



# Rank-revealing fully-connected tensor network decomposition and its application to tensor completion

Yun-Yang Liu <sup>a,b</sup>, Xi-Le Zhao <sup>a,\*</sup>, Gemine Vivone <sup>b,c</sup>

<sup>a</sup> School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, 611731, China

<sup>b</sup> Institute of Methodologies for Environmental Analysis, National Research Council, Tito Scalo, 85050, Italy

<sup>c</sup> National Biodiversity Future Center (NBFC), Palermo, 90133, Italy

## ARTICLE INFO

### Keywords:

Rank-revealing  
Fully-connected tensor network decomposition  
Tensor completion

## ABSTRACT

Fully-connected tensor network (FCTN) decomposition has become a powerful tool for handling high-dimensional data. However, for a given  $N$ th-order data,  $N(N - 1)/2$  tuning parameters (i.e., FCTN rank) in FCTN decomposition is a tricky challenge, which hinders its wide deployments. Although many recent works have emerged to adaptively search for a (near)-optimal FCTN rank, these methods suffer from expensive computational costs since they require too many search and evaluation processes, significantly limiting their applications to high-dimensional data. To tackle the above challenges, we develop a rank-revealing FCTN (revealFCTN) decomposition, whose FCTN rank is adaptively and efficiently inferred. More specifically, by analyzing the sizes of the sub-network tensors in the FCTN decomposition, we establish the equivalent relationships between the FCTN rank and the ranks of single-mode and double-mode unfolding matrices of the given data. The FCTN rank can be directly revealed through the ranks of these unfolding matrices, which does not require any search and evaluation process, making the computational cost almost negligible compared to the search-based methods. To evaluate the performance of the developed revealFCTN decomposition, we test its performance on a representative task: tensor completion (TC). Comprehensive experimental results demonstrate that our method outperforms several state-of-the-art methods, achieving a MPSNR gain of around 1 dB in most cases compared to the original FCTN decomposition.

## 1. Introduction

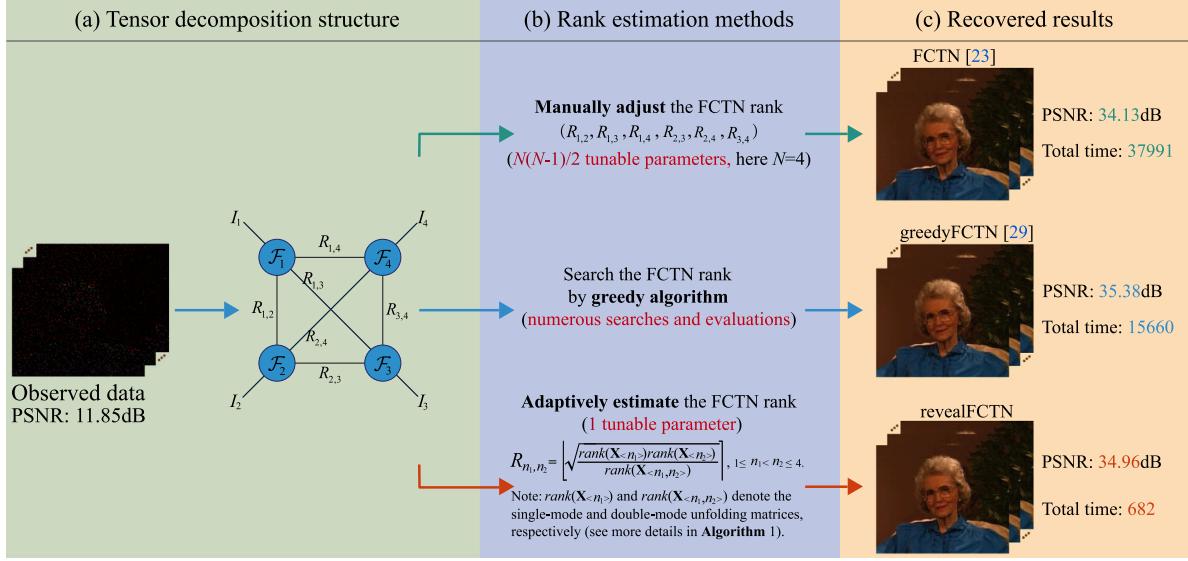
With the rapid development of science and technology, various types of high-dimensional data (e.g., multispectral images, color videos, and light field images) have emerged, which can be intuitively represented by high-dimensional tensors [1,2]. However, the “curse of dimensionality” remains a serious problem [3,4], i.e., the storage and computation costs increase exponentially with the growth of tensor order. Tensor decompositions, which decompose a high-dimensional tensor into a set of low-order latent factors, are efficient tools to address these cost issues. Recently, many tensor network (TN) decompositions have been proposed and successfully applied to image processing [5–7], computer vision [8–10], and neural network compression [11–13].

Many manually constructed TN decompositions have garnered considerable attention. Tensor train (TT) decomposition [14,15] is the most famous one, represented as a chained TN. TT decomposition breaks down an  $N$ th-order tensor into sequential multi-linear products over a sequence of factors with matrices as the first and last factors and  $N - 2$  third-order tensors as intermediate factors [16,17].

In the quantum physics community, the TT format is frequently referred to as the matrix product state representation [18,19]. By further establishing the connection between the first and last factors of TT decomposition, tensor ring (TR) decomposition [20–22] represents an  $N$ th-order tensor into sequential multi-linear products over  $N$  third-order tensors, forming a circular TN. Moreover, by establishing the connection between any two nonadjacent factors of TR decomposition, fully-connected TN (FCTN) decomposition [23–25] decomposes an  $N$ th-order tensor into sequential multi-linear products over  $N$   $N$ th-order tensors, which is a complete TN. Although FCTN decomposition has demonstrated excellent capabilities in high-dimensional data processing, a notable challenge remains: determining the  $N(N - 1)/2$  hyperparameters (i.e., FCTN rank) for FCTN decomposition in practical applications. Identifying a suitable FCTN rank through manual tuning is extremely time-consuming, as illustrated in Fig. 1(c). Furthermore, for higher-order data (e.g.,  $N = 5$ ), the cost of adjusting so many (e.g., 10) hyperparameters is prohibitive.

\* Corresponding author.

E-mail addresses: [lyymath@126.com](mailto:lyymath@126.com) (Y.-Y. Liu), [xlzhao122003@163.com](mailto:xlzhao122003@163.com) (X.-L. Zhao), [genuine.vivone@gmail.com](mailto:genuine.vivone@gmail.com) (G. Vivone).



**Fig. 1.** An overview scheme of the proposed revealFCTN. (a) Topological structure of the FCTN decomposition. (b) Rank estimation methods. (c) Recovered color video *Grandma* with SR = 5%. Our method achieves 57× and 23× speed-up compared with FCTN and greedyFCTN, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Many works have turned to investigating search-based methods [26–28] for adaptively determining a suitable FCTN rank. Hashemizadeh et al. utilized a simple greedy approach (termed as greedyFCTN) to incrementally increase the rank of the most promising edge to improve the performance of the FCTN decomposition model, but each determination of the promising edge requires simulation and comparison of  $N(N-1)/2$  cases [29]. Li and Sun represented the FCTN decomposition as an undirected graph and used a genetic algorithm to search for the optimal FCTN rank iteratively [30]. Subsequently, Li et al. developed a novel meta-heuristic method for TN permutation search by comparing the random sampling parts within a theoretically established neighborhood and then cyclically updating the neighborhood until convergence [31]. While these adaptive search-based methods typically yield excellent results in practice, they require too many search and evaluation processes, significantly increasing the computational cost and making them unaffordable for high-dimensional data [32,33], as depicted in Fig. 1(c).

To address the above challenges, we propose a rank-revealing FCTN (revealFCTN) decomposition, which fully leverages the FCTN structure and the data correlations. On the one hand, this method can adaptively determine an appropriate FCTN rank for any given data. On the other hand, this method does not require any search or evaluation process, so the computational cost is almost negligible compared to the search-based methods.

**Objective.** This study aims to develop a simple and efficient adaptive rank estimation method for the FCTN decomposition that can be applied broadly across various types of high-dimensional data, such as videos, remote sensing images, and traffic flow data.

**Contributions.** The contributions of this paper are twofold:

- We develop a revealFCTN decomposition, whose FCTN rank is efficiently estimated. Specifically, by analyzing the sizes of the sub-network tensors in the FCTN decomposition, we establish the equivalent relationships between the FCTN rank and the ranks of the single-mode and double-mode unfolding matrices of the given tensor. Thus, the FCTN rank can be directly inferred through the ranks of these unfolding matrices.
- We evaluate the potential of the revealFCTN decomposition on tensor completion (TC) task. We propose a revealFCTN decomposition-based TC model and utilize a proximal alternating minimization (PAM)-based algorithm to solve it. Extensive experimental

results show that our method achieves superior results compared to manually constructed TN decompositions in a similar or less time, while achieving comparable results to that obtained by greedyFCTN (a classic search-based method) in significantly less time.

The remainder is structured as follows. Section 2 introduces the proposed revealFCTN decomposition and the revealFCTN decomposition-based TC model. Section 3 presents extensive experiments on third-order, fourth-order, and fifth-order data to verify the effectiveness and efficiency of the proposed method. Section 4 provides a detailed discussion. Finally, the conclusion is drawn in Section 5.

**Notation.** In this paper, we use lowercase letters (e.g.,  $x \in \mathbb{R}^1$ ) to denote scalars, lowercase bold letters (e.g.,  $\mathbf{x} \in \mathbb{R}^{I_1}$ ) to denote vectors, uppercase bold letters (e.g.,  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ ) to denote matrices, and calligraphic letters (e.g.,  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ ) to denote tensors. The  $(i_1, i_2, \dots, i_N)$ -th entry of the  $N$ th-order tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is denoted as  $\mathcal{X}(i_1, i_2, \dots, i_N)$ , where  $i_n \in [I_n]$ ,  $n \in [N]$ , and  $[N]$  represents the set  $\{1, 2, \dots, N\}$ . For index  $(i_1, i_2, \dots, i_N)$ , the notation  $\overline{i_1 i_2 \dots i_N} \stackrel{\text{def}}{=} 1 + \sum_{j=1}^N (i_j - 1) \prod_{n=1}^{j-1} I_n$  is useful for unfolding matrices of a tensor. The inner product between  $\mathcal{X}$  and  $\mathcal{Y}$  is defined as  $\langle \mathcal{X}, \mathcal{Y} \rangle := \sum_{i_1, i_2, \dots, i_N} \mathcal{X}(i_1, i_2, \dots, i_N) \mathcal{Y}(i_1, i_2, \dots, i_N)$ . The Frobenius norm of  $\mathcal{X}$  is denoted as  $\|\mathcal{X}\|_F := \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$ .

