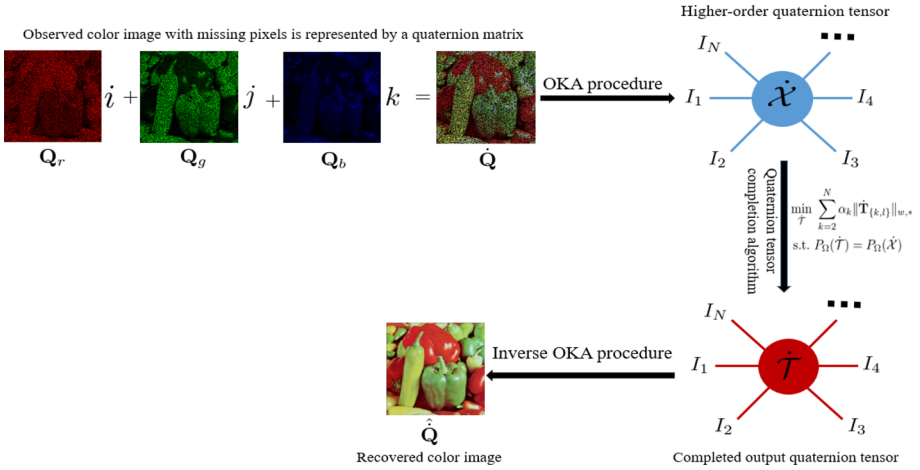

Fig. 2 The entire process of color image inpainting

It is easy to check that the solution of (36) is given by:

$$\dot{\mathcal{T}} = P_{\Omega^c}\left(\frac{\sum_{k=2}^{N}\left(\dot{\mathcal{M}}^{(k)} - \frac{\dot{\mathcal{Y}}^{(k)}}{\mu_k}\right)}{N-1}\right) + P_{\Omega}(\dot{\mathcal{X}}), \tag{37}$$

where Ω^c is the complement of Ω.

Update $\dot{\mathcal{Y}}^{(k)}$: The Lagrange multiplier $\dot{\mathcal{Y}}^{(k)}$ is updated by:

$$\dot{\mathcal{Y}}^{(k)} = \dot{\mathcal{Y}}^{(k)} + \mu_k(\dot{\mathcal{T}} - \dot{\mathcal{M}}^{(k)}). \tag{38}$$

To speed up convergence, each iteration we also update μ_k by: $\mu_k = \min(\mu_{max}, \rho\mu_k)$, where μ_{max} is the default maximum of μ_k, $\rho > 1$ is a constant parameter.

Finally, the proposed LRQTC algorithm is summarized in Algorithm 2.

Algorithm 2 Our proposed LRQTC algorithm.

Input: The observed N-th order quaternion tensor $\dot{\mathcal{T}} \in \mathbb{H}^{I_1 \times I_2 \times \ldots \times I_N}$ with Ω (the index of observed elements), $\{\alpha_k\}_{k=2}^N$, $\mu_{\max}$ and ρ.
Initialize $\{\dot{\mathcal{M}}^{(k)}\}_{k=2}^N$, $\{\dot{\mathcal{Y}}^{(k)}\}_{k=2}^N$, and $\{\mu_k\}_{k=2}^N$.
Repeat
for $k = 2$ to N **do**
Update $\dot{\mathcal{M}}^{(k)}$ via (35);
end for
Update $\dot{\mathcal{T}}$ via (37) (the updated one is labled by $\widetilde{\dot{\mathcal{T}}}$).
for $k = 2$ to N **do**
Update $\dot{\mathcal{Y}}^{(k)}$ via (38);
Update μ_k via $\mu_k = \min(\mu_{max}, \rho\mu_k)$.
end for
Until $\frac{\|\dot{\mathcal{T}} - \widetilde{\dot{\mathcal{T}}}\|_F}{\|\widetilde{\dot{\mathcal{T}}}\|_F} < 10^{-5}$ or reach the preset maximum number of iterations.
Output: The recovered quaternion tensor $\widetilde{\dot{\mathcal{T}}}$.

Fig. 3 The tested natural color images (the first row), color medical images (the second row), and color face images (the third row)

Table 1 Average PSNR and SSIM values (PSNR, SSIM) on the five **natural color images** with five levels of sampling rates (SRs) (**bold** fonts denote the best performance)

Methods	SRs				
	SR=10%	SR=20%	SR=30%	SR=40%	SR=50%
t-SVD [41]	17.266, 0.638	20.206, 0.766	22.639, 0.840	24.854, 0.891	27.176, 0.929
TMac-TT [1]	20.480, 0.778	22.415, 0.847	24.651, 0.903	26.229, 0.931	28.277, 0.956
TRLRF [38]	17.374, 0.633	20.324, 0.761	23.217, 0.851	25.609, 0.903	27.997, 0.940
TRNNM [11]	19.683, 0.801	22.743, 0.880	24.938, 0.920	26.841, 0.945	28.778, 0.963
LRQA-2 [5]	18.063, 0.663	20.950, 0.781	23.241, 0.848	25.337, 0.894	27.545, 0.928
LRQMC [22]	17.738, 0.677	20.838, 0.797	23.367, 0.863	25.589, 0.908	27.976, 0.942
TQLNA [35]	17.819, 0.658	21.124, 0.788	23.632, 0.859	25.870, 0.905	28.252, 0.939
Ours	**23.276**, **0.881**	**25.933**, **0.929**	**27.948**, **0.953**	**29.679**, **0.968**	**31.310**, **0.977**

5 Experiments and Results

In this section, we will first elaborate on how to utilize the proposed LRQTC model for color image inpainting and then present the experimental results.

