Journal of Scientific Computing (2019) 81:941–964
https://doi.org/10.1007/s10915-019-01044-8

Low-Rank Tensor Completion Using Matrix Factorization Based on Tensor Train Rank and Total Variation

Meng Ding[1] · Ting-Zhu Huang[1] · Teng-Yu Ji[2] · Xi-Le Zhao[1] · Jing-Hua Yang[1]

Received: 23 August 2018 / Revised: 10 February 2019 / Accepted: 23 August 2019 /
Published online: 30 August 2019

Abstract

Recently, the method called tensor completion by parallel matrix factorization via tensor train (TMac-TT) has achieved promising performance on estimating the missing information. TMac-TT, which borrows *ket augmentation* to transform a lower-order tensor into a higher-order tensor, suffers from serious block-artifacts. To tackle this issue, we build an optimization model combining low-rank matrix factorization based on tensor train (TT) rank and the total variation to retain the strength of TT rank and alleviate block-artifacts. We develop a block successive upper-bound minimization algorithm to solve the proposed model. Under some mild conditions, we theoretically prove that the proposed algorithm converges to the coordinatewise minimizers. Extensive numerical experiments illustrate the superiority of the proposed method over several existing state-of-the-art methods qualitatively and quantitatively.

Keywords Low-rank tensor completion · Tensor train rank · Total variation · Block successive upper-bound minimization

✉ Ting-Zhu Huang
tingzhuhuang@126.com

✉ Xi-Le Zhao
xlzhao122003@163.com

Meng Ding
dingmeng56@163.com

Teng-Yu Ji
tengyu_j66@126.com

Jing-Hua Yang
yangjinghua110@126.com

[1] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, People’s Republic of China

[2] Department of Applied Mathematics, School of Science, Northwestern Polytechnical University, Xi’an 710072, Shaanxi, People’s Republic of China

1 Introduction

Tensors, treated as high-dimensional generalizations of matrices and vectors, are powerful to express more complex interactions related to higher-order data. Recent years, the tensor analysis plays an important role in a wide range of real-world applications [3,10–12,14, 34]. Among them, one important problem is the tensor completion problem, which aims to estimate the missing values from the observed tensor data, e.g., color image or video inpainting [2,4,8,19], hyperspectral images recovery [5,15,21], magnetic resonance image (MRI) recovery [6,32], and higher-order web link analysis [18].

The low-rank constraint has become a powerful tool to recover the higher-order tensor with missing entries. Mathematically, the low-rank tensor completion (LRTC) problem can be formulated as

$$\begin{aligned} \min_{\mathcal{M}} \quad & \operatorname{rank}(\mathcal{M}) \\ s.t. \quad & \mathcal{P}_{\Omega}(\mathcal{M}) = \mathcal{T}, \end{aligned} \tag{1}$$

where $\mathcal{M} \in \mathbb{R}^{n_1 \times \cdots \times n_j}$ is the underlying tensor, $\mathcal{T} \in \mathbb{R}^{n_1 \times \cdots \times n_j}$ is the observed tensor, Ω is the index of observed entries, and $\mathcal{P}_{\Omega}(\cdot)$ is the projection operator that keeps entries in Ω and zeros out others. However, the definition for the rank of tensors is not unique, such as CANDECOMP/PARAFAC rank and Tucker rank [17]. Both of the corresponding minimization problems are generally NP-hard [9,26]. To tackle this problem, Liu et al. [22] firstly defined the nuclear norm of a tensor based on the mode-k unfolding [17]

$$\begin{aligned} \min_{\mathcal{M}} \quad & \|\mathcal{M}\|_* = \sum_{k=1}^{j} \alpha_k \|\mathbf{M}_{(k)}\|_* \\ s.t. \quad & \mathcal{P}_{\Omega}(\mathcal{M}) = \mathcal{T}, \end{aligned} \tag{2}$$

where $\mathbf{M}_{(k)} \in \mathbb{R}^{n_k \times \prod_{d \neq k} n_d}$ is the mode-k unfolding of $\mathcal{M}$ and $\alpha_k s$ are positive constants satisfying $\sum_{k=1}^{j} \alpha_k = 1$. The nuclear norm minimization methods for solving problem (2) have to calculate the singular value decomposition (SVD) for all matrices in every iteration, which suffer from high computational cost. To improve the capacity of solving large-scale problems, Xu et al. [35] performed the low-rank matrix factorization to all-mode unfolded matrices, i.e., factorize each mode matricization $\mathbf{M}_{(k)}$ into the product of two smaller matrices $\mathbf{X}_k$ and $\mathbf{Y}_k$, and proposed the following model:

$$\begin{aligned} \min_{\mathbf{X}_k, \mathbf{Y}_k, \mathcal{M}} \quad & \sum_{k=1}^{j} \alpha_k \|\mathbf{X}_k \mathbf{Y}_k - \mathbf{M}_{(k)}\|_F^2 \\ s.t. \quad & \mathcal{P}_{\Omega}(\mathcal{M}) = \mathcal{T}, \end{aligned} \tag{3}$$

where $\mathbf{X}_k \in \mathbb{R}^{n_k \times r_k}$, $\mathbf{Y}_k \in \mathbb{R}^{r_k \times \prod_{d \neq k} n_d}$, and r_k is the estimated rank of the matrix $\mathbf{M}_{(k)}$.

As the matrix $\mathbf{M}_{(k)}$ is constructed based on an unbalanced matricization scheme (one mode versus the rest) [7,22], the $rank(\mathbf{M}_{(k)})$ only captures the correlation between a simple mode (not a few modes) and rest modes of the tensor. Therefore, the existing methods based on Tucker rank maybe not suitable for completing higher-order tensors ($j > 3$) [16,28].

Recently, the tensor train (TT) rank, which considers the global correlation of tensors thanks to a well-balanced matricization scheme, is proposed [27,29] as

$$\operatorname{rank}_{tt}(\mathcal{M}) = \big(rank(\mathbf{M}_{[1]}), \ldots, rank(\mathbf{M}_{[j-1]})\big), \tag{4}$$

where $\mathbf{M}_{[k]} \in \mathbb{R}^{(\prod_{d=1}^{k} n_d)\times(\prod_{d=k+1}^{j} n_d)}$ is the mode-k canonical matricization of the tensor $\mathcal{M}$. Bengua et al. [1] applied TT rank to the color image and video completion. They proposed two methods based on TT rank for LRTC. The first one, simple low-rank tensor completion via tensor train (SiLRTC-TT), minimizes the TT nuclear norm, i.e.,

$$\begin{aligned} \min_{\mathcal{M}} \quad & \sum_{k=1}^{j-1} \alpha_k \|\mathbf{M}_{[k]}\|_* \\ s.t. \quad & \mathcal{P}_\Omega(\mathcal{M}) = \mathcal{T}, \end{aligned} \tag{5}$$

where α_k is the constant satisfying $\alpha_k \geq 0$ and $\sum_{k=1}^{j-1} \alpha_k = 1$. Another one, tensor completion by parallel matrix factorization via tensor train (TMac-TT), uses matrix factorization to approximate the TT rank of a tensor, i.e.,

$$\begin{aligned} \min_{\mathbf{X}_k,\mathbf{Y}_k,\mathcal{M}} \quad & \sum_{k=1}^{j-1} \frac{\alpha_k}{2} \|\mathbf{X}_k\mathbf{Y}_k - \mathbf{M}_{[k]}\|_F^2 \\ s.t. \quad & \mathcal{P}_\Omega(\mathcal{M}) = \mathcal{T}, \end{aligned} \tag{6}$$

where $\mathbf{X}_k \in \mathbb{R}^{(\prod_{d=1}^{k} n_d)\times r_k}$, $\mathbf{Y}_k \in \mathbb{R}^{r_k\times(\prod_{d=k+1}^{j} n_d)}$, and r_k is the rank of the matrix $\mathbf{M}_{[k]}$. TT rank is more suitable for higher-order tensors due to the ability of capturing the global correlation of a tensor. In order to handle the third-order tensor data, SiLRTC-TT and TMac-TT use a tensor augmented scheme known as *ket augmentation* (KA), but they cause block-artifacts [24] on restored images.

KA uses an appropriate block structured addressing scheme to cast a lower-order tensor into a higher-order tensor. However, this scheme does not consider the local smoothness between blocks and blocks, so the block-artifacts are caused on restored images. The block-artifacts can be seen in the results for completion of the Lena image with 90% missing entries in Fig. 1b, c. To reduce the block-artifacts, we need to smooth the edges between each block. Total variation (TV), one of the most famous functions to characterize the piecewise smoothness prior, has been shown to preserve edges well in image processing [13,25,31,33,36,37]. Motivated by former works, we introduce TV into LRTC (6),

$$\begin{aligned} \min_{\mathbf{X}_k,\mathbf{Y}_k,\mathcal{M}} \quad & \sum_{k=1}^{j-1} \frac{\alpha_k}{2} \|\mathbf{X}_k\mathbf{Y}_k - \mathbf{M}_{[k]}\|_F^2 + \lambda \mathrm{TV}(\mathcal{M}) \\ s.t. \quad & \mathcal{P}_\Omega(\mathcal{M}) = \mathcal{T}, \end{aligned} \tag{7}$$

where α_k satisfies $\alpha_k \geq 0$ and $\sum_{k=1}^{j-1} \alpha_k = 1$, λ is a regularization parameter, and $\mathrm{TV}(\mathcal{M})$ is the total variation of $\mathcal{M}$ in spatial dimensions (see details in Sect. 3.1). The proposed model is named tensor completion via matrix factorization based on tensor train rank and total variation (MF-TTTV). From Fig. 1, it is clear that our method can effectively alleviate block-artifacts compared with SiLRTC-TT and TMac-TT.

The contributions of this paper are mainly two folds: (1) we propose a new tensor completion model combining low-rank matrix factorization based on TT rank with the TV regularization, which simultaneously exploit the low-rankness and the piecewise smoothness prior of the underlying tensor; (2) we develop a block successive upper-bound minimization (BSUM) algorithm to solve the proposed model. Experiments demonstrate that our method performs better than the compared methods and effectively reduces the block-artifacts caused by using KA.

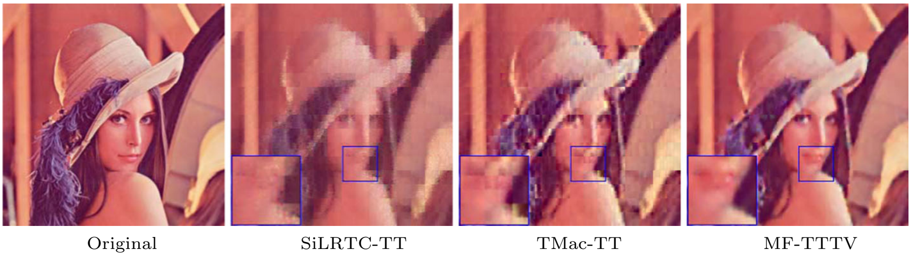


Fig. 1 The results of *Lena* with 90% missing entries by different methods. From left to right: the original image, the result by SiLRTC-TT, TMac-TT, and MF-TTTV, respectively

The outline of this paper is as follows. Section 2 reviews some preliminary knowledge about the tensor, the tensor train rank, the proximal operator, and KA. Section 3 describes the model formulation and an efficient BSUM-based algorithm with convergence analysis. Section 4 demonstrates the effectiveness of the proposed method based on abundant numerical experiments. Section 5 summarizes this paper.

2 Preliminary

2.1 Tensor Basics

A tensor is a high-dimensional array and its order (or mode) is the number of its dimensions. We denote scalars as lowercase letters, e.g., z, vectors as boldface lowercase letters, e.g., $\mathbf{z}$, matrices as capital letters, e.g., $\mathbf{Z}$, and tensors as calligraphic letters, e.g., $\mathcal{Z}$. A jth-order tensor is defined as $\mathcal{Z} \in \mathbb{R}^{n_1 \times \cdots \times n_j}$ whose $(i_1, \ldots, i_j)$-th component is denoted as $z_{i_1,\ldots,i_j}$.