## 2. The proposed revealFCTN method

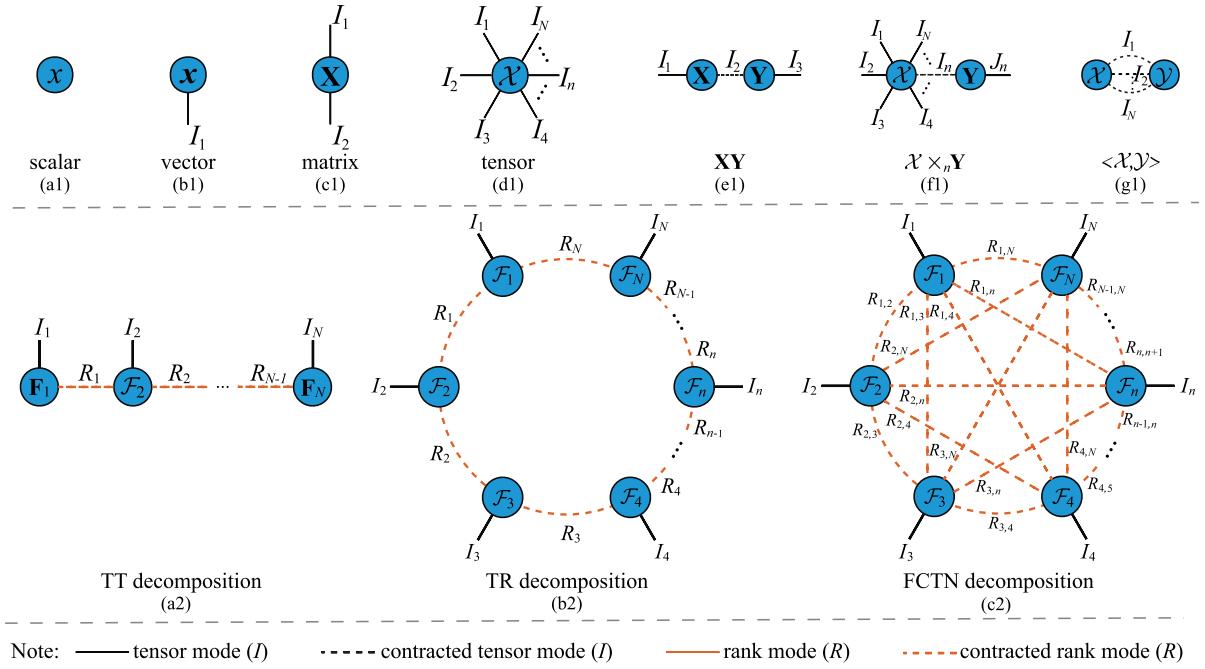
In this section, we introduce the revealFCTN decomposition and its application to TC task detailedly.

### 2.1. Preliminaries

Tensor networks offer intuitive graphical representations for tensors, tensor operations, and TN decompositions.

For an  $N$ th-order tensor, its TN representation is depicted in Fig. 2 (d1). Here, the vertex denotes the tensor, the number of edges denotes the order of the tensor, each edge denotes a mode, and the number on each edge represents the dimension of this mode. Specifically, scalar, vector, and matrix can be viewed as zero-order, first-order, and second-order tensors, respectively, as shown in Fig. 2(a1–c1).

TN can intuitively represent the tensor contraction (one of the most fundamental tensor operators) of two tensors by connecting their



**Fig. 2.** (First row) Tensor network representation of scalar, vector, matrix, tensor, matrix multiplication, tensor–matrix multiplication, and tensor inner product, respectively. (Second row) Tensor network decompositions of an  $N$ th-order tensor  $\mathcal{X}$ .

matching edges with the same dimensions, and the hanging edges represent the modes of the resulting tensor. For instance, the classic matrix multiplication between  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$  and  $\mathbf{Y} \in \mathbb{R}^{I_2 \times I_3}$  is illustrated in Fig. 2(e1), where the hanging edges (i.e.,  $I_1, I_3$ ) denote the modes of the resulting matrix, and the dotted line (i.e.,  $I_2$ ) denotes the contracted mode. Similarly, tensor–matrix multiplication and tensor inner product are presented in Fig. 2(f1) and (g1), respectively.

TNs can also naturally represent TN decompositions, which decompose an  $N$ th-order tensor into the tensor contraction of a series of factor tensors (e.g.,  $\mathcal{F}_n$ ,  $n \in [N]$ ). The TT, TR, and FCTN decompositions mentioned in the introduction are some of the extensively studied TN decompositions, as depicted in Fig. 2(a2–c2).

Given an  $N$ th-order tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , the FCTN decomposition decomposes  $\mathcal{X}$  into  $N$   $N$ th-order factor tensors  $\mathcal{F}_n \in \mathbb{R}^{R_{1,n} \times \dots \times R_{n-1,n} \times I_n \times R_{n,n+1} \times \dots \times R_{n,N}}$  ( $n = 1, 2, \dots, N$ ). The element-wise form of the FCTN decomposition is expressed as

$$\begin{aligned} \mathcal{X}(i_1, i_2, \dots, i_N) = \\ \sum_{r_{1,2}=1}^{R_{1,2}} \sum_{r_{1,3}=1}^{R_{1,3}} \dots \sum_{r_{1,N}=1}^{R_{1,N}} \sum_{r_{2,3}=1}^{R_{2,3}} \dots \sum_{r_{2,N}=1}^{R_{2,N}} \dots \sum_{r_{N-1,N}=1}^{R_{N-1,N}} \\ \{\mathcal{F}_1(i_1, r_{1,2}, r_{1,3}, \dots, r_{1,N}) \\ \mathcal{F}_2(r_{1,2}, i_2, r_{2,3}, \dots, r_{2,N}) \dots \\ \mathcal{F}_n(r_{1,n}, r_{2,n}, \dots, r_{n-1,n}, i_n, r_{n,n+1}, \dots, r_{n,N}) \dots \\ \mathcal{F}_N(r_{1,N}, r_{2,N}, \dots, r_{N-1,N}, i_N)\}. \end{aligned} \quad (1)$$

Generally, we employ  $\mathcal{X} = \text{FCTN}(\{\mathcal{F}_n\}_{n=1}^N)$  to compactly denote the above FCTN decomposition. The corresponding FCTN rank is defined as an upper triangular matrix

$$\mathbf{R} = \begin{pmatrix} 0 & R_{1,2} & R_{1,3} & \cdots & R_{1,N} \\ 0 & 0 & R_{2,3} & \cdots & R_{2,N} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & R_{N-1,N} \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}. \quad (2)$$

The FCTN rank contains  $N(N - 1)/2$  parameters, which grows quadratically with the tensor order  $N$ . This parameter problem significantly limits the widespread application of FCTN decomposition in high-order data.

## 2.2. Adaptive rank reveal of FCTN decomposition

By analyzing the structural characteristics of FCTN decomposition, the FCTN rank can be simply and directly revealed through the ranks of unfolding matrices of the high-dimensional data. More specifically, by analyzing the sizes of the sub-network tensors in FCTN decomposition, we establish the equivalent relationships between the FCTN rank and the ranks of the single-mode and double-mode unfolding matrices of the tensor. Through the equivalent relationships, we can directly infer the FCTN rank based on the ranks of these unfolding matrices.

Firstly, we define the single-mode and double-mode unfolding matrices of a tensor and their ranks.

**Definition 1 (Single-mode Unfolding Matrix and its Rank).** Given an  $N$ th-order tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , its **single-mode unfolding matrix** is  $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times \prod_{k=1}^{N, \neq n} I_k}$  ( $n \in [N]$ ) defined element-wise via

$$\mathbf{X}_{(n)}(i_n, \overline{i_1 \dots i_{n-1} i_{n+1} \dots i_N}) \stackrel{\text{def}}{=} \mathcal{X}(i_1, i_2, \dots, i_N). \quad (3)$$

The **rank** of  $\mathbf{X}_{(n)}$  is estimated by using  $\delta$ -truncated singular value decomposition (SVD) of  $\mathbf{X}_{(n)}$ , denoted as  $\text{rank}(\mathbf{X}_{(n)})$ .

The single-mode unfolding matrix of  $\mathcal{X}$  is essentially the classic mode- $k$  unfolding matrix [1] of  $\mathcal{X}$ .

**Definition 2 (Double-mode Unfolding Matrix and its Rank).** Given an  $N$ th-order tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , its **double-mode unfolding matrix** is  $\mathbf{X}_{(n_1, n_2)} \in \mathbb{R}^{I_{n_1} I_{n_2} \times \prod_{k=1}^{N, \neq n_1, \neq n_2} I_k}$  ( $1 \leq n_1 < n_2 \leq N$ ) defined element-wise via

$$\mathbf{X}_{(n_1, n_2)}(\overline{i_{n_1} i_{n_2}}, \overline{i_1 \dots i_{n_1-1} i_{n_1+1} \dots i_{n_2-1} i_{n_2+1} \dots i_N}) \stackrel{\text{def}}{=} \mathcal{X}(i_1, i_2, \dots, i_N). \quad (4)$$

The **rank** of  $\mathbf{X}_{(n_1, n_2)}$  is estimated by using  $\delta$ -truncated SVD of  $\mathbf{X}_{(n_1, n_2)}$ , denoted as  $\text{rank}(\mathbf{X}_{(n_1, n_2)})$ .