5.1 Color Image Inpainting

Because a color image is essentially a quaternion matrix (a second-order quaternion tensor), it is necessary to increase the order of the quaternion matrix in order to effectively utilize the proposed LRQTC method. Recently, the overlapping ket augmentation (OKA) as a tensor augmentation technique was developed in [40] for increasing the order of tensors. OKA is an improvement upon KA [1] as it overcomes the visual flaws caused by reshaping and eliminates the blocking artifacts introduced by KA. Therefore, in order to increase the order of quaternion matrices used for representing color images, we apply OKA to quaternion matrices. Due to the similarity in the process of applying OKA to tensors [40] and quaternion

Table 2 Average PSNR and SSIM values (PSNR, SSIM) on the five **color medical images** with five levels of sampling rates (SRs) (**bold** fonts denote the best performance)

Methods	SRs				
	SR=10%	SR=20%	SR=30%	SR=40%	SR=50%
t-SVD [41]	17.568, 0.575	19.816, 0.688	21.684, 0.761	23.383, 0.814	25.013, 0.856
TMac-TT [1]	20.139, 0.714	21.980, 0.806	23.937, 0.869	25.668, 0.910	27.192, 0.934
TRLRF [38]	17.142, 0.233	19.720, 0.420	21.686, 0.560	23.221, 0.638	24.771, 0.712
TRNNM [11]	17.982, 0.639	21.340, 0.797	23.884, 0.875	26.142, 0.922	28.610, 0.957
LRQA-2 [5]	18.099, 0.590	20.366, 0.695	22.063, 0.760	23.662, 0.811	25.271, 0.851
LRQMC [22]	17.687, 0.591	20.025, 0.702	21.932, 0.775	23.586, 0.826	25.449, 0.871
TQLNA [35]	17.899, 0.597	20.589, 0.714	22.350, 0.776	23.937, 0.824	25.527, 0.863
Ours	**22.339**, **0.836**	**24.553**, **0.894**	**26.111**, **0.923**	**27.608**, **0.944**	**29.160**, **0.960**

Table 3 Average PSNR and SSIM values (PSNR, SSIM) on the five **color face images** with five levels of sampling rates (SRs) (**bold** fonts denote the best performance)

Methods	SRs				
	SR=10%	SR=20%	SR=30%	SR=40%	SR=50%
t-SVD [41]	18.416, 0.789	22.761, 0.885	25.984, 0.932	28.515, 0.955	31.317, 0.972
TMac-TT [1]	22.862, 0.890	28.633, 0.964	31.093, 0.978	32.952, 0.984	34.534, 0.988
TRLRF [38]	18.424, 0.784	22.422, 0.867	25.248, 0.919	27.554, 0.945	29.689, 0.962
TRNNM [11]	22.452, 0.916	26.878, 0.958	29.508, 0.974	31.729, 0.982	33.940, 0.989
LRQA-2 [5]	20.021, 0.817	24.148, 0.901	26.908, 0.940	29.323, 0.960	31.674, 0.973
LRQMC [22]	19.172, 0.819	23.216, 0.897	26.516, 0.938	28.423, 0.956	31.132, 0.971
TQLNA [35]	19.585, 0.809	24.160, 0.900	27.347, 0.943	29.902, 0.965	32.901, 0.979
Ours	**26.708**, **0.953**	**30.050**, **0.976**	**32.332**, **0.984**	**34.288**, **0.989**	**35.990**, **0.992**

Table 4 The average running time (seconds) of different algorithms when achieving their respective highest PSNR values, for two levels of sampling rates (SRs)

Color images	SRs					
	SR=10%			SR=50%		
Methods	Natural	Medical	Face	Natural	Medical	Face
t-SVD [41]	157.70	141.91	120.46	178.86	175.02	121.62
TMac-TT [1]	4.36	20.89	7.21	4.99	15.80	14.58
TRLRF [38]	298.14	330.09	29.76	323.99	272.36	34.90
TRNNM [11]	304.42	270.83	30.31	273.07	266.56	30.28
LRQA-2 [5]	202.21	225.97	91.49	111.51	162.85	52.67
LRQMC [22]	50.32	44.08	9.90	73.09	62.56	11.43
TQLNA [35]	48.40	45.02	8.23	72.90	62.34	11.09
Ours	341.86	329.02	62.16	342.44	334.31	63.36