The inner product of two tensors $\mathcal{X}$ and $\mathcal{Y}$ with same size is defined as

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1,\ldots,i_j} x_{i_1,\ldots,i_j} \cdot y_{i_1,\ldots,i_j}.$$

The Frobenius norm of a jth-order tensor $\mathcal{Z}$ is $\|\mathcal{Z}\|_F = \sqrt{\langle \mathcal{Z}, \mathcal{Z} \rangle}$.

The mode-k unfolding of a tensor $\mathcal{Z}$ is denoted as $\mathbf{Z}_{(k)} \in \mathbb{R}^{n_k \times \prod_{d \neq k} n_d}$, where the tensor element $(i_1, \ldots, i_j)$ maps to the element (i_k, b) of matrix $\mathbf{Z}_{(k)}$ satisfying

$$b = 1 + \sum_{d=1, d \neq k}^{j} (i_d - 1) j_d \quad \text{with} \quad j_d = \prod_{t=1, t \neq k}^{d-1} n_t. \tag{8}$$

We denote the mode-k unfolding of a tensor $\mathcal{Z}$ as $\mathbf{Z}_{(k)} = \text{unfold}_{(k)}(\mathcal{Z})$. The inverse operator of unfolding is denoted as "fold", i.e., $\mathcal{Z} = \text{fold}_{(k)}(\mathbf{Z}_{(k)})$.

The mode-k canonical matricization of a tensor $\mathcal{Z}$ is defined as $\mathbf{Z}_{[k]} \in \mathbb{R}^{(\prod_{d=1}^{k} n_d) \times (\prod_{d=k+1}^{j} n_d)}$, where the tensor element $(i_1, \ldots, i_j)$ maps to the element (a, b) of matrix $\mathbf{Z}_{[k]}$ satisfying

$$a = 1 + \sum_{d=1}^{k} \left((i_d - 1) \prod_{t=1}^{d-1} n_t \right) \quad \text{and} \quad b = 1 + \sum_{d=k+1}^{j} \left((i_d - 1) \prod_{t=k+1}^{d-1} n_t \right). \tag{9}$$

Table 1 Tensor notations

Notations	Explanations
$\mathcal{Z}$, $\mathbf{Z}$, $\mathbf{z}$, z	Tensor, matrix, vector, scalar
$\langle \mathcal{X}, \mathcal{Y} \rangle$	Inner product of two same-sized tensors $\mathcal{X}$ and $\mathcal{Y}$
$\\|\mathcal{Z}\\|_F$	Frobenius norm of tensor $\mathcal{Z}$
$\mathbf{Z}_{(k)}$, $\text{unfold}_{(k)}(\mathcal{Z})$	Mode-k unfolding of a tensor $\mathcal{Z} \in \mathbb{R}^{n_1 \times \cdots \times n_j}$ denoted as $\mathbf{Z}_{(k)} \in \mathbb{R}^{n_k \times \prod_{d \neq k} n_d}$
$\text{fold}_{(k)}(\mathbf{Z}_{(k)})$	The inverse operator of unfolding denoted as $\mathcal{Z} = \text{fold}_{(k)}(\mathbf{Z}_{(k)})$
$\mathbf{Z}_{[k]}$, $\text{reshape}_{[k]}(\mathcal{Z})$	Mode-k canonical matricization of a tensor $\mathcal{Z} \in \mathbb{R}^{n_1 \times \cdots \times n_j}$ defined as $\mathbf{Z}_{[k]} \in \mathbb{R}^{(\prod_{d=1}^{k} n_d) \times (\prod_{d=k+1}^{j} n_d)}$
$\text{unreshape}_{[k]}(\mathbf{Z}_{[k]})$	The inverse operator of reshape denoted as $\mathcal{Z} = \text{unreshape}_{[k]}(\mathbf{Z}_{[k]})$
$\big(rank(\mathbf{Z}_{[1]}), \ldots, rank(\mathbf{Z}_{[j-1]})\big)$	Tensor train rank

In MATLAB, it can be implemented by the reshape function

$$\mathbf{Z}_{[k]} = \text{reshape}_{[k]}\left(\mathcal{Z}, \Pi_{d=1}^{k} n_d, \Pi_{d=k+1}^{j} n_d\right).$$

We denote the mode-k canonical matricization of a tensor $\mathcal{Z}$ as $\mathbf{Z}_{[k]} = \text{reshape}_{[k]}(\mathcal{Z})$. The inverse operator of reshape is denoted as “unreshape”, i.e., $\mathcal{Z} = \text{unreshape}_{[k]}(\mathbf{Z}_{[k]})$.

The TT rank is defined as the vector

$$\text{rank}_{tt}(\mathcal{Z}) = \big(rank(\mathbf{Z}_{[1]}), \ldots, rank(\mathbf{Z}_{[j-1]})\big).$$

The tensor $\mathcal{Z}$ is low-rank, if $\mathbf{Z}_{[k]}$ is low-rank for all k .

The notations are listed in Table 1.

2.2 Operators

Let Ω be an index set, then the projection operator $\mathcal{P}_\Omega(\mathcal{Z})$ denotes the tensor copying the entries from $\mathcal{Z}$ in the set Ω and letting the remaining entries be zeros, i.e.,

$$(\mathcal{P}_\Omega(\mathcal{Z}))_{i_1,\ldots,i_j} = \begin{cases} z_{i_1,\ldots,i_j}, & (i_1, \ldots, i_j) \in \Omega, \\ 0, & \text{otherwise.} \end{cases}$$

The proximal operator of a given convex function $f(x)$ is defined as

$$\text{prox}_f(y) = \arg\min_x f(x) + \frac{\rho}{2}\|x - y\|^2, \tag{10}$$

where ρ is a positive proximal constant. The problem $\min_x\{f(x)\}$ is equivalent to $\min_{x,y}\{f(x) + \frac{\rho}{2}\|x - y\|^2\}$. Thus one can obtain the minimization of $f(x)$ by iteratively solving $\text{prox}_f(x^l)$, where x^l is the latest update of x. The proximal operator is very attractive in that the objective function (10) is strongly convex with respect to x so long as $f(x)$ is convex.

2.3 Ket Augmentation

In this subsection, we introduce ket augmentation (KA) to represent a lower-order tensor by a higher-order one without changing the number of entries in the tensor. The TT decomposition can effectively exploit the local structure of the data by considering the low-rankness of the augmented tensor. Even though the tensor is slightly correlated, its augmented tensor has low TT rank [20].

KA is firstly introduced in [20] to cast a grayscale image into *real ket state* of a Hilbert space by using an appropriate block structured addressing. In [1], the authors successfully extended KA to third-order tensors. Next, we introduce the details about how KA works on the color image. We define the third-order tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ that represents the color image, where $n_1 \times n_2 = 2^n \times 2^n$ is the number of pixels in the image and $n_3 = 3$ is the number of colors. Let us start with an initial block, labeled as i_1 and containing 2×2 pixels corresponding to a single color j ($j = 1, 2, 3$ denotes red, green and blue colors, respectively). This block can be represented as

$$\tilde{\mathcal{M}} = \sum_{i_1=1}^{4} c_{i_1 j} \mathbf{e}_{i_1},$$

where $c_{i_1 j}$ is the pixel value corresponding to color j and $\mathbf{e}_{i_1}$ is the orthonormal base $\mathbf{e}_1 = (1, 0, 0, 0)$, $\mathbf{e}_2 = (0, 1, 0, 0)$, $\mathbf{e}_3 = (0, 0, 1, 0)$, and $\mathbf{e}_4 = (0, 0, 0, 1)$. For all three color channels, the three blocks is represented as

$$\tilde{\mathcal{M}} = \sum_{i_1=1}^{4} \sum_{j=1}^{3} c_{i_1 j} \mathbf{e}_{i_1} \otimes \mathbf{u}_j,$$

where $\mathbf{u}_j$ is the orthonormal base $\mathbf{u}_1 = (1, 0, 0)$, $\mathbf{u}_2 = (0, 1, 0)$, $\mathbf{u}_3 = (0, 0, 1)$, and $\otimes$ denotes a tensor product [17]. Let us consider a larger block (labeled as i_2) which consists of four inner sub-blocks for each color j as shown in Fig. 2b. So we have the new block

$$\tilde{\mathcal{Z}} = \sum_{i_2=1}^{4} \sum_{i_1=1}^{4} \sum_{j=1}^{3} c_{i_2 i_1 j} \mathbf{e}_{i_2} \otimes \mathbf{e}_{i_1} \otimes \mathbf{u}_j.$$

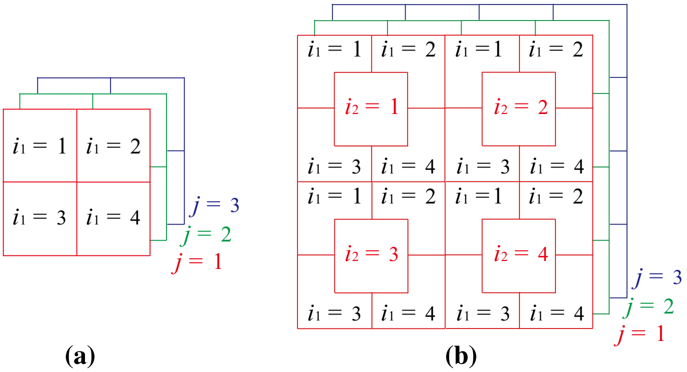


Fig. 2 Graphical illustration of a structured block addressing procedure. **a** Example addressing for a $2 \times 2 \times 3$ color image. **b** Example addressing for a $2^2 \times 2^2 \times 3$ color image (Color figure online)

The color image can be cast into an $(n+1)$th-order tensor as follows:

$$\tilde{\mathcal{M}} = \sum_{i_1,\ldots,i_n=1}^{4} \sum_{j=1}^{3} c_{i_n,\ldots,i_1 j} \mathbf{e}_{i_n} \otimes \cdots \otimes \mathbf{e}_{i_1} \otimes \mathbf{u}_j. \tag{11}$$

Based on the KA method, TT-based methods can effectively exploit the local structure information of visual data and have a better low-rank representation. After the augmented data is processed by the completion method, the reverse operation of KA is performed to obtain the original image form.

3 The Proposed Model and Algorithm

3.1 The Proposed Model

The objective function of our model is:

$$f(\mathbf{X}, \mathbf{Y}, \mathcal{M}) = \min_{\mathcal{M},\mathbf{X},\mathbf{Y}} \sum_{k=1}^{j-1} \frac{\alpha_k}{2} \|\mathbf{X}_k \mathbf{Y}_k - \mathbf{M}_{[k]}\|_F^2 + \lambda \mathrm{TV}(\mathcal{M}), \tag{12}$$

where $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_{j-1})$, $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_{j-1})$, and λ is the regularization parameter. In the proposed model, the first term is the fidelity term, the second term is the regularization term.

In the fidelity term, matrices $\mathbf{M}_{[k]}$, obtained by a well-balanced matricization scheme, capture the global information of the underlying tensor $\mathcal{M}$. This fidelity term enhances the low-rankness of the underlying tensor $\mathcal{M}$.

In the regularization term, $\mathrm{TV}(\mathcal{M})$ denotes the isotropic TV in spatial dimensions. This term aims at enhancing the piecewise smoothness in spatial dimensions. According to the rules of unfolding a tensor (8), we find that the mode-1 unfolding matrix $\mathbf{M}_{(1)}$ has a good structure:

$$\mathbf{M}_{(1)} = [\mathbf{M}^{(1)}, \ldots, \mathbf{M}^{(\hat{s})}] \in \mathbb{R}^{n_1 \times s}, \tag{13}$$

where $\hat{s} = \prod_{k=3}^{j} n_k$, $s = n_2 \times \hat{s}$, and $\mathbf{M}^{(i)} \in \mathbb{R}^{n_1 \times n_2}$ is a matrix in spatial dimensions of the tensor $\mathcal{M}$.