Secondly, we present some special sub-network tensors and their complementary sub-network tensors in the FCTN decomposition. The definition of sub-network tensor and its complementary sub-network tensor is as follows.

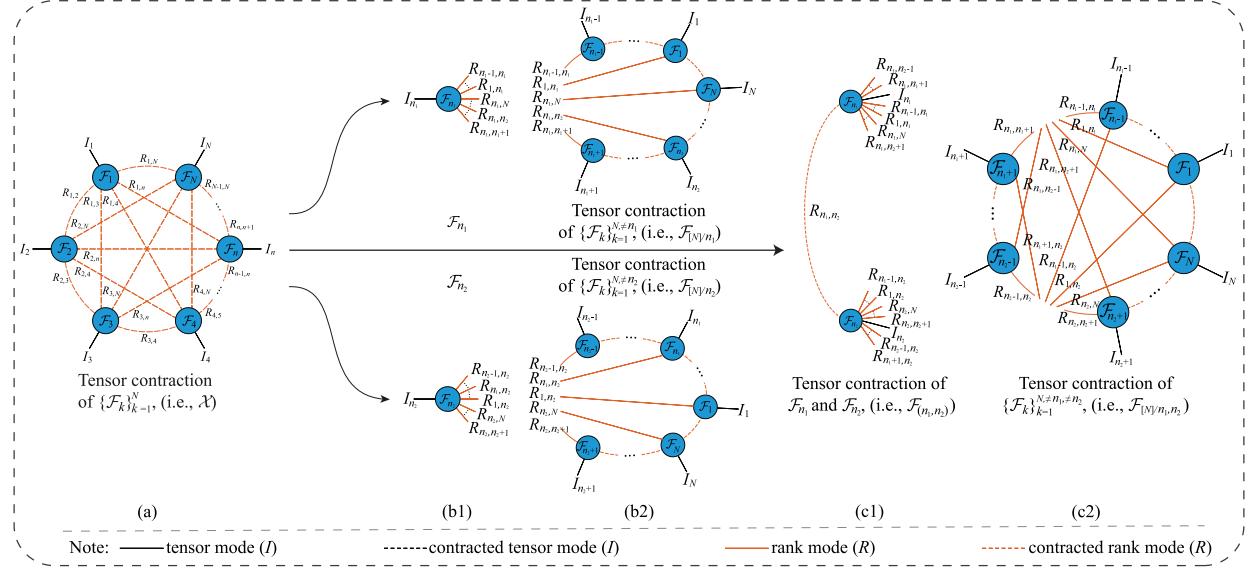


Fig. 3. The sub-network tensors in FCTN decomposition. In (b2) and (c2), we highlight the hanging rank modes and omit some contracted rank modes.

Table 1

The sizes of sub-network tensors and reshaped matrices, where  $\left[\times_{k=1}^{n-1} R_{k,n}\right]$  and  $\prod_{k=1}^{n-1} R_{k,n}$  denote  $R_{1,n} \times R_{2,n} \times \dots \times R_{n-1,n}$  and  $R_{1,n} R_{2,n} \dots R_{n-1,n}$ , respectively.

Sub-network tensor	Size	Reshaped matrices	Size
$F_n$	$\left[\times_{k=1}^{n-1} R_{k,n}\right] \times I_n \times \left[\times_{k=n+1}^N R_{n,k}\right]$	$F_n$	$I_n \times \prod_{k=1}^{n-1} R_{k,n} \prod_{k=n+1}^N R_{n,k}$
$F_{[N]/n}$	$\left[\times_{k=1}^{n-1} (I_k \times R_{k,n})\right] \times \left[\times_{k=n+1}^N (R_{n,k} \times I_k)\right]$	$F_{[N]/n}$	$\prod_{k=1}^{n-1} R_{k,n} \prod_{k=n+1}^N R_{n,k} \times \prod_{k=1}^{N-n} I_k$
$F_{(n_1, n_2)}$	$\left[\times_{k=1}^{n_1-1} R_{k,n_1}\right] \times I_{n_1} \times \left[\times_{k=n_1+1}^{n_2-1} R_{n_1,k}\right] \times \left[\times_{k=n_2+1}^N R_{n_1,k}\right]$ $\times \left[\times_{k=1}^{n_2-1} R_{k,n_2}\right] \times \left[\times_{k=n_1+1}^{n_2-1} R_{k,n_2}\right] \times I_{n_2} \times \left[\times_{k=n_2+1}^N R_{n_2,k}\right]$	$F_{(n_1, n_2)}$	$I_{n_1} I_{n_2} \times \prod_{k=1}^{n_1-1} R_{k,n_1} \prod_{k=n_1+1}^{N-n_2} R_{n_1,k} \prod_{k=1}^{n_2-1, \neq n_1} R_{k,n_2} \prod_{k=n_2+1}^N R_{n_2,k}$
$F_{[N]/n_1, n_2}$	$\left[\times_{k=1}^{n_1-1} (I_k \times R_{k,n_1}) \times R_{k,n_2}\right] \times \left[\times_{k=n_1+1}^{n_2-1} (R_{n_1,k} \times I_k) \times R_{n_2,k}\right]$	$F_{[N]/n_1, n_2}$	$\prod_{k=1}^{n_1-1} R_{k,n_1} \prod_{k=n_1+1}^{N-n_2} R_{n_1,k} \prod_{k=1}^{n_2-1, \neq n_1} R_{k,n_2} \prod_{k=n_2+1}^N R_{n_2,k} \times \prod_{k=1}^{N, \neq n_1, \neq n_2} I_k$

**Definition 3 (Sub-network Tensor and its Complementary Sub-network Tensor).** Suppose an  $N$ th-order tensor  $\mathcal{X}$  has FCTN form with factors  $\{\mathcal{F}_n\}_{n=1}^N$ ,  $n_1$  is a subset of  $\{1, 2, \dots, N\}$ ,  $n_2$  is the complementary set of  $n_1$ . Then the **sub-network tensor** is obtained by tensor contraction of  $\{\mathcal{F}_n\}_{n \in n_1}$ , denoted as  $\mathcal{F}_{n_1}$ , and its **complementary sub-network tensor** is constructed by tensor contraction of  $\{\mathcal{F}_n\}_{n \in n_2}$ , denoted as  $\mathcal{F}_{n_2}$ .

We mainly consider the sub-network tensors obtained by tensor contraction of one factor and two factors, corresponding to the single-mode and double-mode unfolding matrices, respectively. The sub-network tensor computed by tensor contraction of one factor,  $\mathcal{F}_n$  ( $n \in [N]$ ), is denoted as  $\mathcal{F}_n$ , and its complementary sub-network tensor is denoted as  $\mathcal{F}_{[N]/n}$ , which is computed by tensor contraction of the remaining  $N - 1$  factors (i.e.,  $\{\mathcal{F}_k\}_{k=1}^{N, \neq n}$ ), as illustrated in Fig. 3 (b1) and (b2). Similarly, the sub-network tensor obtained by tensor contraction of two factors,  $\mathcal{F}_{n_1}$  and  $\mathcal{F}_{n_2}$  ( $1 \leq n_1 < n_2 \leq N$ ), is denoted as  $\mathcal{F}_{(n_1, n_2)}$ , and its complementary sub-network tensor is denoted as  $\mathcal{F}_{[N]/n_1, n_2}$ , which is obtained by tensor contraction of the remaining  $N - 2$  factors (i.e.,  $\{\mathcal{F}_k\}_{k=1}^{N, \neq n_1, \neq n_2}$ ), as shown in Fig. 3 (c1) and (c2).

Thirdly, we establish the equivalent relationships between the FCTN rank and the ranks of single-mode and double-mode unfolding matrices of the given tensor in the following two lemmas. According to Definition 3,  $\mathcal{X}$  equals the tensor contraction of  $\mathcal{F}_n$  and  $\mathcal{F}_{[N]/n}$ , and also the tensor contraction of  $\mathcal{F}_{(n_1, n_2)}$  and  $\mathcal{F}_{[N]/n_1, n_2}$ . Therefore, we can induce

$$\mathbf{X}_{(n)} = \mathbf{F}_n \mathbf{F}_{[N]/n} \quad \text{and} \quad \mathbf{X}_{(n_1, n_2)} = \mathbf{F}_{(n_1, n_2)} \mathbf{F}_{[N]/n_1, n_2}, \quad (5)$$

where  $\mathbf{F}_n$ ,  $\mathbf{F}_{[N]/n}$ ,  $\mathbf{F}_{(n_1, n_2)}$ , and  $\mathbf{F}_{[N]/n_1, n_2}$  are unfolding matrices of  $\mathcal{F}_n$ ,  $\mathcal{F}_{[N]/n}$ ,  $\mathcal{F}_{(n_1, n_2)}$ , and  $\mathcal{F}_{[N]/n_1, n_2}$ , respectively. Moreover, we summarize the sizes of these sub-network tensors and their unfolding matrices in Table 1.

Assuming that Eq. (5) represents the maximum rank decomposition of  $\mathbf{X}_{(n)}$  and  $\mathbf{X}_{(n_1, n_2)}$ , we can establish the equivalence relationships between the FCTN rank (i.e.,  $R_{n_1, n_2}$ ) and the ranks of single-mode and double-mode unfolding matrices of tensor  $\mathcal{X}$  (i.e.,  $\text{rank}(\mathbf{X}_{(n)})$  and  $\text{rank}(\mathbf{X}_{(n_1, n_2)})$ ). The following two lemmas detail these equivalence relationships.