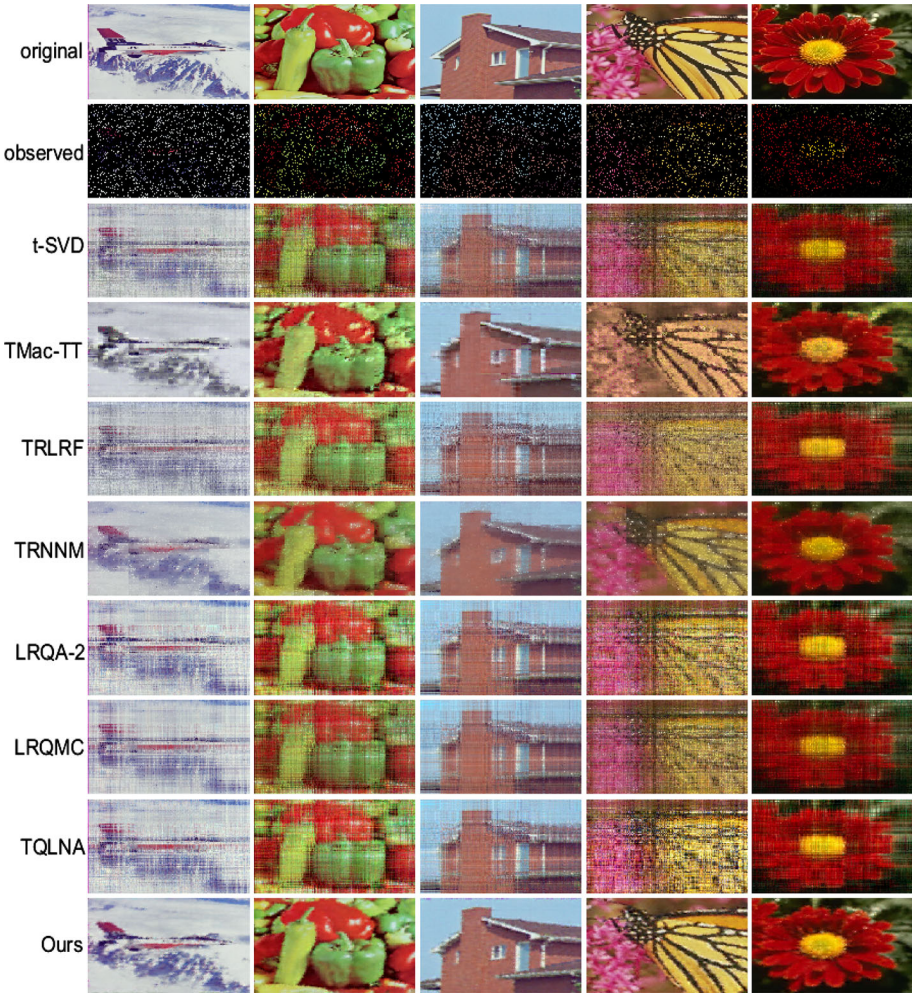

Fig. 4 Recovered natural color images for random missing with $SR = 10\%$. From the top row to the bottom row: the original color images, the observed color images, the results recovered by t-SVD, TMac-TT, TRLRF, TRNNM, LRQA-2, LRQMC, TQLNA, and our method, respectively. **The figure is viewed better in zoomed PDF**

matrices, we will not reiterate it here. Finally, we summarize the proposed entire process of color image inpainting in Fig. 2.

5.2 Experimental Results

To validate the effectiveness of our color image inpainting method, we conducted extensive experiments using a diverse range of images, including natural color images, color medical images, and color face images. We compare our proposed method with several classic and state-of-the-art quaternion matrix and tensor completion methods, including t-SVD [41], TMac-TT [1], TRLRF [38], TRNNM [11], LRQA-2 [5], LRQMC [22], and TQLNA [35].