In our works, the isotropic TV is defined as follows:

$$\mathrm{TV}(\mathcal{M}) = \sum_{i=1}^{\hat{s}} \sum_{m=1}^{n_1} \sum_{n=1}^{n_2} \sqrt{[D_{m,n}^1 \mathbf{M}^{(i)}]^2 + [D_{m,n}^2 \mathbf{M}^{(i)}]^2}, \tag{14}$$

where $D_{m,n}^1 \mathbf{M}^{(i)}$ and $D_{m,n}^2 \mathbf{M}^{(i)}$ are the horizontal and vertical gradient values at the (m, n)-th pixel of matrix $\mathbf{M}^{(i)}$, respectively, and $D_{m,n}^1$ and $D_{m,n}^2$ are the discrete horizontal and vertical gradient operators, respectively.

3.2 The Proposed Algorithm

It is clear that the objective function (12) is not jointly convex for $(\mathbf{X}, \mathbf{Y}, \mathcal{M})$, but is convex with respect to $\mathbf{X}$, $\mathbf{Y}$, and $\mathcal{M}$ independently. In order to solve the nonconvex problem effectively, we develop the BSUM-based algorithm.

Let $\mathcal{Z} = (\mathbf{X}, \mathbf{Y}, \mathcal{M})$ and $\mathcal{Z}^l = (\mathbf{X}^l, \mathbf{Y}^l, \mathcal{M}^l)$, by utilizing the proximal operator, (7) can be updated through the following:

$$\mathcal{Z}^{l+1} = \arg\min_{\mathcal{Z}} h(\mathcal{Z}, \mathcal{Z}^l) = \arg\min_{\mathcal{Z}} f(\mathcal{Z}) + \frac{\rho}{2}\|\mathcal{Z} - \mathcal{Z}^l\|_F^2, \tag{15}$$

where ρ is the positive proximal parameter. Note that (15) can be written as

$$\begin{cases} \mathbf{X}^{l+1} = \arg\min_{\mathbf{X}} h_1(\mathbf{X}, \mathcal{Z}_1^l) = \arg\min_{\mathbf{X}} f(\mathbf{X}, \mathbf{Y}^l, \mathcal{M}^l) + \frac{\rho}{2}\|\mathbf{X} - \mathbf{X}^l\|_F^2, \\ \mathbf{Y}^{l+1} = \arg\min_{\mathbf{Y}} h_2(\mathbf{Y}, \mathcal{Z}_2^l) = \arg\min_{\mathbf{Y}} f(\mathbf{X}^{l+1}, \mathbf{Y}, \mathcal{M}^l) + \frac{\rho}{2}\|\mathbf{Y} - \mathbf{Y}^l\|_F^2, \\ \mathcal{M}^{l+1} = \arg\min_{\mathcal{P}_\Omega(\mathcal{M})=\mathcal{T}} h_3(\mathcal{M}, \mathcal{Z}_3^l) = \arg\min_{\mathcal{P}_\Omega(\mathcal{M})=\mathcal{T}} f(\mathbf{X}^{l+1}, \mathbf{Y}^{l+1}, \mathcal{M}) + \frac{\rho}{2}\|\mathcal{M} - \mathcal{M}^l\|_F^2, \end{cases} \tag{16}$$

where $\mathcal{Z}_1^l = (\mathbf{X}^l, \mathbf{Y}^l, \mathcal{M}^l)$, $\mathcal{Z}_2^l = (\mathbf{X}^{l+1}, \mathbf{Y}^l, \mathcal{M}^l)$, $\mathcal{Z}_3^l = (\mathbf{X}^{l+1}, \mathbf{Y}^{l+1}, \mathcal{M}^l)$. It is easy to see that the $\mathbf{X}$- and $\mathbf{Y}$-subproblem can be decomposed into $j-1$ independent problems. Then, the (16) can be written as

$$\begin{cases} \mathbf{X}_k^{l+1} = \arg\min_{\mathbf{X}_k} \frac{\alpha_k}{2}\|\mathbf{X}_k\mathbf{Y}_k^l - \mathbf{M}_{[k]}^l\|_F^2 + \frac{\rho}{2}\|\mathbf{X}_k - \mathbf{X}_k^l\|_F^2 \\ \qquad = (\alpha_k \mathbf{M}_{[k]}^l (\mathbf{Y}_k^l)^T + \rho\mathbf{X}_k^l)(\alpha_k \mathbf{Y}_k^l (\mathbf{Y}_k^l)^T + \rho\mathbf{I})^{-1}, \\ \mathbf{Y}_k^{l+1} = \arg\min_{\mathbf{Y}_k} \frac{\alpha_k}{2}\|\mathbf{X}_k^{l+1}\mathbf{Y}_k - \mathbf{M}_{[k]}^l\|_F^2 + \frac{\rho}{2}\|\mathbf{Y}_k - \mathbf{Y}_k^l\|_F^2 \\ \qquad = (\alpha_k (\mathbf{X}_k^{l+1})^T\mathbf{X}_k^{l+1} + \rho\mathbf{I})^{-1}(\alpha_k(\mathbf{X}_k^{l+1})^T\mathbf{M}_{[k]}^l + \rho\mathbf{Y}_k^l), \\ \mathcal{M}^{l+1} = \arg\min_{\mathcal{P}_\Omega(\mathcal{M})=\mathcal{T}} \sum_{k=1}^{j-1} \frac{\alpha_k}{2}\|\mathbf{X}_k^{l+1}\mathbf{Y}_k^{l+1} - \mathbf{M}_{[k]}\|_F^2 + \lambda\mathrm{TV}(\mathcal{M}) + \frac{\rho}{2}\|\mathcal{M} - \mathcal{M}^l\|_F^2, \end{cases} \tag{17}$$

where $\mathbf{I} \in \mathbb{R}^{r_k \times r_k}$ is an identify matrix.

The cost of computing $\mathbf{X}_k$ is $O(r_k \prod_{d=1}^j n_d)$ and the cost of computing $\mathbf{Y}_k$ is $O(r_k \prod_{d=1}^j n_d)$.

Since the $\mathcal{M}$-subproblem does not admit a close-form solution, we solve the $\mathcal{M}$-subproblem using the alternating direction method of multipliers (ADMM). By introducing auxiliary variables, we rewrite it as the following equivalent constrained problem:

$$\begin{aligned} \min\ & \lambda \sum_{m=1}^{n_1}\sum_{n=1}^{s} \|\mathbf{E}_{m,n}\|_2 + \sum_{k=1}^{j-1} \frac{\alpha_k}{2}\|\mathbf{X}_k^{l+1}\mathbf{Y}_k^{l+1} - \mathcal{A}_{k[k]}\|_F^2 + \frac{\rho}{2}\|\mathcal{M} - \mathcal{M}^l\|_F^2, \\ s.t.\ & \mathcal{P}_\Omega(\mathcal{M}) = \mathcal{T},\ \mathcal{M} = \mathcal{A}_k,\ \mathcal{M} = \mathcal{Z},\ \mathbf{D}_1\mathbf{Z}_{(1)} = \mathbf{E}_1,\ \mathbf{D}_2\mathbf{Z}_{(1)} = \mathbf{E}_2, \end{aligned} \tag{18}$$

where $\mathbf{E}_{m,n} = [(\mathbf{E}_1)_{m,n}, (\mathbf{E}_2)_{m,n}] \in \mathbb{R}^{1\times 2}$, $(\mathbf{E}_1)_{m,n}$ and $(\mathbf{E}_2)_{m,n}$ are the (m,n)-th entries of $\mathbf{E}_1$ and $\mathbf{E}_2$, respectively, and $\mathbf{D}_1$ and $\mathbf{D}_2$ are the first-order difference matrices in the horizontal and vertical directions based on $D_{m,n}^1$ and $D_{m,n}^2$, respectively.

The augmented Lagrangian function of (18) is defined as

$$\begin{aligned} L(\mathcal{M}, \mathcal{A}_k, \mathcal{Z}, \mathbf{E}_i, \mathcal{B}_k, \mathcal{Q}_i, \mathbf{F}_i) = \lambda \sum_{m=1}^{n_1}\sum_{n=1}^{s} \|\mathbf{E}_{m,n}\|_2 &+ \sum_{k=1}^{j-1} \frac{\alpha_k}{2} \|\mathbf{X}_k^{l+1}\mathbf{Y}_k^{l+1} - \mathcal{A}_{k[k]}\|_F^2 \\ &+ \frac{\rho}{2}\|\mathcal{M} - \mathcal{M}^l\|_F^2 \\ &+ \sum_{k=1}^{j-1}\left(\frac{\beta_1}{2}\|\mathcal{M} - \mathcal{A}_k\|_F^2 + \langle \mathcal{M} - \mathcal{A}_k, \mathcal{B}_k\rangle\right) \\ &+ \frac{\beta_2}{2}\|\mathcal{M} - \mathcal{Z}\|_F^2 + \langle \mathcal{M} - \mathcal{Z}, \mathcal{Q}\rangle \\ &+ \frac{\beta_3}{2}\|\mathbf{D}_1\mathbf{Z}_{(1)} - \mathbf{E}_1\|_F^2 + \langle \mathbf{D}_1\mathbf{Z}_{(1)} - \mathbf{E}_1, \mathbf{F}_1\rangle \\ &+ \frac{\beta_3}{2}\|\mathbf{D}_2\mathbf{Z}_{(1)} - \mathbf{E}_2\|_F^2 + \langle \mathbf{D}_2\mathbf{Z}_{(1)} - \mathbf{E}_2, \mathbf{F}_2\rangle, \end{aligned} \tag{19}$$

where $\mathcal{Q}$, $\{\mathcal{B}_k\}_{k=1}^{j-1}$, and $\{\mathbf{F}_i\}_{i=1}^{2}$ are Lagrangian multipliers of the linear constraint and $\beta_1, \beta_2, \beta_3 > 0$ are penalty parameters. The iterative scheme for solving (18) is as follows:

$$\begin{cases} \mathcal{M}^{l+1,p+1} = \underset{\mathcal{P}_\Omega(\mathcal{M})=\mathcal{T}}{\arg\min}\ L(\mathcal{M}, \mathcal{A}_k^p, \mathcal{Z}^p, \mathcal{B}_k^p, \mathcal{Q}_i^p), \\ \mathcal{A}_k^{p+1} = \underset{\mathcal{A}_k}{\arg\min}\, L(\mathcal{M}^{l+1,p+1}, \mathcal{A}_k, \mathcal{B}_k^{p+1}), \\ \mathcal{Z}^{p+1} = \underset{\mathcal{Z}}{\arg\min}\, L(\mathcal{M}^{l+1,p+1}, \mathcal{Z}, \mathbf{E}_i^p, \mathcal{Q}^p, \mathbf{F}_i^p), \\ \mathbf{E}_i^{p+1} = \underset{E_i}{\arg\min}\, L(\mathcal{Z}^{p+1}, \mathbf{E}_i, \mathbf{F}_i^p), \\ \mathcal{B}_k^{p+1} = \mathcal{B}_k^p + \beta_1(\mathcal{M}^{l+1,p+1} - \mathcal{A}_k^{p+1}), \\ \mathcal{Q}^{p+1} = \mathcal{Q}^p + \beta_2(\mathcal{M}^{l+1,p+1} - \mathcal{Z}^{p+1}), \\ \mathbf{F}_i^{p+1} = \mathbf{F}_i^p + \beta_3(\mathbf{D}_i\mathbf{Z}_{(1)}^{p+1} - \mathbf{E}_i^{p+1}). \end{cases} \tag{20}$$

We give the details for solving the first four subproblems in (20).