**Lemma 1.** Suppose an  $N$ th-order tensor  $\mathcal{X}$  has FCTN form, i.e.,  $\mathcal{X} = \text{FCTN}(\{\mathcal{F}_n\}_{n=1}^N)$ , the relationship between the FCTN rank  $R_{n_1, n_2}$  and the rank of single-mode unfolding matrix  $\text{rank}(\mathbf{X}_{(n)})$  is

$$\text{rank}(\mathbf{X}_{(n)}) = \prod_{k=1}^{n-1} R_{k,n} \prod_{k=n+1}^N R_{n,k}. \quad (6)$$

**Lemma 2.** Suppose an  $N$ th-order tensor  $\mathcal{X}$  has FCTN form, i.e.,  $\mathcal{X} = \text{FCTN}(\{\mathcal{F}_n\}_{n=1}^N)$ , the relationship between the FCTN rank  $R_{n_1, n_2}$  and the rank of double-mode unfolding matrix  $\text{rank}(\mathbf{X}_{(n_1, n_2)})$  is

$$\begin{aligned} & \text{rank}(\mathbf{X}_{(n_1, n_2)}) \\ &= \prod_{k=1}^{n_1-1} R_{k,n_1} \prod_{k=n_1+1}^{N, \neq n_2} R_{n_1,k} \prod_{k=1}^{n_2-1, \neq n_1} R_{k,n_2} \prod_{k=n_2+1}^N R_{n_2,k} \\ &= \frac{\prod_{k=1}^{n_1-1} R_{k,n_1} \prod_{k=n_1+1}^N R_{n_1,k} \prod_{k=1}^{n_2-1} R_{k,n_2} \prod_{k=n_2+1}^N R_{n_2,k}}{R_{n_1, n_2}^2} \\ &= \frac{\text{rank}(\mathbf{X}_{(n_1)}) \text{rank}(\mathbf{X}_{(n_2)})}{R_{n_1, n_2}^2}. \end{aligned} \quad (7)$$

According to Lemmas 1 and 2, FCTN rank can be estimated by the ranks of its single-mode and double-mode unfolding matrices.

**Algorithm 1** : FCTN rank estimation

**Input:** An  $N$ th-order tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  and prescribed relative error  $\delta$ .

**Output:** FCTN rank  $R_{n_1, n_2}$  ( $1 \leq n_1 < n_2 \leq N$ ).

- 1: **for**  $n_1 = 1$  to  $N$  **do**
- 2:   Estimate rank( $\mathbf{X}_{\langle n_1 \rangle}$ ) by  $\delta$ -truncated SVD.
- 3:   **for**  $n_2 = n_1 + 1$  to  $N$  **do**
- 4:     Estimate rank( $\mathbf{X}_{\langle n_1, n_2 \rangle}$ ) by  $\delta$ -truncated SVD.
- 5:   **end for**
- 6: **end for**
- 7: Compute FCTN rank according to equation (8).

Specifically, the **adaptively revealed FCTN rank** of a given tensor  $\mathcal{X}$  is

$$R_{n_1, n_2} = \left\lceil \sqrt{\frac{\text{rank}(\mathbf{X}_{\langle n_1 \rangle}) \text{rank}(\mathbf{X}_{\langle n_2 \rangle})}{\text{rank}(\mathbf{X}_{\langle n_1, n_2 \rangle})}} \right\rceil, \quad 1 \leq n_1 < n_2 \leq N, \quad (8)$$

where  $\mathbf{X}_{\langle n_1 \rangle}$  and  $\mathbf{X}_{\langle n_1, n_2 \rangle}$  denote the single-mode and double-mode unfolding matrices of  $\mathcal{X}$ , respectively, and  $\lceil x \rceil$  denotes rounding  $x$ .

The whole process of FCTN rank estimation is summarized in Algorithm 1.

**2.3. The revealFCTN decomposition**

Based on the adaptively revealed FCTN rank, we propose the revealFCTN decomposition.

**Algorithm 2** revealFCTN decomposition.

**Input:** An  $N$ th-order data  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  and prescribed relative error  $\delta$ .

**Output:** The factors  $\{\mathcal{F}_n\}_{n=1}^N$  of the revealFCTN decomposition.

- 1: Initialize  $t = 0$  and  $t^{\max} = 1000$ ;
- 2: Estimate the FCTN rank of  $\mathcal{X}$  by Algorithm 1, and draw factors  $\{\mathcal{F}_n^t\}_{n=1}^N$  sampled from a uniform distribution with the estimated FCTN rank;
- 3: **while** not converged and  $t < t^{\max}$  **do**
- 4:   **for**  $n = 1$  to  $N$  **do**
- 5:     Compute sub-network tensor  $\mathcal{F}_{[N]/n}^t$  according to Definition 3.
- 6:     Obtain  $\mathbf{F}_n^{t+1} = [\mathbf{X}_{\langle n \rangle} (\mathbf{F}_{[N]/n}^t)^T] / [\mathbf{F}_{[N]/n}^t (\mathbf{F}_{[N]/n}^t)^T]$ ;
- 7:     Update  $\mathcal{F}_n^{t+1}(r_{1,n}, \dots, r_{n-1,n}, i_n, r_{n,n+1}, \dots, r_{n,N}) = \mathcal{F}_n^{t+1}(i_n, r_{1,n} \dots r_{n-1,n} r_{n,n+1} \dots r_{n,N})$ ;
- 8:   **end for**
- 9:   Check convergence criteria:
- 10:     $\|\mathcal{X} - \text{revFCTN}(\{\mathcal{F}_n^{t+1}\}_{n=1}^N)\|_F / \|\mathcal{X}\|_F < 10^{-5}$ ;
- 11: **end while**

**Definition 4 (revealFCTN Decomposition).** The revealFCTN decomposition of an  $N$ th-order tensor  $\mathcal{X}$  is defined as

$$\mathcal{X} = \mathcal{F}_1 \times_{\mathbf{m}_2}^{\mathbf{n}_2} \mathcal{F}_2 \times_{\mathbf{m}_3}^{\mathbf{n}_3} \mathcal{F}_3 \dots \times_{\mathbf{m}_n}^{\mathbf{n}_n} \mathcal{F}_n \dots \times_{\mathbf{m}_N}^{\mathbf{n}_N} \mathcal{F}_N, \quad (9)$$

where  $\mathcal{F}_n \in \mathbb{R}^{R_{1,n} \times \dots \times R_{n-1,n} \times I_n \times R_{n,n+1} \times \dots \times R_{n,N}}$ ,  $R_{n_1, n_2}$  is the revealed FCTN rank,  $\mathbf{m}_n = (2, (N - (n - 1)) * 1 + 2, \dots, (N - (n - 1)) * (n - 2) + 2)$ ,  $\mathbf{n}_n = (1, 2, \dots, n - 1)$ , and  $\times_{\mathbf{m}}^{\mathbf{n}}$  denotes the tensor contraction.

We detail the tensor contraction operator. Given an  $M$ th-order tensor  $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$  and an  $N$ th-order tensor  $\mathcal{Z} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$  with  $d$  common modes. Assume that two vectors  $\mathbf{m} = (m_1, m_2, \dots, m_M)$  and  $\mathbf{n} = (n_1, n_2, \dots, n_N)$  respectively indicate the rearrangement of vectors  $(1, 2, \dots, M)$  and  $(1, 2, \dots, N)$ , satisfying  $I_{m_l} = J_{n_l}$  for  $l =$

$1, 2, \dots, d$ ,  $m_{d+1} < m_{d+2} < \dots < m_M$ , and  $n_{d+1} < n_{d+2} < \dots < n_N$ , then the **tensor contraction** between  $\mathcal{Y}$  and  $\mathcal{Z}$  can be computed by the following four steps: (i) Transpose  $\mathcal{Y}$  and  $\mathcal{Z}$  respectively in the orders  $(m_{d+1}, \dots, m_M, m_1, \dots, m_d)$  and  $(n_1, \dots, n_d, n_{d+1}, \dots, n_N)$ ; (ii) Reshape the transposed tensors into matrices  $\mathbf{Y} \in \mathbb{R}^{\prod_{l=d+1}^M I_{m_l} \times \prod_{l=1}^d I_{m_l}}$  and  $\mathbf{Z} \in \mathbb{R}^{\prod_{l=1}^d J_{n_l} \times \prod_{l=d+1}^N I_{n_l}}$ ; (iii) Compute matrix multiplication  $\mathbf{X} = \mathbf{Y}\mathbf{Z}$ ; (iv) Reshape  $\mathbf{X}$  into a  $(M + N - 2d)$ th-order tensor  $\mathcal{X} \in \mathbb{R}^{I_{m_{d+1}} \times \dots \times I_{m_M} \times_{\mathbf{n}_{d+1}}^{\mathbf{n}_d} \dots \times_{\mathbf{n}_N}^{\mathbf{n}_M}}$ . For brevity, the tensor contraction between  $\mathcal{Y}$  and  $\mathcal{Z}$  is compactly denoted as  $\mathcal{X} = \mathcal{Y} \times_{\mathbf{m}_1, \dots, \mathbf{m}_d}^{\mathbf{n}_1, \dots, \mathbf{n}_d} \mathcal{Z}$ .

For simplicity, we use  $\mathcal{X} = \text{revFCTN}(\{\mathcal{F}_n\}_{n=1}^N)$  to compactly represent the above revealFCTN decomposition. Furthermore, the factors of the revealFCTN decomposition can be obtained by Algorithm 2.