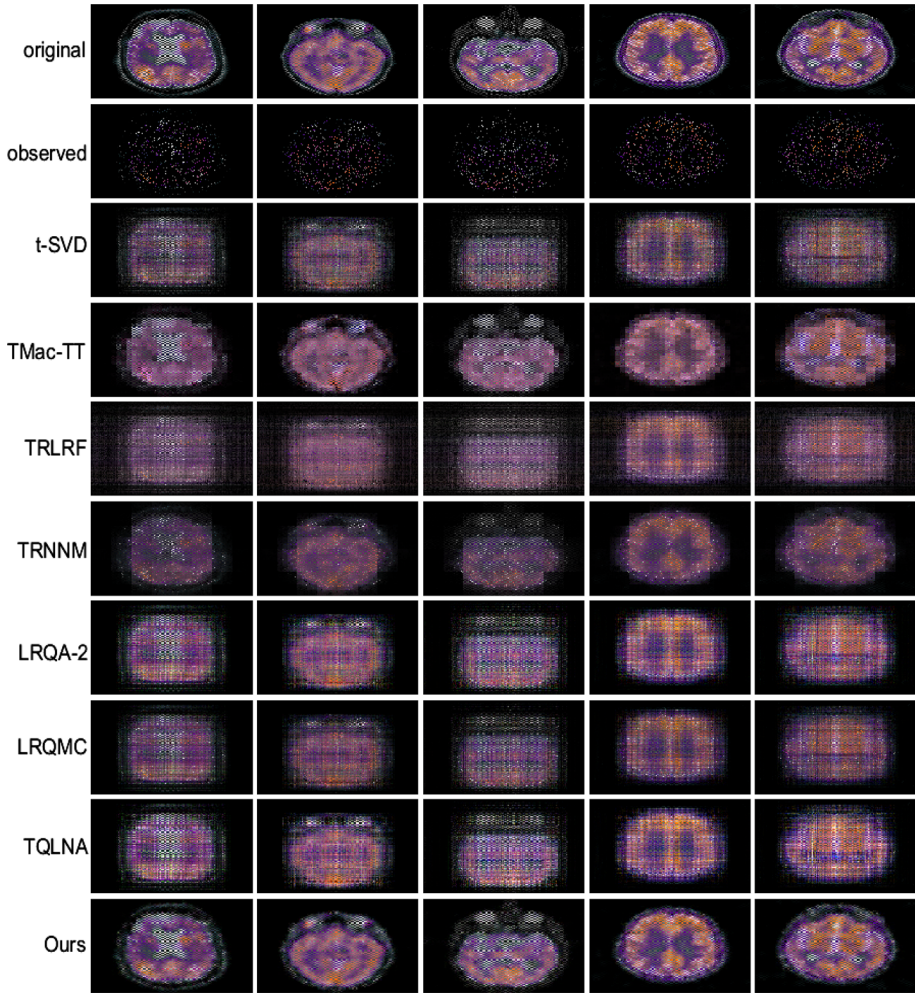

Fig. 5 Recovered color medical images for random missing with $SR = 10\%$. From the top row to the bottom row: the original color images, the observed color images, the results recovered by t-SVD, TMac-TT, TRLRF, TRNNM, LRQA-2, LRQMC, TQLNA, and our method, respectively. **The figure is viewed better in zoomed PDF**

In order to assess the performance of the proposed method, we considered not only visual quality but also utilized two commonly used quantitative quality metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [33]. All the experiments are run in MATLAB 2014*b* under Windows 10 on a personal computer with a $1.60GHz$ CPU and $8GB$ memory.

Natural color image inpainting: Five natural color images (shown in the first row of Fig. 3) with a spatial resolution of 256×256 are utilized for the evaluation. For our proposed method, the natural color images are transformed into ninth-order quaternion tensors of size $4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4$ using OKA. We set $\alpha_k = \frac{\omega_k}{\sum_{k=2}^{N} \omega_k}$ with $\omega_k = \min(\prod_{n=1}^{k-1} I_n, \prod_{n=k}^{N} I_n)$ for $k = 2, 3, \ldots, N$, $\mu_{\max} = 10^6$, and

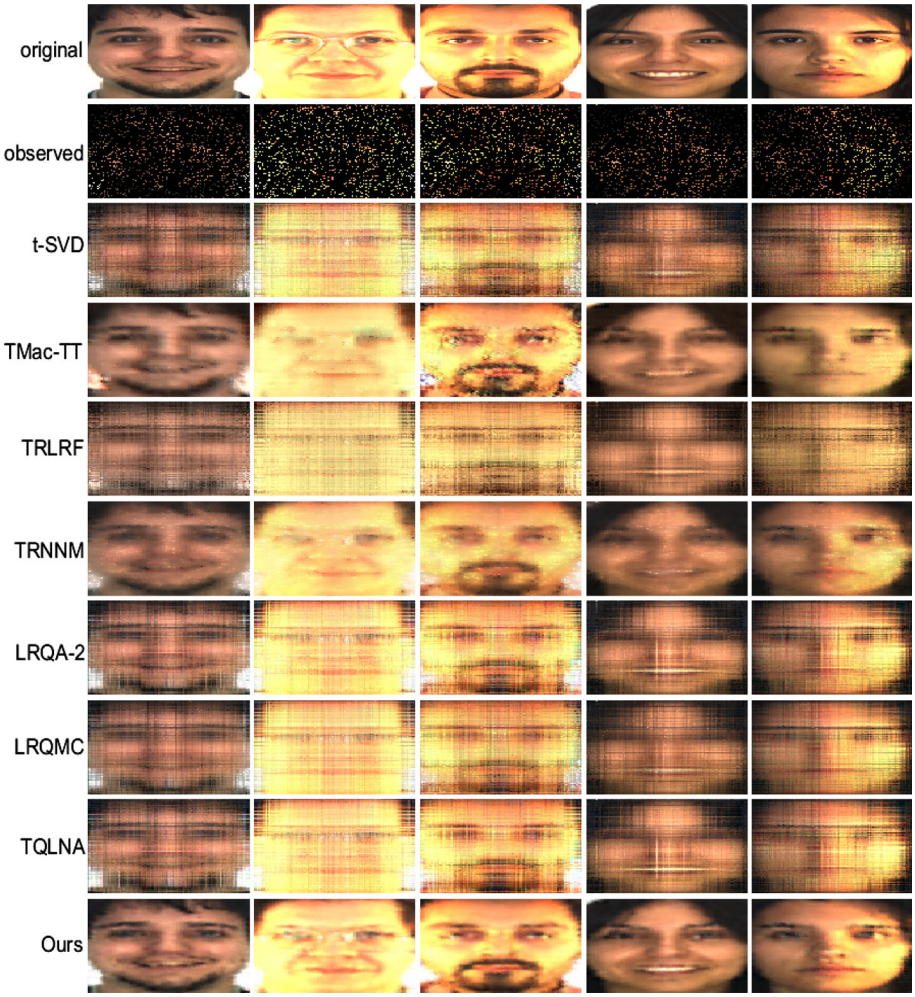

Fig. 6 Recovered color face images for random missing with $SR = 10\%$. From the top row to the bottom row: the original color images, the observed color images, the results recovered by t-SVD, TMac-TT, TRLRF, TRNNM, LRQA-2, LRQMC, TQLNA, and our method, respectively. **The figure is viewed better in zoomed PDF**

$\rho = 1.03$. We initialize $\dot{\mathcal{M}}^{(k)} = \dot{\mathcal{T}}$, $\dot{\mathcal{Y}}^{(k)} = \mathbf{0}$ for $k = 2, 3, \ldots, N$, and $\mu = \{0.5, 0.5, 0.001, 10^{-4.1}, 10^{-4.1}, 0.001, 0.5, 0.5\}$. Furthermore, all the compared methods were implemented using their source codes, and the parameter configurations were set according to the recommendations provided in the original papers, with adjustments made to optimize performance as closely as possible.