1. **$\mathcal{M}$-subproblem** We fix all variables but $\mathcal{M}$, the optimal $\mathcal{M}$ is obtained as

$$\begin{aligned} \mathcal{M}^{l+1,p+1} &= \underset{\mathcal{P}_\Omega(\mathcal{M})=\mathcal{T}}{\arg\min} \sum_{k=1}^{j-1}\left(\frac{\beta_1}{2}\|\mathcal{M} - \mathcal{A}_k^p\|_F^2 + \langle \mathcal{M} - \mathcal{A}_k^p, \mathcal{B}_k^p\rangle\right) \\ &\quad + \frac{\beta_2}{2}\|\mathcal{M} - \mathcal{Z}^p\|_F^2 + \langle \mathcal{M} - \mathcal{Z}^p, \mathcal{Q}^p\rangle + \frac{\rho}{2}\|\mathcal{M} - \mathcal{M}^l\|_F^2 \\ &= \underset{\mathcal{P}_\Omega(\mathcal{M})=\mathcal{T}}{\arg\min} \sum_{k=1}^{j-1} \frac{\beta_1}{2}\left\|\mathcal{M} - \mathcal{A}_k^p + \frac{\mathcal{B}_k^p}{\beta_1}\right\|_F^2 + \frac{\beta_2}{2}\left\|\mathcal{M} - \mathcal{Z}^p + \frac{\mathcal{Q}^p}{\beta_2}\right\|_F^2 \\ &\quad + \frac{\rho}{2}\|\mathcal{M} - \mathcal{M}^l\|_F^2. \end{aligned} \tag{21}$$

The function is quadratic in terms of $\mathcal{M}$. The optimal $\mathcal{M}$ is

$$\mathcal{M}^{l+1,p+1} = \mathcal{P}_{\Omega^c}\left(\frac{\sum_{k=1}^{j-1}(\beta_1\mathcal{A}_k^p - \mathcal{B}_k^p) + \beta_2\mathcal{Z}^p - \mathcal{Q}^p + \rho\mathcal{M}^l}{(j-1)\beta_1 + \beta_2 + \rho}\right) + \mathcal{T}. \tag{22}$$

2. $\{\mathcal{A}_k\}$-**subproblem** We give the optimal $\mathcal{A}_k$ by

$$\begin{aligned}\mathcal{A}_k^{p+1} &= \arg\min_{\mathcal{A}_k} \frac{\alpha_k}{2} \left\| \mathbf{X}_k^{l+1}\mathbf{Y}_k^{l+1} - \mathcal{A}_{k[k]} \right\|_F^2 \\ &\quad + \frac{\beta_1}{2} \|\mathcal{M}^{l+1,p+1} - \mathcal{A}_k\|_F^2 + \langle \mathcal{M}^{l+1,p+1} - \mathcal{A}_k, \mathcal{B}_k^p \rangle \\ &= \arg\min_{\mathcal{A}_k} \frac{\alpha_k}{2} \left\| \mathbf{X}_k^{l+1}\mathbf{Y}_k^{l+1} - \mathcal{A}_{k[k]} \right\|_F^2 + \frac{\beta_1}{2} \left\| \mathcal{M}^{l+1,p+1} - \mathcal{A}_k + \frac{\mathcal{B}_k^p}{\beta_1} \right\|_F^2 .\end{aligned} \tag{23}$$

Utilizing the equation $\|\mathbf{M}_{[k]}\|_F = \|\mathcal{M}\|_F$ and the definition of unreshape$_{[k]}(\cdot)$ in Sect. 2.1, we rewrite the optimization problem (23) as the following problem:

$$\mathcal{A}_k^{p+1} = \arg\min_{\mathcal{A}_k} \frac{\alpha_k}{2} \left\| \text{unreshape}_{[k]}(\mathbf{X}_k^{l+1}\mathbf{Y}_k^{l+1}) - \mathcal{A}_k \right\|_F^2 + \frac{\beta_1}{2} \left\| \mathcal{M}^{l+1,p+1} - \mathcal{A}_k + \frac{\mathcal{B}_k^p}{\beta_1} \right\|_F^2 . \tag{24}$$

The function is quadratic in terms of $\mathcal{A}_k$. The optimal $\mathcal{A}_k$ is

$$\mathcal{A}_k^{p+1} = \frac{\alpha_k \text{unreshape}_{[k]}(\mathbf{X}_k^{l+1}\mathbf{Y}_k^{l+1}) + \beta_1 \mathcal{M}^{l+1,p+1} + \mathcal{B}_k^p}{\alpha_k + \beta_1}. \tag{25}$$

3. $\mathcal{Z}$-**subproblem** The $\mathcal{Z}$-subproblem is

$$\begin{aligned}\mathcal{Z}^{p+1} &= \arg\min_{\mathcal{Z}} \frac{\beta_2}{2} \left\| \mathcal{M}^{l+1,p+1} - \mathcal{Z} \right\|_F^2 + \langle \mathcal{M}^{l+1,p+1} - \mathcal{Z}, \mathcal{Q}^p \rangle + \frac{\beta_3}{2} \left\| \mathbf{D}_1\mathbf{Z}_{(1)} - \mathbf{E}_1^p \right\|_F^2 \\ &\quad + \langle \mathbf{D}_1\mathbf{Z}_{(1)} - \mathbf{E}_1^p, \mathbf{F}_1^p \rangle + \frac{\beta_3}{2} \|\mathbf{D}_2\mathbf{Z}_{(1)} - \mathbf{E}_2^p\|_F^2 + \langle \mathbf{D}_2\mathbf{Z}_{(1)} - \mathbf{E}_2^p, \mathbf{F}_2^p \rangle \\ &= \arg\min_{\mathcal{Z}} \frac{\beta_2}{2} \left\| \mathcal{M}^{l+1,p+1} - \mathcal{Z} + \frac{\mathcal{Q}^p}{\beta_2} \right\|_F^2 + \frac{\beta_3}{2} \left\| \mathbf{D}_1\mathbf{Z}_{(1)} - \mathbf{E}_1^p + \frac{\mathbf{F}_1^p}{\beta_3} \right\|_F^2 \\ &\quad + \frac{\beta_3}{2} \|\mathbf{D}_2\mathbf{Z}_{(1)} - \mathbf{E}_2^p + \frac{\mathbf{F}_2^p}{\beta_3}\|_F^2 .\end{aligned} \tag{26}$$

Using the equation $\|\mathbf{Z}_{(1)}\|_F = \|\mathcal{Z}\|_F$, we solve its equivalent problem as

$$\begin{aligned}\mathbf{Z}_{(1)}^{p+1} &= \arg\min_{\mathbf{Z}_{(1)}} \frac{\beta_2}{2} \left\| \mathbf{M}_{(1)}^{l+1,p+1} - \mathbf{Z}_{(1)} + \frac{\mathbf{Q}_{(1)}^p}{\beta_2} \right\|_F^2 + \frac{\beta_3}{2} \left\| \mathbf{D}_1\mathbf{Z}_{(1)} - \mathbf{E}_1^p + \frac{\mathbf{F}_1^p}{\beta_3} \right\|_F^2 \\ &\quad + \frac{\beta_3}{2} \left\| \mathbf{D}_2\mathbf{Z}_{(1)} - \mathbf{E}_2^p + \frac{\mathbf{F}_2^p}{\beta_3} \right\|_F^2 .\end{aligned} \tag{27}$$

Optimizing this problem can be treated as solving the following linear system:

$$\mathbf{A}\mathbf{Z}_{(1)} = \mathbf{B}, \tag{28}$$

where $\mathbf{A} = \beta_2 + \beta_3\mathbf{D}_1^T\mathbf{D}_1 + \beta_3\mathbf{D}_2^T\mathbf{D}_2$ and $B = \beta_2\mathbf{M}_{(1)}^{l+1,p+1} + \mathbf{Q}_{(1)}^p + \beta_3\mathbf{D}_1^T\mathbf{E}_1^p - \mathbf{D}_1^T\mathbf{F}_1^p + \beta_3\mathbf{D}_2^T\mathbf{E}_2^p - \mathbf{D}_2^T\mathbf{F}_2^p$.

Assuming the periodic boundary condition, $\mathbf{D}_1$ and $\mathbf{D}_2$ have block circulant with circulating block (BCCB) structure. After applying 2D FFTs, the optimal $\mathcal{Z}$ is formed directly

$$\mathcal{Z} = \text{fold}_{(1)}\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathbf{B})}{\mathcal{F}(\mathbf{A})}\right)\right). \tag{29}$$

4. **E-subproblem** For the **E**-subproblem, we solve the following optimization problem:

$$\begin{aligned}\mathbf{E}^{p+1} &= \arg\min_{\mathbf{E}} \lambda \sum_{m=1}^{n_1}\sum_{n=1}^{s} \left\|\mathbf{E}_{m,n}\right\|_2 + \frac{\beta_3}{2}\left\|\mathbf{D}_1\mathbf{Z}_{(1)}^{p+1} - \mathbf{E}_1\right\|_F^2 + \langle \mathbf{D}_1\mathbf{Z}_{(1)}^{p+1} - \mathbf{E}_1, \mathbf{F}_1^p\rangle \\ &\quad + \frac{\beta_3}{2}\left\|\mathbf{D}_2\mathbf{Z}_{(1)}^{p+1} - \mathbf{E}_2\right\|_F^2 + \langle \mathbf{D}_2\mathbf{Z}_{(1)}^{p+1} - \mathbf{E}_2, \mathbf{F}_2^p\rangle \\ &= \arg\min_{E} \lambda \sum_{m=1}^{n_1}\sum_{n=1}^{s} \left\|\mathbf{E}_{m,n}\right\|_2 + \frac{\beta_3}{2}\left\|\mathbf{D}_1\mathbf{Z}_{(1)}^{p+1} - \mathbf{E}_1 + \frac{\mathbf{F}_1^p}{\beta_3}\right\|_F^2 \\ &\quad + \frac{\beta_3}{2}\left\|\mathbf{D}_2\mathbf{Z}_{(1)}^{p+1} - \mathbf{E}_2 + \frac{\mathbf{F}_2^p}{\beta_3}\right\|_F^2 .\end{aligned} \tag{30}$$

Given fixed $\mathbf{Z}_{(1)}$ and $\mathbf{F}_i$, the optimal **E** consists of solving $n_1 s$ independent two-variable minimization problems

$$\begin{aligned}\arg\min_{(\mathbf{E}_1,\mathbf{E}_2)_{m,n}} \quad & \lambda\sqrt{\left[(\mathbf{E}_1)_{m,n}\right]^2 + \left[(\mathbf{E}_2)_{m,n}\right]^2} + \frac{\beta_3}{2}\left[(\mathbf{E}_1)_{m,n} - (\mathbf{D}_1\mathbf{Z}_{(1)}^{p+1})_{m,n} - \frac{1}{\beta_3}(\mathbf{F}_1^p)_{m,n}\right]^2 \\ & + \frac{\beta_3}{2}\left[(\mathbf{E}_2)_{m,n} - (\mathbf{D}_2\mathbf{Z}_{(1)}^{p+1})_{m,n} - \frac{1}{\beta_3}(\mathbf{F}_2^p)_{m,n}\right]^2 .\end{aligned} \tag{31}$$

And (31) can be solved by using the 2-D shrinkage formula as follows:

$$[(\mathbf{E}_1)_{m,n}, (\mathbf{E}_2)_{m,n}] = \max\left\{\left\|\mathbf{W}_{m,n}\right\|_2 - \frac{\lambda}{\beta_3}, 0\right\}\frac{\mathbf{W}_{m,n}}{\left\|\mathbf{W}_{m,n}\right\|_2}, 1 \le m \le n_1, 1 \le n \le s, \tag{32}$$

where $\mathbf{W}_{m,n} = [(\mathbf{D}_1\mathbf{Z}_{(1)}^{p+1})_{m,n} + \frac{1}{\beta_3}(\mathbf{F}_1^p)_{m,n}, (D_2\mathbf{Z}_{(1)}^{p+1})_{m,n} + \frac{1}{\beta_3}(\mathbf{F}_2^p)_{m,n}]$ for $1 \le m \le n_1, 1 \le n \le s$, and set $0 \cdot (0/0) = 0$.