**2.4. The revealFCTN decomposition-based TC model**

To evaluate the potential of the proposed revealFCTN decomposition, we test it on a representative application, i.e., tensor completion (TC). TC aims to reconstruct the data from partial observations. Let  $\mathcal{O} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  represent the observed data from the original data  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  and  $\Omega$  denote the set of indexes of the observed entries. The revealFCTN decomposition-based TC model (revealFCTN) is formulated as

$$\min_{\mathcal{X}, \mathcal{F}_n} \ell_{\Omega}(\mathcal{X}) + \frac{1}{2} \|\mathcal{X} - \text{revFCTN}(\{\mathcal{F}_n\}_{n=1}^N)\|_F^2, \quad (10)$$

where

$$\ell_{\Omega}(\mathcal{X}) := \begin{cases} 0, & \mathcal{X}_{\Omega} = \mathcal{O}_{\Omega}, \\ \infty, & \text{otherwise}. \end{cases} \quad (11)$$

**2.5. The solving algorithm for the proposed model**

We solve the nonsmooth and nonconvex problem (10) by applying the PAM algorithm [34], which iteratively updates one variable while the others are fixed, and then repeats this procedure until some convergence criterion is satisfied. Based on the PAM algorithm, each variable in the optimization problem (10) is updated alternately as follows:

$$\begin{cases} \mathcal{F}_n^{t+1} \in \arg \min_{\mathcal{F}_n} \frac{1}{2} \left\| \mathcal{X}^t - \text{revFCTN}(\{\mathcal{F}_k^{t+1}\}_{k=1}^{n-1}, \mathcal{F}_n, \{\mathcal{F}_k^t\}_{k=n+1}^N) \right\|_F^2 \\ \quad + \frac{\rho}{2} \|\mathcal{F}_n - \mathcal{F}_n^t\|_F^2, \quad n = 1, 2, \dots, N, \\ \mathcal{X}^{t+1} \in \arg \min_{\mathcal{X}} \ell_{\Omega}(\mathcal{X}) + \frac{1}{2} \|\mathcal{X} - \text{revFCTN}(\{\mathcal{F}_n^{t+1}\}_{n=1}^N)\|_F^2 \\ \quad + \frac{\rho}{2} \|\mathcal{X} - \mathcal{X}^t\|_F^2, \end{cases}$$

where  $\rho$  and  $t$  are the positive proximal parameter and the iteration index, respectively. The solution to each subproblem is presented below.

**Update  $\mathcal{F}_n$ :** The  $\mathcal{F}_n$  ( $n = 1, 2, \dots, N$ )-subproblems at the  $t$ th iteration are

$$\mathcal{F}_n^{t+1} = \arg \min_{\mathcal{F}_n} \frac{1}{2} \|\mathbf{X}_{\langle n \rangle} - \mathbf{F}_n \mathbf{F}_{[N]/n}^t\|_F^2 + \frac{\rho}{2} \|\mathbf{F}_n - \mathbf{F}_n^t\|_F^2, \quad (12)$$

where  $\mathbf{F}_{[N]/n}^t$  is obtained by tensor contraction of  $\{\mathcal{F}_k^{t+1}\}_{k=1}^{n-1}$  and  $\{\mathcal{F}_k^t\}_{k=n+1}^N$ , and  $\mathbf{F}_{[N]/n}^t$  is the unfolding matrix of  $\mathcal{F}_{[N]/n}^t$  with size in Table 1. The problem (12) can be directly solved as

$$\mathcal{F}_n^{t+1} = [\mathbf{X}_{\langle n \rangle} (\mathbf{F}_{[N]/n}^t)^T + \rho \mathbf{F}_n^t] [\mathbf{F}_{[N]/n}^t (\mathbf{F}_{[N]/n}^t)^T + \rho \mathbf{I}]^{-1}, \quad (13)$$

and  $\mathcal{F}_n^{t+1}$  is the corresponding tensor of  $\mathbf{F}_n^{t+1}$  satisfying

$$\mathcal{F}_n^{t+1}(r_{1,n}, \dots, r_{n-1,n}, i_n, r_{n,n+1}, \dots, r_{n,N}) = \mathbf{F}_n^{t+1}(i_n, r_{1,n} \dots r_{n-1,n} r_{n,n+1} \dots r_{n,N}).$$

**Update  $\mathcal{X}$ :** The  $\mathcal{X}$ -subproblem at the  $t$ th iteration is

$$\mathcal{X}^{t+1} = \arg \min_{\mathcal{X}} \ell_{\Omega}(\mathcal{X}) + \frac{1+\rho}{2} \left\| \mathcal{X} - \frac{\text{revFCTN}(\{\mathcal{F}_n^{t+1}\}_{n=1}^N) + \rho \mathcal{X}^t}{1+\rho} \right\|_F^2, \quad (14)$$

which has the following closed-form solution

$$\mathcal{X}^{t+1} = \mathcal{O}_{\Omega} + \left( \frac{\text{revFCTN}(\{\mathcal{F}_n^{t+1}\}_{n=1}^N) + \rho \mathcal{X}^t}{1+\rho} \right)_{\Omega^C}, \quad (15)$$

**Table 2**  
Summary of all compared methods.

Algorithm	Tensor decomposition type	Is tensor rank adaptive?	Number of tunable parameters
HaLRTC	Tucker	–	–
TFTC	T-SVD	✗	1
TT	TT	✗	$N - 1$
TR	TR	✗	$N$
FCTN	FCTN	✗	$N(N - 1)/2$
greedyFCTN	FCTN	✓	Numerous searches and evaluations (greedy)
revealFCTN	FCTN	✓	1 (constructive)

**Algorithm 3** : PAM-based algorithm for revealFCTN.

**Input:** Observed data  $\mathcal{O} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , set  $\Omega$ , threshold parameter  $\delta$ , proximal parameter  $\rho$ , and maximum iteration number  $t^{\max}$ .

- 1: Initialize  $t = 0$  and  $\mathcal{X}^0$ , estimate the FCTN rank of  $\mathcal{X}^0$  by Algorithm 1, and draw factors  $\mathcal{F}_n^0 (n = 1, 2, \dots, N)$  sampled from a uniform distribution with the estimated FCTN rank;
- 2: **while** not converged and  $t < t^{\max}$  **do**
- 3:   Update  $\mathcal{F}_n^{t+1}$  via (13);
- 4:   Update  $\mathcal{X}^{t+1}$  via (15);
- 5:   **if** mod<sup>1</sup>( $t+1, 400$ ) = 0 **then**
- 6:     Update FCTN rank of  $\mathcal{X}^{t+1}$  by Algorithm 1;
- 7:     Adjust factors  $\mathcal{F}_n^{t+1} (n = 1, 2, \dots, N)$  according to the new estimated FCTN rank;
- 8:   **end if**
- 9:   Check convergence criteria:

$$\|\mathcal{X}^{t+1} - \mathcal{X}^t\|_F / \|\mathcal{X}^t\|_F < 10^{-5};$$

- 10: **end while**

**Output:** The final FCTN rank and reconstructed result  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ .

where  $\Omega^C$  denotes the complementary set of  $\Omega$ .

The PAM-based algorithm for solving model (10) is described in Algorithm 3. For Algorithm 3, we further introduce two key details:

(i) We update the FCTN rank after a certain number of iterations rather than at each iteration. Typically, the FCTN ranks of  $\mathcal{X}^t$  and  $\mathcal{X}^{t+1}$  remain unchanged, making frequent updates unnecessary. For efficiency, we update the FCTN rank every 400 steps in the subsequent experiments.

(ii) We adjust the factors  $\mathcal{F}_n^{t+1} (n = 1, 2, \dots, N)$  by discarding more and compensating less after updating the FCTN rank. Taking the first factor  $\mathcal{F}_1 \in \mathbb{R}^{I_1 \times R_{1,2} \times R_{1,3} \times R_{1,4}}$  of the fourth-order data  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$  as an example, we explain this process in detail. Suppose that the adjusted target factor is  $\hat{\mathcal{F}}_1 \in \mathbb{R}^{I_1 \times R_{1,2}^{\text{new}} \times R_{1,3}^{\text{new}} \times R_{1,4}^{\text{new}}}$ ,  $R_{1,2}^{\text{new}} = R_{1,2} + d_1$ ,  $R_{1,3}^{\text{new}} = R_{1,3} - d_2$ , and  $R_{1,4}^{\text{new}} = R_{1,4}$ , where  $d_1$  and  $d_2$  are two positive integers. To achieve it, we first augment factor  $\mathcal{F}_1$  by incorporating tensor  $\mathcal{G} \in \mathbb{R}^{I_1 \times d_1 \times R_{1,3} \times R_{1,4}}$  sampled from the uniform distribution at the end of the second mode, resulting in  $\mathcal{F}_1^{\text{new}} \in \mathbb{R}^{I_1 \times R_{1,2}^{\text{new}} \times R_{1,3}^{\text{new}} \times R_{1,4}^{\text{new}}}$ . Afterwards, we remove a tensor of size  $I_1 \times R_{1,2}^{\text{new}} \times d_2 \times R_{1,4}$  from the end of the third mode, yielding  $\mathcal{F}_1^{\text{new}} \in \mathbb{R}^{I_1 \times R_{1,2}^{\text{new}} \times R_{1,3}^{\text{new}} \times R_{1,4}^{\text{new}}}$ . No adjustment is required for the fourth mode, and  $\hat{\mathcal{F}}_1 \in \mathbb{R}^{I_1 \times R_{1,2}^{\text{new}} \times R_{1,3}^{\text{new}} \times R_{1,4}^{\text{new}}}$  is finally obtained.

### 3. Numerical experiments

To validate the effectiveness of the revealFCTN, we conduct extensive numerical experiments on high-dimensional data, including third-order multispectral images, fourth-order color videos, and fifth-order light field images.

**Parameter settings.** We compare the proposed method with six completion methods, including HaLRTC [35], TFTC [36], TT [16], and TR [20], FCTN [23], and greedyFCTN [29], which are summarized in Table 2. To eliminate the influence of the solved algorithm, the TFTC, TT, TR, FCTN, and revealFCTN models are all solved using the PAM algorithm. The algorithm parameters are set as follows: proximal parameter  $\rho = 0.1$ , maximum iteration number  $t^{\max} = 1000$ , and stopping criteria  $\|\mathcal{X}^{t+1} - \mathcal{X}^t\|_F / \|\mathcal{X}^t\|_F \leq 10^{-5}$ . In addition, the parameter of HaLRTC is set to  $\alpha / \|\alpha\|_1$ , where  $\alpha$  is a vector of length  $N$  with all elements equal to 1. The parameters of TFTC, TT, TR, and FCTN are tubal rank (i.e.,  $R^{\text{Tub}}$ ), TT rank (i.e.,  $(R_1^{\text{TT}}, R_2^{\text{TT}}, \dots, R_{N-1}^{\text{TT}})$ ), TR rank (i.e.,  $(R_1^{\text{TR}}, R_2^{\text{TR}}, \dots, R_N^{\text{TR}})$ ), and FCTN rank (i.e.,  $(R_{1,2}^{\text{FCTN}}, R_{1,3}^{\text{FCTN}}, \dots, R_{1,N}^{\text{FCTN}}, R_{2,3}^{\text{FCTN}}, \dots, R_{2,N}^{\text{FCTN}}, \dots, R_{N-1,N}^{\text{FCTN}})$ ), respectively, which are adjusted manually to achieve the best performance. For revealFCTN, we initialize  $\mathcal{X}^0$  with the recovered result of TFTC to obtain a better initial FCTN rank.