Color medical image inpainting: Five color medical images (shown in the second row of Fig. 3) with a spatial resolution of 256×256 are utilized for the evaluation. The experimental settings are the same as those for natural color image inpainting.

Color face image inpainting: Five color face images (shown in the third row of Fig. 3) with a spatial resolution of 120×165 are utilized for the evaluation. For our proposed method, the color face images are transformed into eighth-order quaternion tensors of size

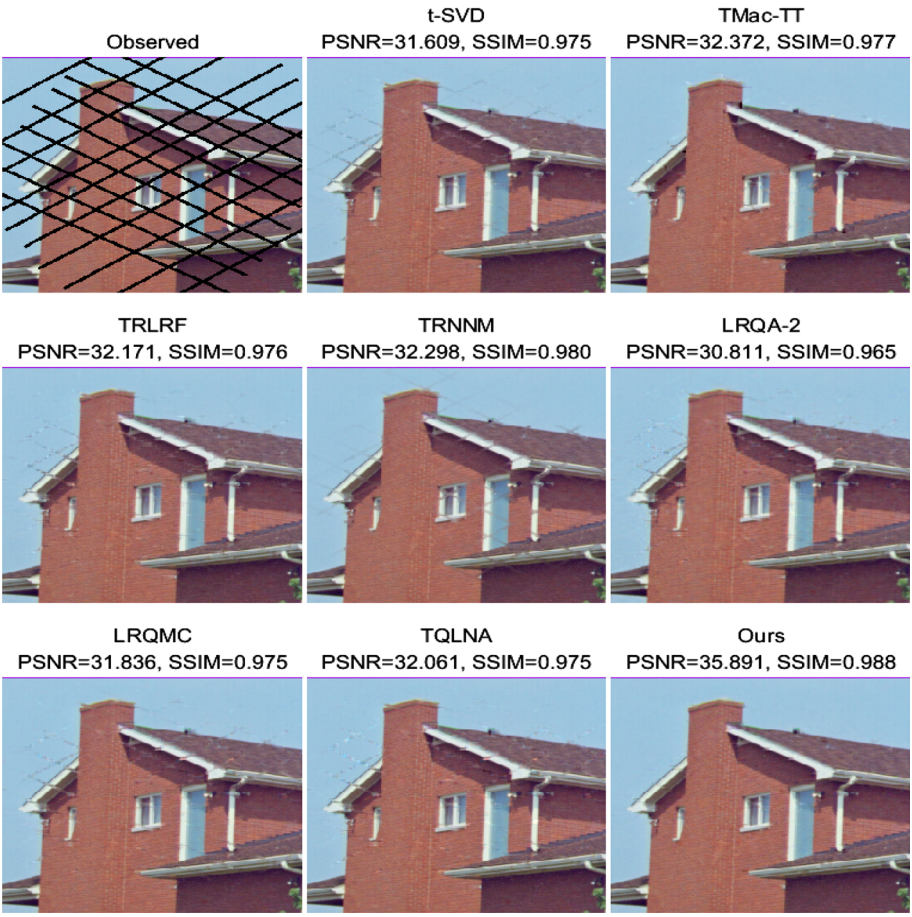

Fig. 7 Recovery of structurally missing natural color image: a comparison of various methods in terms of visual and quantitative metrics. **The figure is viewed better in zoomed PDF**

$4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 5 \times 4$ using OKA. The other experimental settings are the same as those for natural color image inpainting.

For random missing, we set five levels of sampling rates (SRs) which are SR $= 10\%$, SR $= 20\%$, SR $= 30\%$, SR $= 40\%$, and SR $= 50\%$. The average PSNR and SSIM values are reported in Tables 1, 2, and 3 for the five natural color images, color medical images, and color face images at five different levels of SRs. Table 4 presents the average running time of different algorithms when achieving their respective highest PSNR values (i.e., the PSNR values in Tables 1, 2 and 3). Figs. 4, 5 and 6 visually demonstrate the recovered results obtained by various methods for natural color images, color medical images, and color face images at SR $= 10\%$. Furthermore, we also validated the performance of these methods in recovering color images with structural missing. We conducted experiments on randomly selected one color image from each of the three categories, and the visual and quantitative results are shown in Figs. 7, 8 and 9. From these extensive experiments, it is evident that our proposed method exhibits significant advantages both visually and in terms of quantitative metrics when compared to both quaternion-based methods and TR-based