Now we discuss the computational complexity of the $\mathcal{M}$ problem. The $\mathcal{A}_k$-subproblem (25) involves the product of $\mathbf{X}_k$ and $\mathbf{Y}_k$ with sizes $(\prod_{d=1}^{k} n_d) \times r_k$ and $r_k \times (\prod_{d=k+1}^{j} n_d)$, whose complexity is $O(r_k \prod_{d=1}^{j} n_d)$. The main computation of the $\mathcal{Z}_k$-subproblem (29) is the fast Fourier transforms on the matrix with size $\prod_{d=1}^{j} n_d$, whose complexity is $O(\prod_{d=1}^{j} n_d \log \prod_{d=1}^{j} n_d)$. The complexity of $[\mathbf{E}_1, \mathbf{E}_2]$ (32) is $O(\prod_{d=1}^{j} n_d)$. The computational task of solving the $\mathcal{M}$ problem involves a few inner iterations and its complexity is $O(\sum_{k=1}^{j-1} r_k \prod_{d=1}^{j} n_d + \prod_{d=1}^{j} n_d \log \prod_{d=1}^{j} n_d)$.

We summarize the proposed algorithm in Algorithm 1. In each iteration, the total cost of computing $\mathbf{X}_k$, $\mathbf{Y}_k$, and $\mathcal{M}$ in Algorithm 1 is

$$O\left(\left(\sum_{k=1}^{j-1} r_k + \log \prod_{d=1}^{j} n_d\right)\prod_{d=1}^{j} n_d\right).$$

Algorithm 1 BSUM-based algorithm for tensor completion.

```
Input: The observed tensor $\mathcal{Y}$, index set $\Omega$.
Parameters: $\alpha_k, r_k, k = 1, \ldots, j-1, \lambda, \beta_1, \beta_2, \beta_3$, and $\rho$.
Initialize: $\mathbf{X}^0, \mathbf{Y}^0, \mathcal{M}^0$, out iteration $l_{max}$, inner iteration $p_{max}$.
Out loop: While not converged and $l \leq l_{max}$
    for $k = 1$ to $j-1$ do
        update $\mathbf{X}_k$ via (17);
        update $\mathbf{Y}_k$ via (17);
        Inner loop: While not converged and $p \leq p_{max}$ do
            update $\mathcal{M}$ via (22);
            update $\mathcal{A}_k$ via (25);
            update $\mathcal{Z}$ via (29);
            update $\mathbf{E}$ via (32);
            update $\mathcal{B}_k$, $\mathcal{Q}$, and $\mathbf{F}$ via (20), respectively.
        end while
end while
Output: Recovered data $\mathcal{M}$.
```

3.3 Convergence Analysis

In the subsection, we study the convergence of the proposed algorithm. Recently, Razaviyayn et al. [30] proposed the BSUM algorithm for the non-smooth optimization problem. It is an alternative inexact block coordinate descent method. Following, we reviewed the convergence result in [30] for convenience.

Lemma 1 *Given the problem* min $f(x)$, *and subject to* $x \in \mathcal{X}$, *where* $\mathcal{X}$ *is the feasible set. Assume* $u(x, x^{l-1})$ *is an approximation of* $f(x)$ *at the* $(l-1)$*-th iteration, which satisfied the following conditions:*

$$\begin{cases} u_i(y_i, y) = f(y), \forall y \in \mathcal{X}, \forall i; \\ u_i(x_i, y) \geq f(y_1, \ldots, y_{i-1}, x_i, y_{i+1}, \ldots, y_n), \forall x_i \in \mathcal{X}_i, \forall y \in \mathcal{X}, \forall i; \\ u_i'(x_i, y; d_i)|_{x_i = y_i} = f'(y; d), \forall d = (0, \ldots, d_i, \ldots, 0), s.t. y_i + d_i \in \mathcal{X}_i, \forall i; \\ u_i(x_i, y) \textit{ is continuous in } (x_i, y), \forall i; \end{cases} \tag{33}$$

where $u_i(x_i, y)$ *is the sub-problem with respect to the* i*-th block and* $f'(y; d)$ *is the direction derivative of* f *at the point* y *in direction* d. *Suppose* $u_i(x_i, y)$ *is quasi-convex in* x_i *for* $i = 1, \ldots, n$. *Furthermore, assume that each sub-problem* arg min $u_i(x_i, x^{l-1})$, $s.t. x \in \mathcal{X}_i$ *has a unique solution for any point* $x^{l-1} \in \mathcal{X}$. *Then, the iterates generated by the BSUM algorithm converge to the set of coordinatewise minimum of* f. *In addition, if* $f(\cdot)$ *is regular at* z, *then* z *is a stationary point.*

Next, we will show that the convergence of the proposed algorithm for the model (7) is guaranteed, as it fits the framework of the BSUM method.

Theorem 1 *The iterates generated by* (16) *converge to the set of coordinatewise minimizers.*

Proof It is easy to verify that $h(\mathcal{Z}, \mathcal{Z}^k)$ is an approximation and a global upper bound of $f(\mathcal{Z})$ at the k-th iteration, which satisfies the following conditions:

$$\begin{cases} h_i(\mathcal{Z}_i, \mathcal{Z}) = f(\mathcal{Z}), \ \forall \mathcal{Z}, \ i = 1, 2, 3; \\ h_i(\bar{\mathcal{Z}}_i, \mathcal{Z}) \geq f(\mathcal{Z}_1, \ldots, \bar{\mathcal{Z}}_i, \ldots, \mathcal{Z}_3), \forall \bar{\mathcal{Z}}_i, \forall \mathcal{Z}, i = 1, 2, 3; \\ h_i'(\bar{\mathcal{Z}}_i, \mathcal{Z}; d_i)|_{\bar{\mathcal{Z}}_i = \mathcal{Z}_i} = f'(\mathcal{Z}; d), \forall d = (0, \ldots, d_i, \ldots, 0), i = 1, 2, 3; \\ h_i(\bar{\mathcal{Z}}_i, \mathcal{Z}) \text{ is continuous in } (\bar{\mathcal{Z}}_i, \mathcal{Z}), i = 1, 2, 3; \end{cases} \tag{34}$$

where $\mathcal{Z} = (\mathbf{X},\ \mathbf{Y},\ \mathcal{M})$, and $\mathcal{Z}_i$ is equal to $\mathbf{X}, \mathbf{Y}, \mathcal{M}$, for $i = 1, 2, 3$, respectively. In addition, the sub-problem $h_i (i = 1, 2, 3)$ is strictly convex with respect to $\mathbf{X}$, $\mathbf{Y}$ and $\mathcal{M}$ respectively and thus each sub-problem has a unique solution. Therefore, all assumptions in Lemma 1 are satisfied. □

4 Numerical Experiments

In this section, we evaluate the performance of the proposed method for the reconstruction of color images, multispectral images (MSIs), MRI, and color videos. We compare the results with five well-known tensor completion methods, including SiLRTC [22], TMac [35], the method based on the tensor nuclear norm in [23] ("TNN" for short), SiLRTC-TT, and TMac-TT [1]. All test tensors are rescaled between [0, 255] to allow a fair quantitative evaluation. In color videos tests, as TNN is only applicable to third-order tensors, we perform it on each frame separately.

The quality of recovered images is measured by the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM), which are defined as

$$\text{PSNR} = 10 \log_{10} \frac{\text{Max}^2_{\mathbf{M},\mathbf{M}^*}}{\|\mathbf{M} - \mathbf{M}^*\|_F^2}$$

and

$$\text{SSIM} = \frac{(2\mu_{\mathbf{M}}\mu_{\mathbf{M}^*})(2\sigma_{\mathbf{M}\mathbf{M}^*} + c_2)}{(\mu^2_{\mathbf{M}} + \mu^2_{\mathbf{M}^*} + c_1)(\sigma^2_{\mathbf{M}} + \sigma^2_{\mathbf{M}^*} + c_2)},$$

where $\mathbf{M}^*$ is the true image, $\mathbf{M}$ is the recovered image, $\text{Max}_{\mathbf{M},\mathbf{M}^*}$ is the maximum pixel value of the images $\mathbf{M}$ and $\mathbf{M}^*$, $\mu_{\mathbf{M}}$ and $\mu_{\mathbf{M}^*}$ are the mean values of images $\mathbf{M}$ and $\mathbf{M}^*$, $\sigma_{\mathbf{M}}$ and $\sigma_{\mathbf{M}^*}$ are the standard variances of $\mathbf{M}$ and $\mathbf{M}^*$, respectively, $\sigma_{\mathbf{M}\mathbf{M}^*}$ is the covariance of $\mathbf{M}$ and $\mathbf{M}^*$, and c_1 and $c_2 > 0$ are constants. By calculating average PSNR and SSIM values for all bands, we obtain the PSNR and SSIM values of a higher-order tensor. Higher PSNR and SSIM values imply better image quality.

The convergence criterion of our proposed algorithm is defined by computing the relative error of the tensor $\mathcal{M}$ between two successive iterations as follows:

$$\frac{\|\mathcal{M}^{l+1} - \mathcal{M}^l\|_F}{\|\mathcal{M}^l\|_F} \leq 10^{-4}. \tag{35}$$

In the experiments, there are two parameters that should be initialized: the weighting parameters α and the initial ranks for TMac, TMac-TT, and MF-TTTV. Firstly, the weights α_k for TMac-TT and MF-TTTV are defined as

$$\alpha_k = \frac{\delta_k}{\sum_{k=1}^{j-1} \delta_k} \quad \text{with} \quad \delta_k = \min(\Pi_{d=1}^k n_d, \Pi_{d=k+1}^j n_d), \tag{36}$$

where $k = 1, \ldots, j - 1$. In this way, we assign the large weights to the more balanced matrices. Similarly, for TMac, the weights α_k are defined as

$$\alpha_k = \frac{n_k}{\sum_{k=1}^j n_k}, \tag{37}$$

where $k = 1, \ldots, j$. Secondly, to obtain the initial ranks for TMac, TMac-TT, and MF-TTTV, each rank r_k is approximated by the numbers of the singular values that satisfies the following inequality:

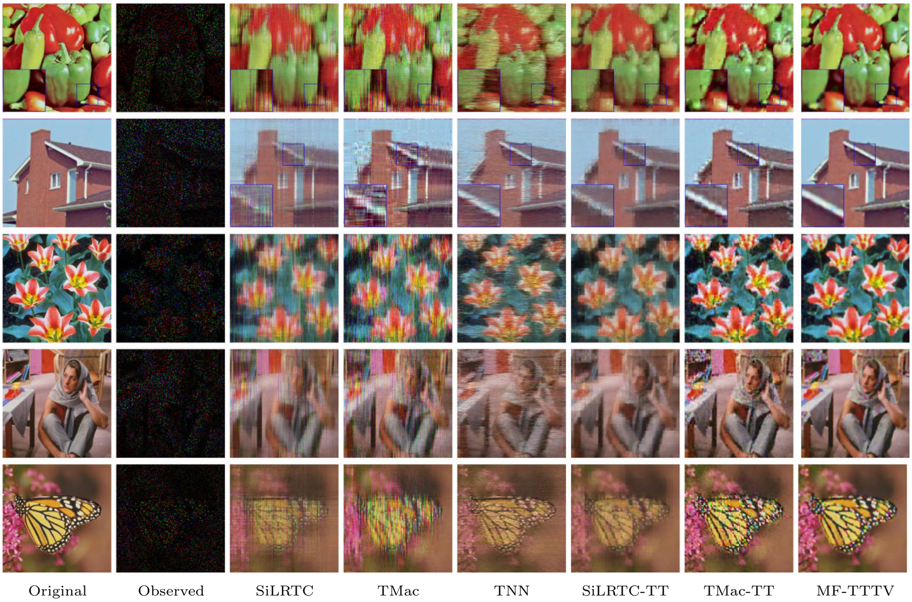


Fig. 3 The results of testing color images with $SR = 0.1$ recovered by different methods. From left to right: the original data, the observed data, the reconstructed results by SiLRTC, TMac, TNN, SiLRTC-TT, TMac-TT, and MF-TTTV, respectively (Color figure online)

$$\frac{\sigma_q^{[k]}}{\sigma_1^{[k]}} > th, \tag{38}$$

where $q = 1, \ldots, r_k$, th is the threshold, and $\sigma_q^{[k]}$ is assumed to be in descending order. This condition is chosen such that the matricizations with lower-rank will have more singular values truncated. Parameters in competing methods SiLRTC, TNN, and SiLRTC-TT are automatically optimized according to the author's suggestion.