**Evaluation metrics.** For numerical experiments, the mean peak signal-to-noise ratio (MPSNR), the mean structural similarity (MSSIM) [37], and total computational time (includes the time required to tune all parameters) are selected as quality metrics.

All experiments are implemented in MATLAB (R2019a) on a computer with 64Gb RAM and Intel(R) Core(TM) i9-10900KF CPU: @3.60 GHz.

#### 3.1. Third-order multispectral images

In this subsection, we conduct experiments on third-order data. Specifically, three multispectral images<sup>1</sup> (i.e., Toy, Stuffed toys, and Flowers) of size  $256 \times 256 \times 31$  (i.e., 31 bands of size  $256 \times 256$ ), are chosen as test data. For each image, we test three sampling rates (SRs): 1%, 5%, and 10%.

Table 3 reports the MPSNR/MSSIM values and the total times (seconds) of the multispectral images recovered by different methods under different SRs. For third-order data, the FCTN decomposition is essentially the TR decomposition, thus their performance is the same. As observed, the proposed revealFCTN achieves superior results compared to manually constructed TN decompositions in a similar or even less time, while achieving comparable results to that obtained by greedyFCTN in a much less time, approximately a 5-15× speed-up.

For better visual comparison, we display the pseudo-color images (composed of the 31st, 20th, and 10th bands) of the restructured multispectral images by different methods at different SRs in Fig. 4. Moreover, we present the enlarged image of the same subregion of each multispectral image in a green box, with the corresponding residual image in a red box. Obviously, the proposed revealFCTN can not only recover the global structure, but also better preserve the local details, such as the notation in Toy, the eyes in Stuffed toys, and the texture in Flowers. As observed, our revealFCTN achieves better results than FCTN. This improvement is because our method can flexibly estimate an appropriate FCTN rank for the data at each stage (see more details in the discussion).

<sup>1</sup> Modulo operation.

<sup>1</sup> Available at <http://www.cs.columbia.edu/CAVE/databases/multispectral/>.

**Table 3**

Quality metrics and total times (seconds) of various methods at different SRs on third-order multispectral images.

Data	Method	SR = 1%			SR = 5%			SR = 10%		
		MPSNR	MSSIM	Total time	MPSNR	MSSIM	Total time	MPSNR	MSSIM	Total time
Toy	Observed	10.452	0.2119	–	10.629	0.2425	–	10.865	0.2786	–
	HaLRTC	11.022	0.2889	30	20.197	0.7384	31	28.322	0.8790	33
	TFTC	15.657	0.5316	108	28.758	0.8520	173	33.326	0.9315	326
	TT	20.752	0.5635	361	28.303	0.7786	554	32.109	0.8851	716
	TR	20.260	0.5356	882	30.119	0.8315	1703	35.729	0.9368	3411
	FCTN	20.260	0.5356	882	30.119	0.8315	1703	35.729	0.9368	3411
	greedyFCTN	<b>21.706</b>	<b>0.5815</b>	473	<b>31.634</b>	<b>0.8789</b>	1199	<b>37.654</b>	<b>0.9587</b>	2659
Stuffed toys	revealFCTN	21.052	0.5765	98	31.129	0.8679	152	37.168	0.9504	202
	Observed	12.048	0.2189	–	12.226	0.2485	–	12.460	0.2824	–
	HaLRTC	12.503	0.2884	32	16.892	0.6501	33	24.105	0.8201	35
	TFTC	16.425	0.4561	119	28.059	0.8099	187	33.224	0.9174	369
	TT	18.772	0.4628	405	28.352	0.7663	573	32.204	0.8749	778
	TR	18.722	0.4091	917	29.965	0.7874	1948	35.568	0.9151	3935
	FCTN	18.722	0.4091	917	29.965	0.7874	1948	35.568	0.9151	3935
Flowers	greedyFCTN	<b>19.556</b>	0.4373	525	<b>31.962</b>	<b>0.8585</b>	1383	<b>37.705</b>	<b>0.9523</b>	3225
	revealFCTN	18.772	<b>0.4628</b>	112	31.557	0.8500	193	37.281	0.9436	245
	Observed	13.365	0.3032	–	13.543	0.3298	–	13.779	0.3611	–
	HaLRTC	13.854	0.3736	29	18.168	0.6505	31	23.883	0.7779	32
	TFTC	18.126	0.4863	102	27.334	0.7527	158	31.634	0.8709	295
	TT	21.106	0.4997	323	27.221	0.6612	464	31.409	0.8145	659
	TR	20.581	0.4589	816	29.391	0.7367	1636	34.230	0.8788	3548
FCTN	greedyFCTN	<b>22.201</b>	0.4888	451	<b>30.412</b>	<b>0.7762</b>	1120	<b>35.551</b>	<b>0.9072</b>	2847
	revealFCTN	21.408	<b>0.5216</b>	92	30.010	0.7721	139	35.015	0.9001	189

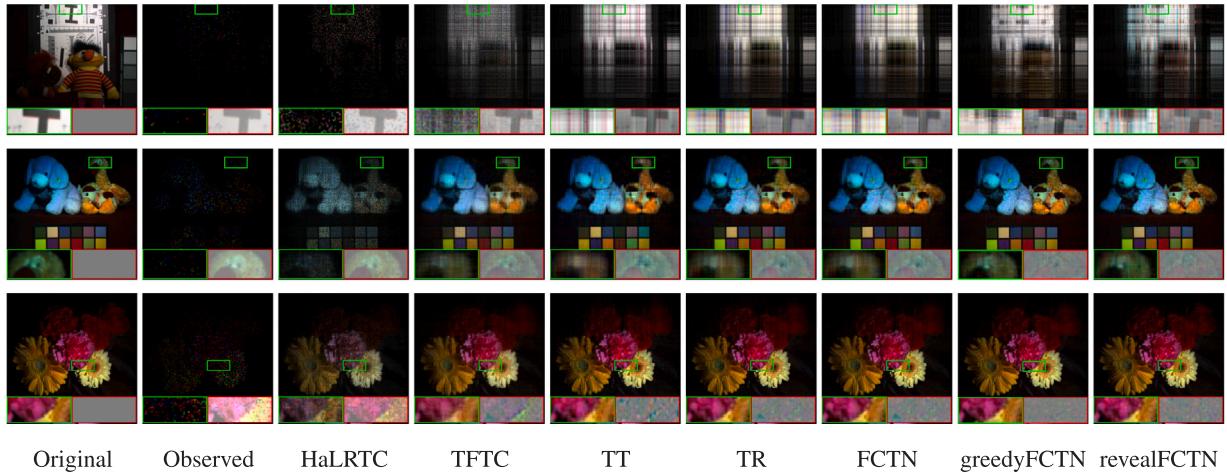


Fig. 4. Recovered results on third-order multispectral images using different methods (Green box: enlarged image; red box: corresponding residual image). From top to bottom, the sampling rates are 1%, 5%, and 10%, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

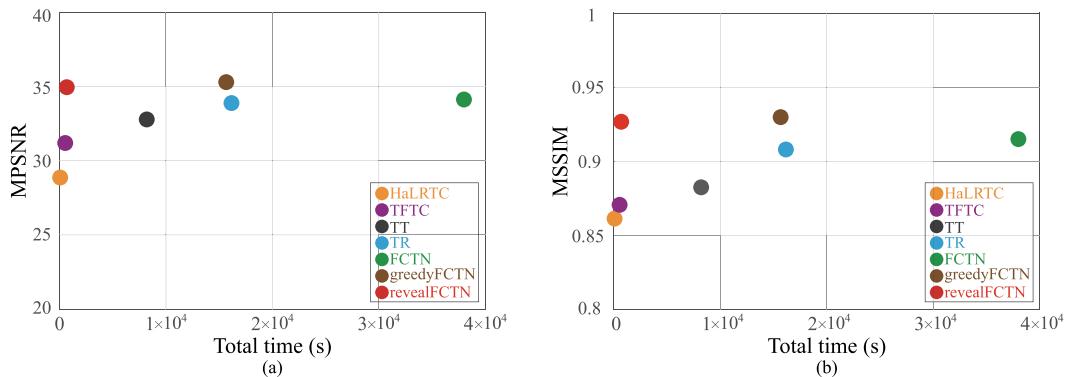
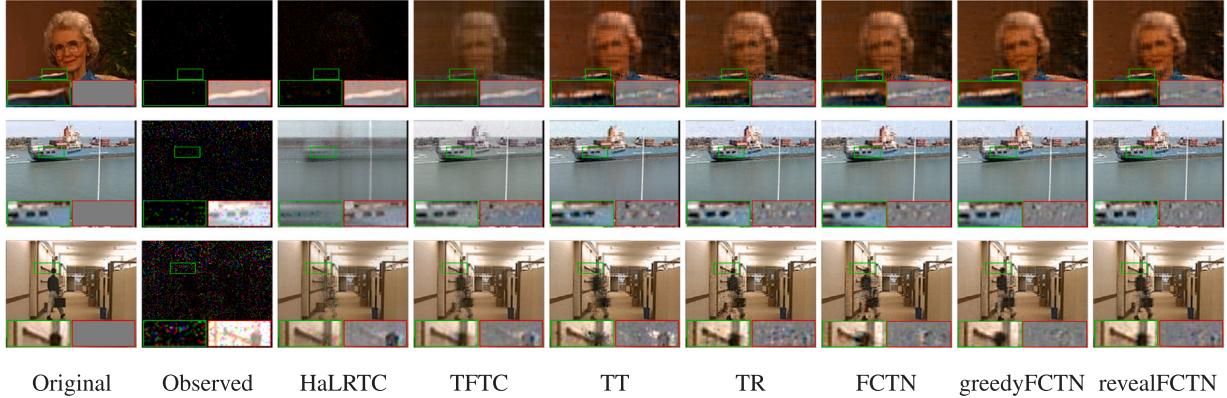


Fig. 5. Comparison of quality metrics and total times of different methods on the fourth-order color video *Grandma* with SR = 5%. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 4**

Quality metrics and total times (seconds) of various methods at different SRs on fourth-order color videos.