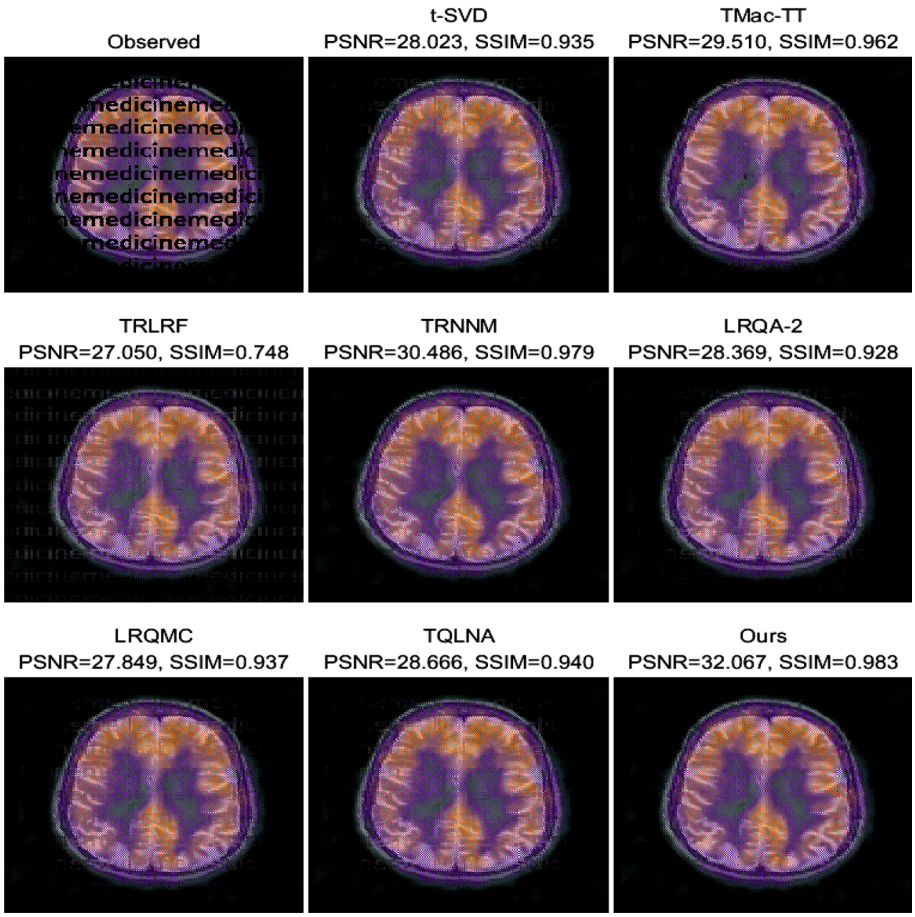

Fig. 8 Recovery of structurally missing color medical image: a comparison of various methods in terms of visual and quantitative metrics. **The figure is viewed better in zoomed PDF**

methods. This aligns with our expectation of complementary strengths between quaternions and TR. Additionally, we observed that these methods perform poorly when recovering color images with large areas of complete loss, as illustrated in Fig. 9. While our proposed method is relatively acceptable visually, the recovery of edges remains less than ideal. Hence, for inpainting tasks involving color images with large areas of complete loss, there is a requirement for specific model improvements in forthcoming research. Lastly, it should be noted that, as evident from Table 4, our method does not have an advantage in terms of running time. Therefore, our method may be better suited for handling offline color image inpainting tasks that do not have strict time constraints but require high recovery quality.

6 Conclusions

In this paper, we have defined the QTLR decomposition of quaternion tensors and introduced the relevant theory. The QTLR decomposition combines the advantages of both quater-

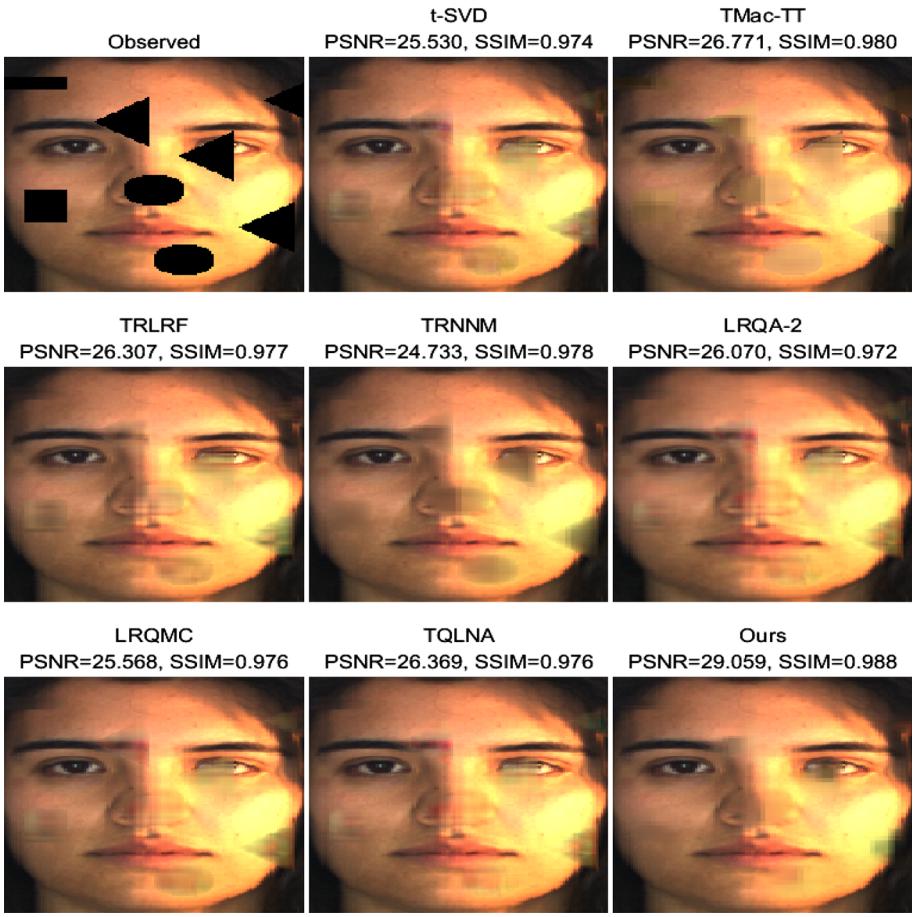

Fig. 9 Recovery of structurally missing color face image: a comparison of various methods in terms of visual and quantitative metrics. **The figure is viewed better in zoomed PDF**