In our method, we set the proximal parameter $\rho = 10^{-3}$, penalty parameters $\beta_1 = 5{*}10^{-3}$, and $\beta_2 = 0.1$ for all experiments. And we empirically select the regularization parameter λ and the penalty parameter β_3 from the candidate set: $\{0.01, 0.03, 0.05, 0.1, 0.3\}$, to attain the highest PSNR value.

All numerical experiments are performed on Windows 10 64-bit and MATLAB R2012a running on a desktop equipped with an Intel(R) Core(TM) i7-6700M CPU with 3.40 GHz and 8 GB of RAM.

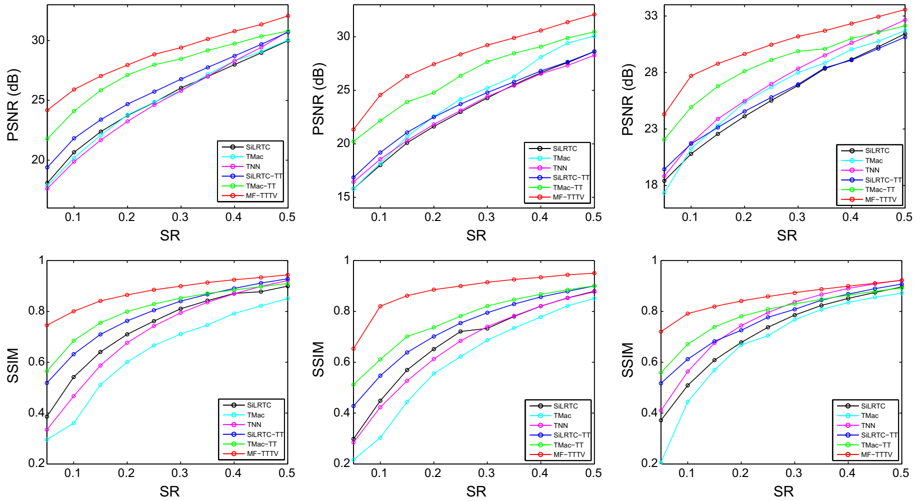
4.1 Color Images

In this subsection, we evaluate the performance of the proposed method on color images. The size of the testing image is 256 × 256 × 3. By applying KA, we cast a third-order tensor $\mathcal{M} \in \mathbb{R}^{256\times256\times3}$ into a ninth-order tensor $\tilde{\mathcal{M}} \in \mathbb{R}^{4\times4\times4\times4\times4\times4\times4\times4\times3}$ and then apply the proposed method to restore color images. The sampling rate (SR) is tested from 0.05 to 0.5.

Figure 3 shows the visual results of color images with $SR = 0.1$ recovered by SiLRTC, TMac, TNN, SiLRTC-TT, TMac-TT, and MF-TTTV, respectively. Obviously, there are several vertical thorns in the recovered images by SiLRTC and TMac, the results by TNN appear

Table 2 The PSNR and SSIM values obtained by compared methods and the proposed method for color images with different sampling rates (SRs)

Image	SR	0.1		0.2		0.3		0.4	
	Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	SiLRTC	20.66	0.5416	23.74	0.7099	26.04	0.8103	27.99	0.8708
	TMac	20.20	0.3601	23.79	0.6007	25.82	0.7107	28.31	0.7917
	TNN	19.88	0.4667	23.25	0.6765	25.81	0.7942	28.30	0.8695
	SiLRTC-TT	21.83	0.6318	24.69	0.7626	26.78	0.8402	28.72	0.8905
	TMac-TT	24.11	0.6848	27.12	0.7992	28.47	0.8522	29.75	0.8845
	MF-TTTV	**25.91**	**0.8005**	**27.96**	**0.8649**	**29.41**	**0.8993**	**30.78**	**0.9245**
Peppers	SiLRTC	18.01	0.4485	21.62	0.6516	24.27	0.7330	26.65	0.8205
	TMac	18.17	0.3031	22.56	0.5555	25.22	0.6866	28.10	0.7781
	TNN	18.57	0.4232	21.82	0.6126	24.41	0.7401	26.54	0.8210
	SiLRTC-TT	19.19	0.5469	22.50	0.7014	24.79	0.7948	26.81	0.8569
	TMac-TT	22.17	0.6111	24.78	0.7370	27.66	0.8210	29.08	0.8675
	MF-TTTV	**24.35**	**0.8205**	**27.44**	**0.8855**	**29.22**	**0.9147**	**30.59**	**0.9342**
House	SiLRTC	20.79	0.5090	24.12	0.6780	26.85	0.7856	29.16	0.8511
	TMac	21.25	0.4442	25.36	0.6681	27.99	0.7675	30.05	0.8352
	TNN	21.75	0.5634	25.48	0.7442	28.34	0.8368	30.62	0.8904
	SiLRTC-TT	21.69	0.6121	24.56	0.7258	26.95	0.8082	29.09	0.8681
	TMac-TT	24.93	0.6709	28.11	0.7806	29.88	0.8297	31.01	0.8643
	MF-TTTV	**27.70**	**0.7912**	**29.64**	**0.8410**	**31.19**	**0.8727**	**32.33**	**0.8999**
Tulips	SiLRTC	15.51	0.3032	18.95	0.5273	21.63	0.6579	23.97	0.7710
	TMac	15.12	0.2387	19.53	0.4718	22.64	0.6346	25.02	0.7471
	TNN	16.09	0.3656	19.34	0.5759	21.88	0.7139	24.36	0.8154
	SiLRTC-TT	16.77	0.4194	19.96	0.6121	22.30	0.7312	24.56	0.8196
	TMac-TT	19.89	0.5438	22.71	0.6946	24.46	0.7684	27.21	0.8505
	MF-TTTV	**22.24**	**0.7161**	**25.55**	**0.8493**	**27.94**	**0.9003**	**29.63**	**0.9297**
Barbara	SiLRTC	19.18	0.4498	22.46	0.6512	24.95	0.7531	27.15	0.8299
	TMac	18.94	0.3412	23.48	0.6093	25.77	0.7371	28.03	0.8072
	TNN	19.34	0.4511	22.43	0.6470	25.01	0.7740	27.48	0.8529
	SiLRTC-TT	20.41	0.5481	23.29	0.7065	25.43	0.8005	27.45	0.8634
	TMac-TT	22.95	0.6391	25.05	0.7461	27.65	0.8371	29.43	0.8782
	MF-TTTV	**24.44**	**0.7666**	**26.89**	**0.8466**	**28.44**	**0.8832**	**30.12**	**0.9131**
Monarch	SiLRTC	17.29	0.4622	19.51	0.6226	21.63	0.7455	23.90	0.8313
	TMac	16.63	0.3761	18.41	0.4850	22.15	0.6796	24.94	0.7841
	TNN	18.43	0.5530	21.28	0.7190	23.76	0.8245	26.38	0.8913
	SiLRTC-TT	17.97	0.6044	20.04	0.7330	22.04	0.8198	24.05	0.8794
	TMac-TT	18.70	0.6283	22.77	0.7917	24.94	0.8518	27.49	0.9054
	MF-TTTV	**21.47**	**0.8170**	**24.85**	**0.8966**	**27.86**	**0.9493**	**30.28**	**0.9681**



Lena *Peppers* *House*

Fig. 4 The PSNR and SSIM values of the reconstructed color image results with different sampling rates by SiLRTC, TMac, TNN, SiLRTC-TT, TMac-TT, and MF-TTTV, respectively (Color figure online)

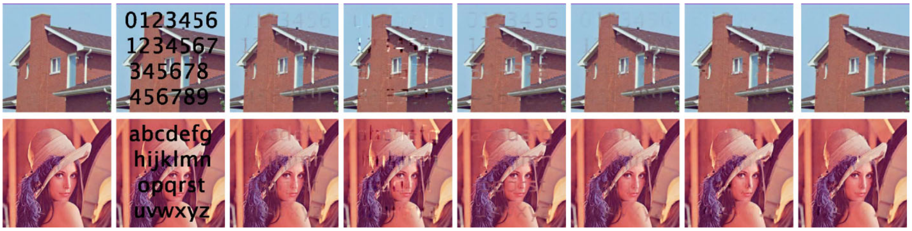


Fig. 5 The results of images *House* and *Lena* with structural missing entries described by the black letters by SiLRTC, TMac, TNN, SiLRTC-TT, TMac-TT, and MF-TTTV, respectively

dark and have horizontal thorns, and block-artifacts can be easily observed on the restored images by SiLRTC-TT and TMac-TT. While the recovered results by the proposed method are visually better than compared methods and keep the smoothness best. From the zoom-in regions of recovered images, the proposed method can effectively reduce block-artifacts compared with SiLRTC-TT and TMac-TT. Table 2 lists the numerical results by different methods, where the best PSNR and SSIM values are emphasized in bold font. Figure 4 shows the recovery PSNR and SSIM curves by different methods when applied to images *Lena*, *Peppers*, and *House*. We can see that for different SRs, the method performs better than the compared methods in terms of the PSNR and SSIM.

In Fig. 5, we show the results by applying different methods on images *House* and *Lena* with structural missing entries. The missing entries are selected as the black text. The outlines of the text can still be clearly seen on the recovered images by the compared methods, while the text is almost removed using MF-TTTV.

In Table 3, we show the PSNR, SSIM, and Time (in seconds) values of the restored images by different methods. We test the color images *Lena* and *Barbara* with $SR = 0.1$ and 0.3. It can be seen that our method outperforms the compared ones in terms of both PSNR and

Table 3 The PSNR (dB), SSIM, and Time (in seconds) values obtained by compared methods and the proposed method for images *Lena* and *Barbara* with different SRs

Image	SR	0.1			0.3		
	Method	PSNR	SSIM	Time	PSNR	SSIM	Time
Lena	SiLRTC	20.66	0.5416	30.91	26.04	0.8103	19.34
	TMac	20.20	0.3601	**2.92**	25.82	0.7107	**1.01**
	TNN	19.88	0.4667	4.85	25.81	0.7942	4.33
	SiLRTC-TT	21.83	0.6318	77.95	26.78	0.8402	43.21
	TMac-TT	24.11	0.6848	46.68	28.47	0.8522	22.05
	MF-TTTV	**25.91**	**0.8005**	388.28	**29.41**	**0.8993**	153.57
Barbara	SiLRTC	19.18	0.4498	48.44	24.95	0.7531	19.27
	TMac	18.94	0.3412	**2.11**	25.77	0.7371	**1.76**
	TNN	19.34	0.4511	4.79	25.01	0.7740	4.31
	SiLRTC-TT	20.41	0.5481	107.41	25.43	0.8005	53.81
	TMac-TT	22.95	0.6391	79.19	27.65	0.8371	58.98
	MF-TTTV	**24.44**	**0.7666**	353.66	**28.44**	**0.8832**	201.25