Data	Method	SR = 1%			SR = 5%			SR = 10%		
		MPSNR	MSSIM	Total time	MPSNR	MSSIM	Total time	MPSNR	MSSIM	Total time
Grandma	Observed	11.666	0.0147	–	11.846	0.0350	–	12.082	0.0550	–
	HaLRTC	12.438	0.0924	40	28.870	0.8614	43	33.686	0.9315	54
	TFTC	22.623	0.5735	369	31.198	0.8707	664	34.353	0.9267	875
	TT	26.104	0.6552	3209	32.787	0.8825	7946	36.879	0.9513	10168
	TR	24.637	0.5763	5024	34.128	0.9101	16353	37.283	0.9514	24082
	FCTN	25.662	0.6470	7326	34.296	0.9150	37991	37.424	0.9532	48326
Container	greedyFCTN	<b>28.655</b>	<b>0.7312</b>	4719	<b>35.289</b>	<b>0.9303</b>	15660	<b>37.931</b>	<b>0.9587</b>	21266
	revealFCTN	27.890	0.7190	428	34.962	0.9267	776	37.740	0.9571	992
	Observed	4.421	0.0020	–	4.600	0.0067	–	4.835	0.0113	–
	HaLRTC	5.077	0.0270	35	19.054	0.6336	37	28.912	0.9125	51
	TFTC	11.437	0.2133	351	29.588	0.9074	636	33.686	0.9490	771
	TT	21.241	0.6373	2953	28.221	0.8795	7572	31.623	0.9122	9587
Hall	TR	20.858	0.6185	4745	30.314	0.8988	14069	34.759	0.9523	21038
	FCTN	21.017	0.6249	6958	30.762	0.9098	33679	35.215	0.9538	41390
	greedyFCTN	<b>22.716</b>	<b>0.7254</b>	4495	<b>31.935</b>	<b>0.9192</b>	14544	<b>35.921</b>	<b>0.9595</b>	19815
	revealFCTN	22.047	0.7105	391	31.691	0.9138	725	35.778	0.9575	956
	Observed	5.563	0.0027	–	5.742	0.0093	–	5.977	0.0162	–
	HaLRTC	6.232	0.0349	37	19.008	0.6539	41	27.634	0.8887	55
Grandma	TFTC	12.823	0.2739	360	28.885	0.8882	653	31.692	0.9292	818
	TT	21.495	0.6093	3099	27.231	0.8488	7832	30.489	0.9006	9741
	TR	20.723	0.5652	4874	30.210	0.8826	15561	32.379	0.9241	23982
	FCTN	21.078	0.5937	7140	30.451	0.8954	34679	32.840	0.9318	44517
	greedyFCTN	<b>23.409</b>	<b>0.6761</b>	4638	<b>31.403</b>	<b>0.9130</b>	14901	<b>33.732</b>	<b>0.9410</b>	20149
	revealFCTN	22.762	0.6616	414	31.040	0.9005	759	33.551	0.9373	971



**Fig. 6.** Recovered results on fourth-order color videos using different methods (Green box: enlarged image; red box: corresponding residual image). From top to bottom, the sampling rates are 1%, 5%, and 10%, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

### 3.2. Fourth-order color videos

In this subsection, we perform experiments on fourth-order data. Specifically, three color videos<sup>2</sup> (i.e., *Grandma*, *Container*, and *hall*) of size  $144 \times 176 \times 3 \times 50$  (i.e., the first 50 frames of video sequences with each frame of size  $144 \times 176 \times 3$ ), are chosen as test data. For each video, we test three SRs: 1%, 5%, and 10%.

Table 4 lists the MPSNR/MSSIM values and the total times (seconds) for color videos recovered by different methods under different SRs. For a clearer comparison, we illustrate the comparison of quality metrics and total times of different methods on the color video *Grandma* with SR = 5% in Fig. 5. The experimental results demonstrate the superiority of the proposed revealFCTN. In most cases, our method achieves around 1 dB gain in the MPSNR compared to the best-performing constructed TN decomposition, while achieving similar results to that obtained by greedyFCTN in significantly less time, approximately a 10–18× speed-up.

To visually compare the performance of all recovered results, we show one frame of the restructured color videos by different methods under different SRs in Fig. 6. Moreover, we present the enlarged image of the same subregion of each color video in a green box and the corresponding residual image in a red box. From Fig. 6, we observe that our method is promising in recovering the details of moving people or objects in the color videos, especially their brightness and outlines.

### 3.3. Fifth-order light field image

In this subsection, we undertake experiments on a more challenging case, i.e., fifth-order data. Specifically, we consider a light field image<sup>3</sup> (i.e., *Lego Truck*) comprising various views arranged on a  $17 \times 17$  grid. A sub-image of size  $108 \times 162 \times 3 \times 15 \times 4$  is used as test data. For this data, we test six SRs: 1%, 5%, 10%, 20%, 30%, and 40%. Here, we did not conduct experiments with FCTN and revealFCTN due to their prohibitive computational costs. Specifically, FCTN requires adjusting

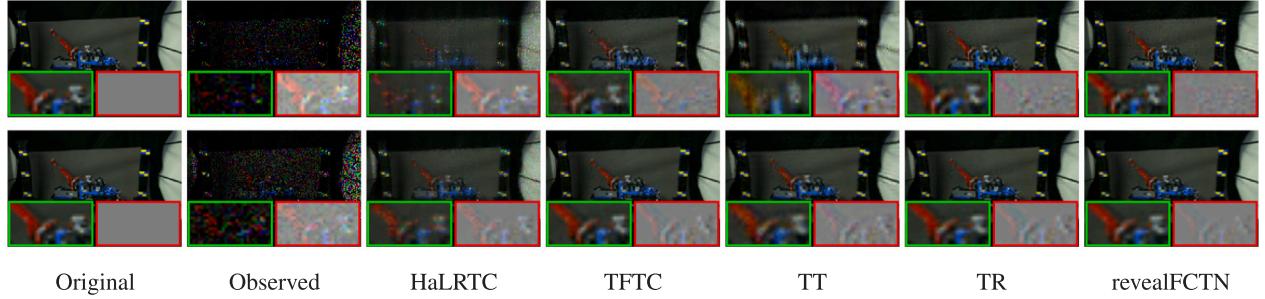
<sup>2</sup> Available at <https://media.xiph.org/video/derf/>.

<sup>3</sup> Available at <http://lightfield.stanford.edu/lfs.html>.

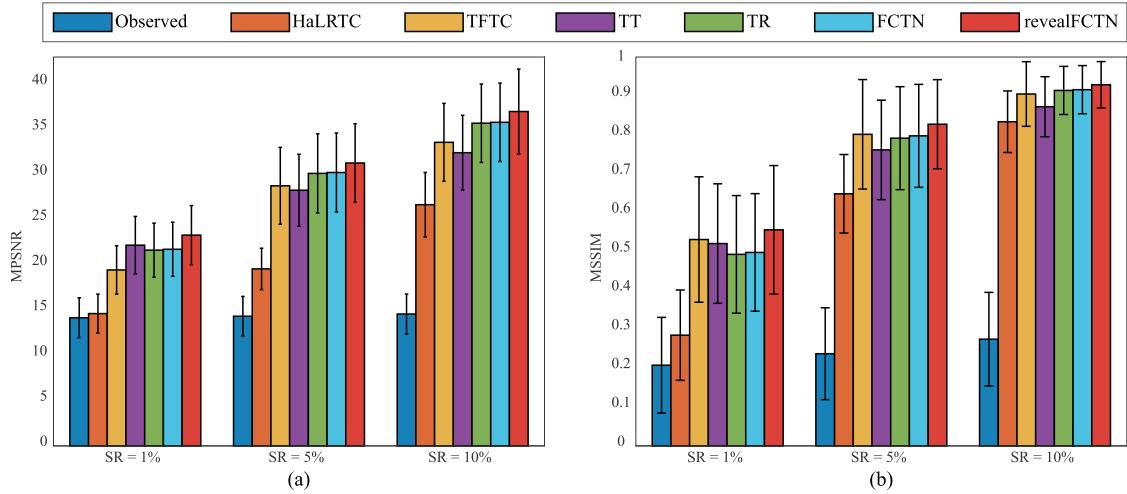
**Table 5**

Quality metrics and total times (seconds) of various methods at different SRs on fifth-order light field image.

SR	Observed		HaLRTC		TFTC		TT		TR		revealFCTN	
	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM
1%	13.430	0.0558	13.471	0.0613	20.424	<b>0.5217</b>	20.627	0.4740	20.788	0.4672	<b>21.231</b>	0.5216
5%	13.611	0.0866	14.065	0.1414	23.664	0.6808	23.183	0.6643	<b>24.244</b>	0.6989	24.016	<b>0.7134</b>
10%	13.846	0.1228	15.206	0.3578	25.634	0.7685	24.467	0.7359	26.343	0.7818	<b>26.878</b>	<b>0.7875</b>
20%	14.360	0.1939	22.343	0.7013	28.737	0.8624	25.643	0.7943	30.194	0.8793	<b>30.900</b>	<b>0.8853</b>
30%	14.935	0.2632	24.517	0.7957	31.386	0.9135	28.060	0.8764	32.454	0.9187	<b>32.694</b>	<b>0.9257</b>
40%	15.602	0.3312	26.461	0.8609	33.881	0.9447	29.432	0.9094	33.914	0.9489	<b>35.422</b>	<b>0.9514</b>
Total time	-		70		943		8938		26 938		2502	



**Fig. 7.** Recovered results on fifth-order light field image using different methods (Green box: enlarged image; red box: corresponding residual image). From top to bottom, the sampling rates are 20% and 40%, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 8.** The statistics of MPSNR/MSSIM values for the third-order CAVE dataset at different SRs.

ten rank parameters, and employing greedyFCTN to search for such a large number of rank parameters is also computationally expensive.