nions and TR decomposition, providing a new theoretical foundation for the field of color image processing. This constitutes the core research focus of this paper. Furthermore, as an example of the application of the QTLR decomposition, we have proposed a method for color image inpainting. Specifically, we define the circular unfolding of quaternion tensors, establish a correlation between the rank of circular unfolding quaternion matrices and QTLR-rank, and leverage this connection to propose an LRQTC model for color image inpainting. The experiments provide evidence that the LRQTC method we introduce showcases exceptional performance across a spectrum of color image inpainting tasks. Regardless of whether compared to existing quaternion matrix-based methods or TR-based methods, our approach exhibits significant advantages both visually and in terms of quantitative metrics. This underscores the substantial potential inherent in the fusion of tensor TR with quaternions.

Nevertheless, owing to the non-commutative nature of quaternion multiplication, QTLR has yet to attain the same degree of theoretical robustness as TR. For instance, concerning the learning algorithm for the QTLR format, we have put forth solely a QTLR-QSVD approach. Ensuring the validity of Theorem 3, in which (25) holds, entails the imposition of constraints

such as $l = N - k + 1$, among others. These aspects underscore the need for additional refinement and enhancement in our forthcoming research endeavors. Furthermore, in order to better cater to time-sensitive color image inpainting tasks, it is possible to further develop online algorithms based on the proposed theory to reduce the running time.

Funding The work of the first author was supported by Yunnan Fundamental Research Projects (202401AU070203). The work of the second author was supported by the University of Macau (MYRG2022-00108-FST, MYRG-CRG2022-00010-ICMS), The Science and Technology Development Fund, Macau S.A.R (0036/2021/AGJ). The work of the third author was supported in part by the National Key Research and Development Program of China (2022YFE0112200), the National Natural Science Foundation of China (U21A20520, 62325204), Science and Technology Project of Guangdong Province (2022A0505050014), the Key-Area Research and Development Program of Guangzhou City (202206030009). The work of the fourth author was supported by the Science and Technology Planning Project of Guangzhou City, China (201907010043).

Data Availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Bengua, J.A., Phien, H.N., Tuan, H.D., Do, M.N.: Efficient tensor completion for color image and video recovery: low-rank tensor train. IEEE Trans. Image Process. **26**(5), 2466–2479 (2017)
2. Chen, J., Ng, M.K.: Color image inpainting via robust pure quaternion matrix completion: error bound and weighted loss. SIAM J. Imag. Sci. **15**(3), 1469–1498 (2022)
3. Chen, J.F., Wang, Q.W., Song, G.J., Li, T.: Quaternion matrix factorization for low-rank quaternion matrix completion. Mathematics **11**(9), 2144 (2023)
4. Chen, Y., Qi, L., Zhang, X., Xu, Y.: A low rank quaternion decomposition algorithm and its application in color image inpainting. arXiv preprint arXiv:2009.12203 (2020)
5. Chen, Y., Xiao, X., Zhou, Y.: Low-rank quaternion approximation for color image processing. IEEE Trans. Image Process. **29**, 1426–1439 (2019)
6. Chen, Z., Li, Y., Lu, J.: Tensor ring decomposition: optimization landscape and one-loop convergence of alternating least squares. SIAM J. Matrix Anal. Appl. **41**(3), 1416–1442 (2020)
7. Cichocki, A.: Era of big data processing: A new approach via tensor networks and tensor decompositions. arXiv preprint arXiv:1403.2048 (2014)
8. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**(4), 1253–1278 (2000)
9. Hamilton, W.R.: Elements of quaternions. Green, & Company, Longmans (1866)
10. He, Z.H., Wang, X.X., Zhao, Y.F.: Eigenvalues of quaternion tensors with applications to color video processing. J. Sci. Comput. **94**(1), 1 (2023)
11. Huang, H., Liu, Y., Liu, J., Zhu, C.: Provable tensor ring completion. Signal Process. **171**, 107486 (2020)
12. Huang, H., Liu, Y., Long, Z., Zhu, C.: Robust low-rank tensor ring completion. IEEE Trans. Comput. Imag. **6**, 1117–1126 (2020)
13. Jia, Z., Jin, Q., Ng, M.K., Zhao, X.L.: Non-local robust quaternion matrix completion for large-scale color image and video inpainting. IEEE Trans. Image Process. **31**, 3868–3883 (2022)
14. Jia, Z., Ng, M.K.: Structure preserving quaternion generalized minimal residual method. SIAM J. Matrix Anal. Appl. **42**(2), 616–634 (2021)
15. Jia, Z., Ng, M.K., Song, G.J.: Robust quaternion matrix completion with applications to image inpainting. Numer. Linear Algebra Appl. **26**(4), e2245 (2019)
16. Kilmer, M.E., Martin, C.D.: Factorization strategies for third-order tensors. Linear Algebra Appl. **435**(3), 641–658 (2011)
17. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
18. Li, H., Liu, Z., Huang, Y., Shi, Y.: Quaternion generic fourier descriptor for color object recognition. Pattern Recogn. **48**(12), 3895–3903 (2015)