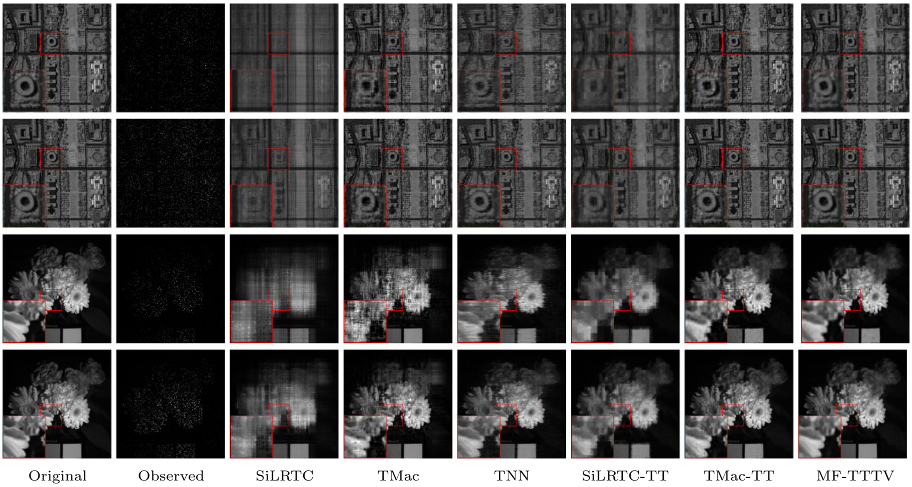
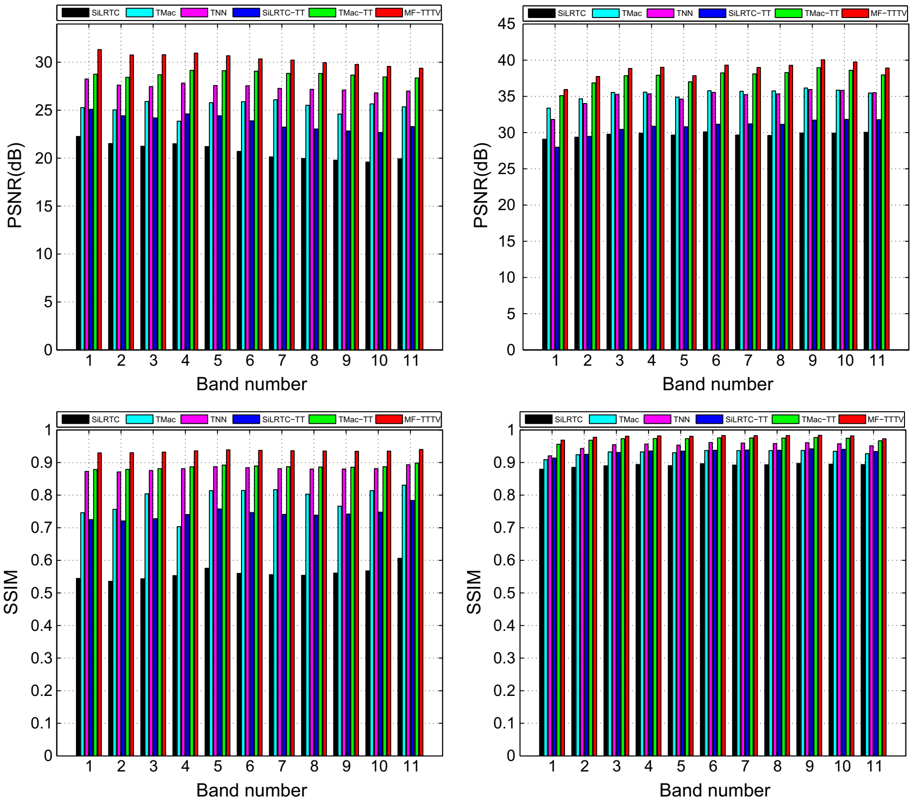


Fig. 6 The results of one band of testing MSIs recovered by different methods. The first (third) and second (fourth) rows: the results of observed MSIs *Washington DC* (*Flowers*) with $SR = 0.05$ and 0.1, respectively. From left to right: the original data, the observed data, the reconstructed results by SiLRTC, TMac, TNN, SiLRTC-TT, TMac-TT, and MF-TTTV, respectively

SSIM, although the proposed method requires more time than compared methods. The reason is that for better recovery results, we introduce the TV regularization and several variables when solving the proposed model, which lead to a more complex numerical algorithm.

4.2 MSIs Data

In this subsection, we compare the performance of different methods on MSIs *Faces*, *Flowers*, and *Washington DC* data. The testing images of size $256 \times 256 \times 11$ are selected from the



Washington DC

Faces

Fig. 7 The PSNR and SSIM values of all bands of the reconstructed MSIs with $SR = 0.1$ recovered by different methods for *Washington DC* and *Faces*

original MSIs. The third-order tensor is converted to the ninth-order tensor of size $4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 11$ by using KA. The SRs are set to be 0.05, 0.1, and 0.15, respectively. In Fig. 6, we display the recovered results by different compared methods. From the zoom-in regions of recovered images, it is clear that the proposed method performs best in preserving the edges and details of recovered images. The PSNR and SSIM values of each band of the reconstructed MSIs *Washington DC* and *Faces* with $SR = 0.1$ are shown in Fig. 7. From this figure, we can see that the PSNR and SSIM values in all bands obtained by the proposed method are better than those obtained by the compared methods. In addition, Table 4 shows the average PSNR and SSIM values of the reconstructed tensors by different methods. We can note that our proposed method obtains the highest quality results for different MSIs with different sampling rates.

4.3 MRI Data

In this subsection, we use the MRI data of size $256 \times 256 \times 11$ to compare the performance of SiLRTC, TMac, TNN, SiLRTC-TT, TMac-TT, and MF-TTTV. The SRs are set to be 0.05, 0.1, and 0.2, respectively. Table 5 lists the average PSNR and SSIM of the recovered MRI

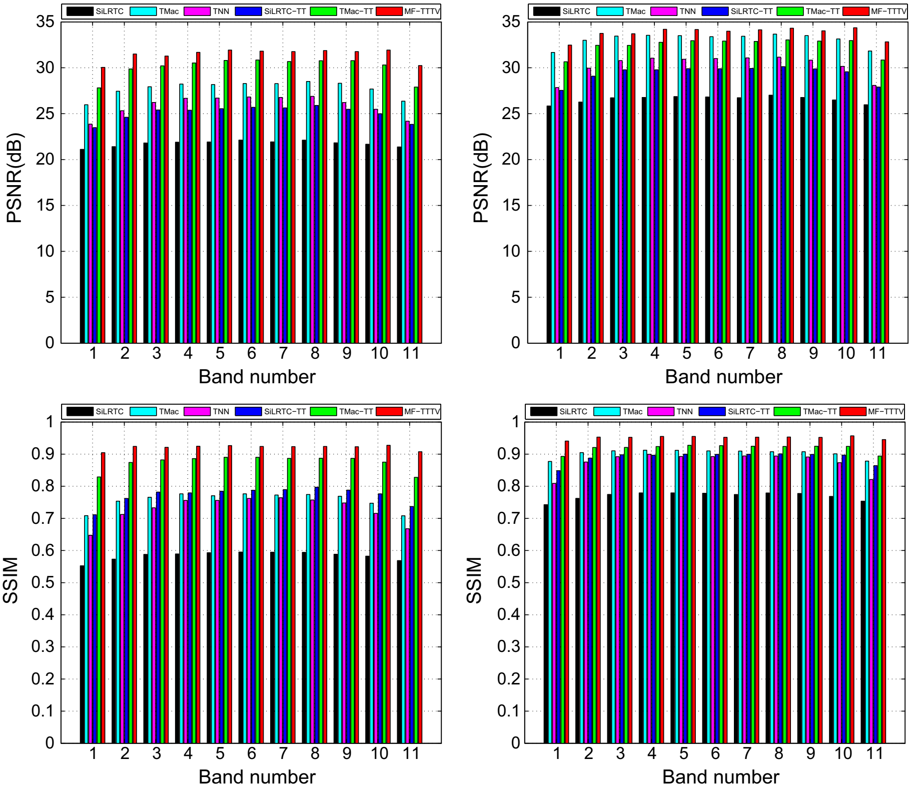
Table 4 The average PSNR and SSIM values obtained by compared methods and the proposed method for MSIs with different SRs

Image	SR	0.05		0.1		0.15	
	Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Washington DC	SiLRTC	19.12	0.4344	20.72	0.5597	22.07	0.6569
	TMac	22.39	0.6349	25.37	0.7881	27.45	0.8770
	TNN	22.56	0.6957	26.93	0.8807	30.54	0.9475
	SiLRTC-TT	21.06	0.5450	23.80	0.7428	26.30	0.8556
	TMac-TT	25.14	0.7641	28.77	0.8866	30.65	0.9424
	MF-TTTV	**25.67**	**0.8243**	**30.35**	**0.9349**	**31.98**	**0.9576**
Flowers	SiLRTC	20.70	0.5881	23.98	0.7051	26.12	0.7792
	TMac	24.12	0.6483	29.47	0.8173	31.91	0.8754
	TNN	26.19	0.7428	31.03	0.8857	34.78	0.9460
	SiLRTC-TT	23.20	0.6974	26.79	0.8268	29.20	0.8889
	TMac-TT	28.85	0.8273	33.43	0.9286	35.82	0.9538
	MF-TTTV	**30.28**	**0.8748**	**34.66**	**0.9488**	**36.79**	**0.9634**
Faces	SiLRTC	25.31	0.7778	29.74	0.8915	32.48	0.9310
	TMac	31.43	0.8774	35.35	0.9307	38.46	0.9649
	TNN	30.06	0.8815	34.95	0.9525	38.11	0.9759
	SiLRTC-TT	26.82	0.8680	30.77	0.9337	33.24	0.9596
	TMac-TT	33.60	0.9370	37.72	0.9717	38.94	0.9786
	MF-TTTV	**34.62**	**0.9613**	**38.70**	**0.9796**	**39.73**	**0.9838**

Table 5 The average PSNR and SSIM values obtained by compared methods and the proposed method for MRI data with different SRs

SR	0.05		0.1		0.2	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SiLRTC	18.39	0.4372	21.74	0.5836	26.57	0.7700
TMac	23.10	0.5480	27.74	0.7566	33.10	0.9028
TNN	22.71	0.5689	25.92	0.7291	30.26	0.8759
SiLRTC-TT	21.68	0.6283	25.08	0.7723	29.40	0.8901
TMac-TT	26.20	0.7626	30.04	0.8741	32.44	0.9184
MF-TTTV	**28.93**	**0.8806**	**31.44**	**0.9209**	**33.81**	**0.9517**

data using different methods. Figure 8 shows the PSNR and SSIM values of every band with $SR = 0.1$ and 0.2. From Table 5, we see that the proposed method achieves higher quality results for different sampling rates. The better visual performance of the recovered image by our method are shown in Fig. 9. Compared the results recovered by TMac-TT, the block-artifacts produced by using KA are obviously reduced in the restored images by MF-TTTV.



$SR = 0.1$ $SR = 0.2$

Fig. 8 The PSNR and SSIM values of all bands of the reconstructed MRI recovered by different methods

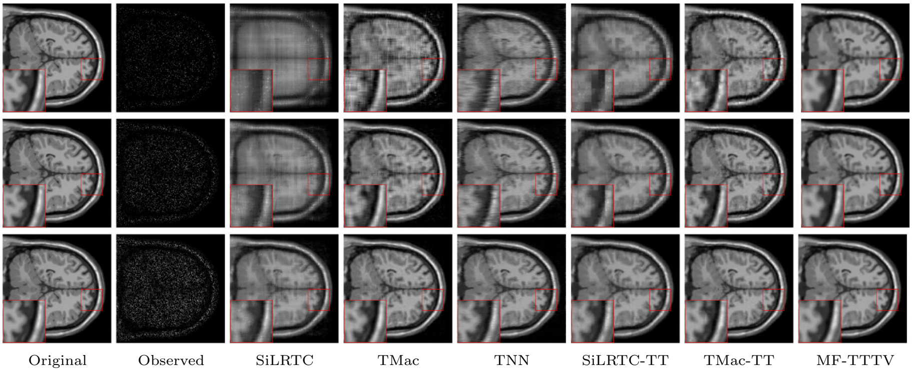


Fig. 9 The results of one band of testing MRI recovered by different methods. From top to bottom: the results of observed MRI with $SR = 0.05, 0.1$, and 0.2, respectively. From left to right: the original data, the observed data, the reconstructed results by SiLRTC, TMac, TNN, SiLRTC-TT, TMac-TT, and MF-TTTV, respectively

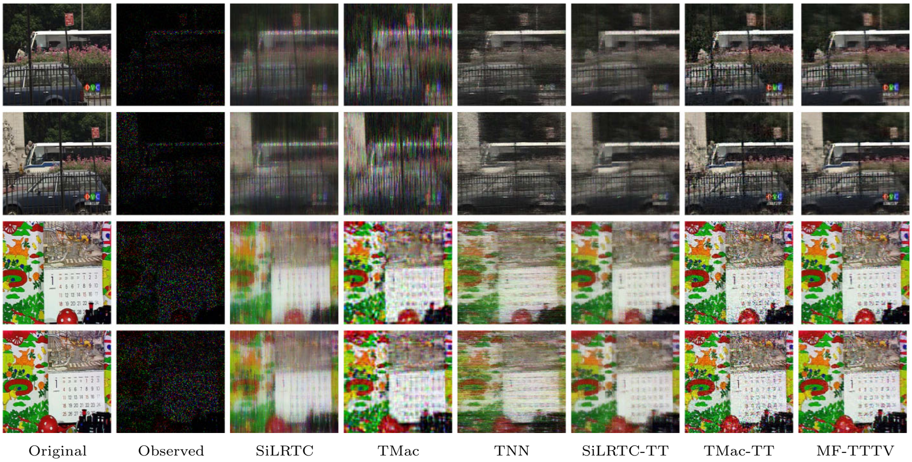


Fig. 10 The results of two frames of testing color videos recovered by different methods. The first (third) and second (fourth) rows: the results of color videos *Bus* (*Mobile*), respectively. From left to right: the original data, the observed data, the reconstructed results by SiLRTC, TMac, TNN, SiLRTC-TT, TMac-TT, and MF-TTTV, respectively (Color figure online)

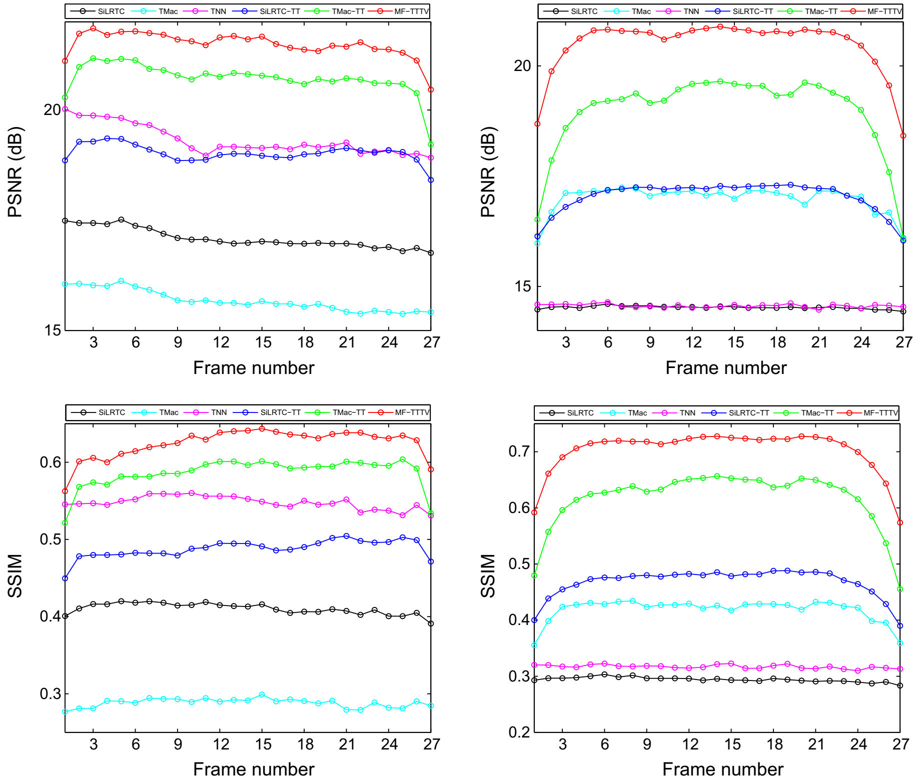
4.4 Color Videos

In this subsection, we test the proposed method on two color videos Bus and $Mobile$. The size of testing videos is $243 \times 256 \times 3 \times 27$. We reshape the tensor to a ninth-order tensor of size $6 \times 6 \times 6 \times 6 \times 6 \times 6 \times 6 \times 6 \times 3$. The ninth-order data is used for the proposed tensor completion algorithm. The SR is set as 0.1 in this task.

Results of using different methods are shown in Fig. 10. It is obvious that our method visually outperforms SiLRTC, TMac, TNN, SiLRTC-TT, and TMac-TT in keeping smoothness and details of recovered images. The PSNR and SSIM values of each frame of two reconstructed color videos are plotted in Fig. 11. We note that the PSNR and SSIM values of each frame recovered by the proposed method are higher than all compared methods.

5 Conclusion

In this paper, we propose a new model combining low-rank matrix factorization based on TT rank with the TV regularization for LRTC. An effective BSUM-based algorithm is developed to solve the proposed model with guaranteed convergence. Moreover, a tensor augmentation tool is introduced to improve the performance of our method. Numerical experiments demonstrate that the proposed method can effectively keep the piecewise smoothness of tensor data in spatial dimensions and distinctly reduce the block-artifacts compared with TMac-TT using KA. In the future work, we will try to speed up the proposed method or develop the numerical algorithm with a faster convergence rate.



Bus *Mobile*

Fig. 11 The PSNR and SSIM values of all frames of color videos recovered by different methods for *Bus* and *Mobile* (Color figure online)

Acknowledgements The authors would like to thank the anonymous referees and editor for their valuable remarks, questions, and comments that enabled the authors to improve this paper. This research is supported by the National Science Foundation of China (61772003, 61876203), the Fundamental Research Funds for the Central Universities (ZYGX2016J132, 31020180QD126), and Science Strength Promotion Programme of UESTC.

References

1. Bengua, J.A., Phiem, 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. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. Siggraph **4**(9), 417–424 (2000)
3. Cao, W.-F., Wang, Y., Sun, J., Meng, D.-Y., Yang, C., Cichocki, A., Xu, Z.-B.: Total variation regularized tensor RPCA for background subtraction from compressive measurements. IEEE Trans. Image Process. **25**(9), 4075–4090 (2016)
4. Chan, S.H., Khoshabeh, R., Gibson, K.B., Gill, P.E., Nguyen, T.Q.: An augmented Lagrangian method for total variation video restoration. IEEE Trans. Image Process. **20**(11), 3097–3111 (2011)
5. Chen, Y., Huang, T.-Z., Zhao, X.-L.: Destriping of multispectral remote sensing image using low-rank tensor decomposition. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **11**(12), 4950–4967 (2018)
6. Fu, Y., Dong, W.-S.: 3D magnetic resonance image denoising using low-rank tensor approximation. Neurocomputing **195**, 30–39 (2016)

7. Gandy, S., Recht, B., Yamada, I.: Tensor completion and low-n-rank tensor recovery via convex optimization. Inverse Probl. **27**(2), 025010 (2011)
8. Gao, S.-Q., Fan, Q.-B.: A mixture of nuclear norm and matrix factorization for tensor completion. J. Sci. Comput. **75**, 43–64 (2018)
9. Hillar, C.J., Lim, L.H.: Most tensor problems are NP-hard. J. ACM **60**(6), 45 (2013)
10. Iordache, M.D., Bioucas-Dias, J.M., Plaza, A.: Total variation spatial regularization for sparse hyperspectral unmixing. IEEE Trans. Geosci. Remote Sens. **50**(11), 4484–4502 (2012)
11. Ji, H., Liu, C., Shen, Z., Xu, Y.: Robust video denoising using low rank matrix completion. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1791–1798 (2010)
12. Ji, T.-Y., Huang, T.-Z., Zhao, X.-L., Ma, T.-H., Deng, L.-J.: A non-convex tensor rank approximation for tensor completion. Appl. Math. Model. **48**, 410–422 (2017)
13. Ji, T.-Y., Huang, T.-Z., Zhao, X.-L., Ma, T.-H., Liu, G.: Tensor completion using total variation and low-rank matrix factorization. Inf. Sci. **326**, 243–257 (2016)
14. Jiang, T.-X., Huang, T.-Z., Zhao, X.-L., Deng, L.-J., Wang, Y.: FastDeRain: a novel video rain streak removal method using directional gradient priors. IEEE Trans. Image Process. **28**(4), 2089–2102 (2019)
15. Jiang, T.-X., Huang, T.-Z., Zhao, X.-L., Ji, T.-Y., Deng, L.-J.: Matrix factorization for low-rank tensor completion using framelet prior. Inf. Sci. **436–437**, 403–417 (2018)
16. Khoromskij, B., Khoromskaia, V.: Multigrid accelerated tensor approximation of function related multidimensional arrays. SIAM J. Sci. Comput. **31**(4), 3002–3026 (2009)
17. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
18. Kolda, T.G., Bader, B.W., Kenny, J.P.: Higher-order web link analysis using multilinear algebra. In: IEEE International Conference on Data Mining, pp. 242–249 (2005)
19. Komodakis, N.: Image completion using global optimization. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 442–452 (2006)
20. Latorre, J.I.: Image compression and entanglement. (2005). arXiv:quant-ph/0510031
21. Li, F., Ng, M.K., Plemmons, R.J.: Coupled segmentation and denoising/deblurring models for hyperspectral material identification. Numer. Linear Algebra Appl. **19**(1), 153–173 (2012)
22. Liu, J., Musialski, P., Wonka, P., Ye, J.: Tensor completion for estimating missing values in visual data. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 208–220 (2013)
23. Lu, C.-Y., Feng, J.-S., Lin, Z.-C., Yan, S.-C.: Exact low tubal rank tensor recovery from Gaussian measurements. In: International Joint Conference on Artificial Intelligence (2018)
24. Luo, Y., Ward, R.K.: Removing the blocking artifacts of block-based DCT compressed images. IEEE Trans. Image Process. **12**(7), 838–842 (2003)
25. Mei, J.-J., Dong, Y.-Q., Huang, T.-Z., Yin, W.-T.: Cauchy noise removal by nonconvex admm with convergence guarantees. J. Sci. Comput. **74**, 743–766 (2018)
26. Mu, C., Huang, B., Wright, J., Goldfarb, D.: Square deal: lower bounds and improved relaxations for tensor recovery. In: International Conference on Machine Learning, pp. 73–81 (2014)
27. Oseledets, I.V.: Tensor-train decomposition. SIAM J. Sci. Comput. **33**(5), 2295–2317 (2011)
28. Oseledets, I.V., Savostianov, D.V., Tyrtyshnikov, E.E.: Tucker dimensionality reduction of three-dimensional arrays in linear time. SIAM J. Matrix Anal. Appl. **30**(3), 939–956 (2008)
29. Oseledets, I.V., Tyrtyshnikov, E., Zamarashkin, N.: Tensor-train ranks for matrices and their inverses. Comput. Methods Appl. Math. **11**(3), 394–403 (2011)
30. Razaviyayn, M., Hong, M., Luo, Z.-Q.: A unified convergence analysis of block successive minimization methods for nonsmooth optimization. SIAM J. Optim. **23**(2), 1126–1153 (2012)
31. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D Nonlinear Phenom. **60**(1–4), 259–268 (1992)
32. Varghees, V.N., Manikandan, M.S., Gini, R.: Adaptive MRI image denoising using total-variation and local noise estimation. In: International Conference on Advances in Engineering, Science and Management, pp. 506–511 (2012)
33. Wang, Y., Peng, J.-J., Zhao, Q., Leung, Y., Zhao, X.-L., Meng, D.-Y.: Hyperspectral image restoration via total variation regularized low-rank tensor decomposition. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **11**(4), 1227–1243 (2018)
34. Wang, Y.-T., Zhao, X.-L., Jiang, T.-X., Deng, L.-J., Ma, T.-H., Zhang, Y.-T., Huang, T.-Z.: A total variation and group sparsity based tensor optimization model for video rain streak removal. Signal Process. Image Commun. (2018). https://doi.org/10.1016/j.image.2018.11.008
35. Xu, Y.-Y., Hao, R.-R., Yin, W.-T., Su, Z.-X.: Parallel matrix factorization for low-rank tensor completion. Inverse Probl. Imaging **9**(2), 601–624 (2017)
36. Zhao, X.-L., Wang, F., Ng, M.: A new convex optimization model for multiplicative noise and blur removal. SIAM J. Imaging Sci. **7**(1), 456–475 (2014)

37. Zhao, X.-L., Wang, W., Zeng, T.-Y., Huang, T.-Z., Ng, M.K.: Total variation structured total least squares method for image restoration. SIAM J. Sci. Comput. **35**(6), 1304–1320 (2013)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.