19. Li, X.T., Zhao, X.L., Jiang, T.X., Zheng, Y.B., Ji, T.Y., Huang, T.Z.: Low-rank tensor completion via combined non-local self-similarity and low-rank regularization. Neurocomputing **367**, 1–12 (2019)
20. Liu, Q., Ling, S., Jia, Z.: Randomized quaternion singular value decomposition for low-rank matrix approximation. SIAM J. Sci. Comput. **44**(2), A870–A900 (2022)
21. Miao, J., Kou, K.I.: Quaternion-based bilinear factor matrix norm minimization for color image inpainting. IEEE Trans. Signal Process. **68**, 5617–5631 (2020)
22. Miao, J., Kou, K.I.: Color image recovery using low-rank quaternion matrix completion algorithm. IEEE Trans. Image Process. **31**, 190–201 (2021)
23. Miao, J., Kou, K.I.: Quaternion tensor singular value decomposition using a flexible transform-based approach. Signal Process. **206**, 108910 (2023)
24. Miao, J., Kou, K.I., Liu, W.: Low-rank quaternion tensor completion for recovering color videos and images. Pattern Recogn. **107**, 107505 (2020)
25. Miao, J., Kou, K.I., Yang, L., Cheng, D.: Quaternion tensor train rank minimization with sparse regularization in a transformed domain for quaternion tensor completion. Knowl.-Based Syst. **284**, 111222 (2024)
26. Miao, J., Kou, K.I., Yang, Y., Yang, L., Han, J.: Quaternion matrix completion using untrained quaternion convolutional neural network for color image inpainting. Signal Process. (2024). https://doi.org/10.1016/j.sigpro.2024.109504
27. Oseledets, I.V.: Tensor-train decomposition. SIAM J. Sci. Comput. **33**(5), 2295–2317 (2011)
28. Qin, Z., Ming, Z., Zhang, L.: Singular value decomposition of third order quaternion tensors. Appl. Math. Lett. **123**, 107597 (2022)
29. Qiu, Y., Zhou, G., Zhao, Q., Xie, S.: Noisy tensor completion via low-rank tensor ring. IEEE Trans. Neural Netw. Learn. Syst. **35**(1), 1127–1141 (2022)
30. Schulz, D., Seitz, J., da Costa, J.P.C.L.: Widely linear simo filtering for hypercomplex numbers. In: 2011 IEEE Information Theory Workshop, pp. 390–395. IEEE (2011)
31. Schulz, D., Thomä, R.S.: Using quaternion-valued linear algebra. arXiv preprint arXiv:1311.7488 (2013)
32. Wang, W., Aggarwal, V., Aeron, S.: Efficient low rank tensor ring completion. In: Proceedings of the IEEE international conference on computer vision, pp. 5697–5705 (2017)
33. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
34. Wu, P.L., Zhao, X.L., Ding, M., Zheng, Y.B., Cui, L.B., Huang, T.Z.: Tensor ring decomposition-based model with interpretable gradient factors regularization for tensor completion. Knowl.-Based Syst. **259**, 110094 (2023)
35. Yang, L., Miao, J., Kou, K.I.: Quaternion-based color image completion via logarithmic approximation. Inf. Sci. **588**, 82–105 (2022)
36. Yu, J., Li, C., Zhao, Q., Zhao, G.: Tensor-ring nuclear norm minimization and application for visual: Data completion. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 3142–3146. IEEE (2019)
37. Yu, Y., Zhang, Y., Yuan, S.: Quaternion-based weighted nuclear norm minimization for color image denoising. Neurocomputing **332**, 283–297 (2019)
38. Yuan, L., Li, C., Mandic, D., Cao, J., Zhao, Q.: 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, pp. 9151–9158 (2019)
39. Zhang, F.: Quaternions and matrices of quaternions. Linear Algebra Appl. **251**, 21–57 (1997)
40. Zhang, Y., Wang, Y., Han, Z., Tang, Y., et al.: Effective tensor completion via element-wise weighted low-rank tensor train with overlapping ket augmentation. IEEE Trans. Circuits Syst. Video Technol. **32**(11), 7286–7300 (2022)
41. Zhang, Z., Aeron, S.: Exact tensor completion using t-svd. IEEE Trans. Signal Process. **65**(6), 1511–1526 (2016)
42. Zhao, Q., Zhou, G., Xie, S., Zhang, L., Cichocki, A.: Tensor ring decomposition. arXiv preprint arXiv:1606.05535 (2016)
43. Zheng, J., Qin, M., Xu, H., Feng, Y., Chen, P., Chen, S.: Tensor completion using patch-wise high order hankelization and randomized tensor ring initialization. Eng. Appl. Artif. Intell. **106**, 104472 (2021)
44. Zheng, Y.B., Huang, T.Z., Zhao, X.L., Zhao, Q., Jiang, T.X.: Fully-connected tensor network decomposition and its application to higher-order tensor completion. In: Proceedings of the AAAI conference on artificial intelligence, vol. 35, pp. 11071–11078 (2021)