**Table 5** presents the MPSNR/MSSIM values and the total times (seconds) of the light field image recovered by different methods under different SRs. For this more challenging fifth-order data, the numbers of tunable parameters of similar types methods (i.e., TT and TR) are 4 and 5, respectively. As observed, the proposed method still outperforms the compared ones in terms of both MPSNR and MSSIM values with significantly less time.

**Fig. 7** illustrates the restructured light field image by different methods with SR = 20% and SR = 40%. Moreover, we provide the enlarged image of the same subregion of each recovered result in a green box and the corresponding residual image in a red box. From the enlarged subregions, particularly the residual images in the red box, we observe that the proposed method achieves a more accurate approximation to the original image.

#### 4. Discussion

(1) *Statistical significance.* To assess the statistical significance, we evaluate these methods on the CAVE dataset, which includes 32 multi-spectral images. In **Fig. 8**, we report the statistics (mean and standard deviation) of MPSNR and MSSIM values at different SRs. The results show that our method consistently outperforms the compared methods at various SRs.

(2) *revealFCTN vs. FCTN:* As shown in **Table 3** and **Table 4**, our method achieves promising results with a 10-50× speed-up compared to FCTN. For third-order and fourth-order data, FCTN requires tuning three and six parameters (i.e., FCTN rank) respectively, while our method only involves a single threshold parameter. Therefore, the total time required for tuning parameters with our method is significantly lower than that of FCTN. To explore why our method performs better than FCTN, we present the recovered results at different iterations and the corresponding FCTN rank of FCTN and revealFCTN in **Fig. 9**. We

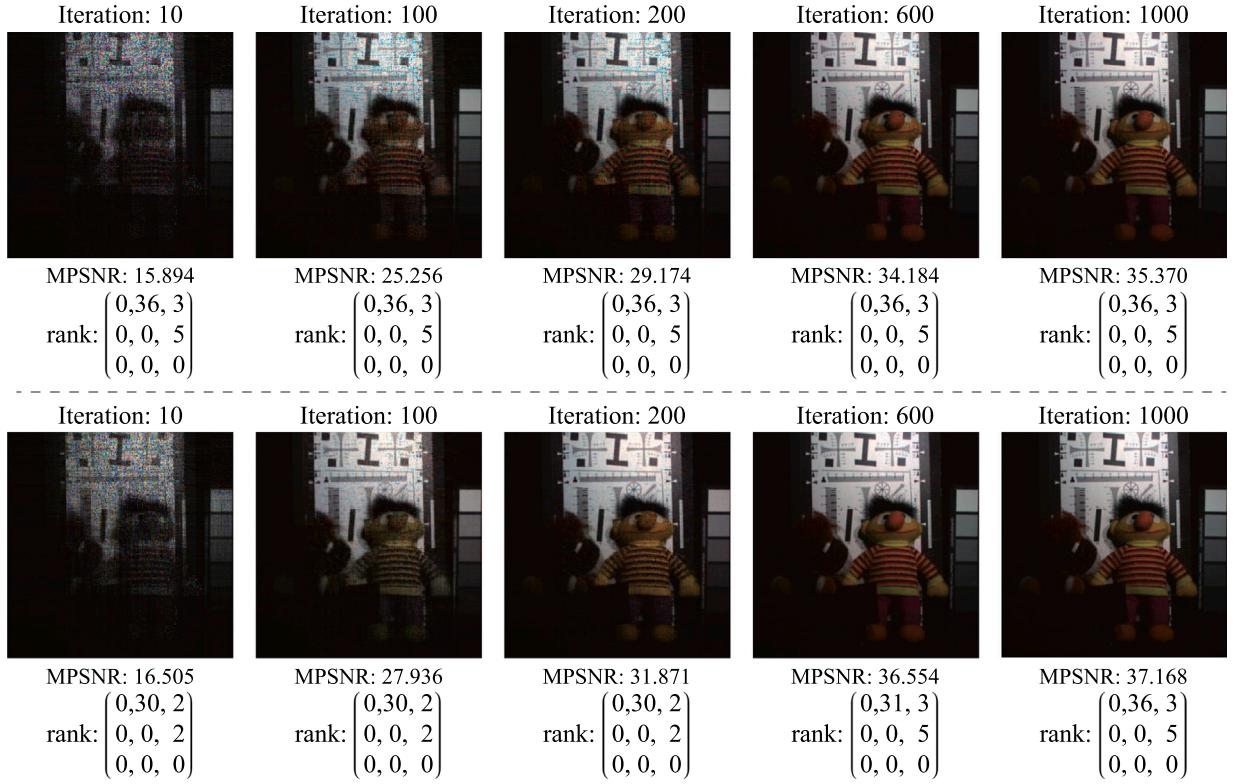


Fig. 9. The recovered results at different iterations and the corresponding FCTN rank of FCTN (the first row) and revealFCTN (the second row) for Toy with SR = 10%, respectively.

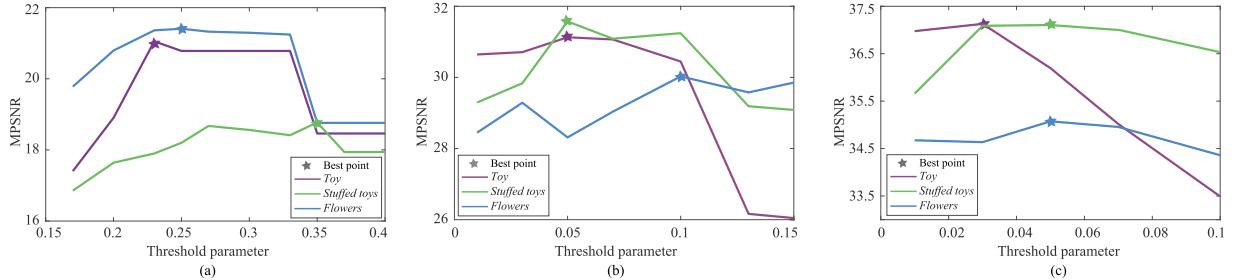


Fig. 10. The MPSNR curves of the recovered results by revealFCTN with respect to the threshold parameter  $\delta$  on multispectral images under different SRs. (a) SR = 1%. (b) SR = 5%. (c) SR = 10%.

observe that our method can flexibly determine an appropriate FCTN rank for the data at each stage. Specifically, the FCTN rank calculated by Algorithm 1 for the initial data is relatively small. The revealFCTN can rapidly capture the rough structural information of the original tensor when the FCTN rank is small. As the image is recovered more accurately, the corresponding FCTN rank computed by Algorithm 1 increases. With the increasing FCTN rank, the proposed revealFCTN can gradually reconstruct local details. Thus, revealFCTN can achieve better recovery results than FCTN.

(3) *Influence of threshold parameter  $\delta$ :* The threshold parameter  $\delta$  in revealFCTN controls the size of the FCTN rank. Using the multispectral images as an example, we illustrate the influence of the threshold parameter  $\delta$  on the recovered results in Fig. 10. As observed, the revealFCTN achieves optimal performance with a suitable threshold parameter. Generally speaking, as the SR increases, the appropriate threshold parameter becomes smaller, indicating that the FCTN rank becomes larger.

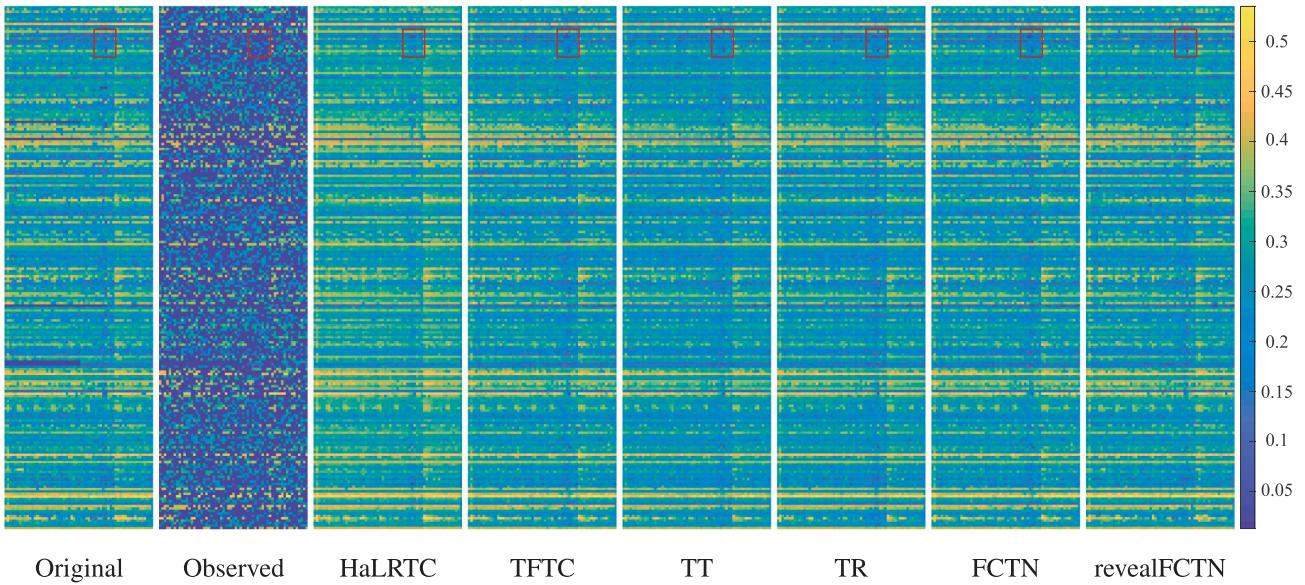
(4) *Extension to other tensor applications.* We extend the proposed method to additional tensor applications, including traffic speed data inpainting and remote sensing image cloud removal.

**Traffic speed data inpainting.** We evaluate our method on an urban traffic speed dataset<sup>4</sup> consisting of 214 road segments within two months at 10-min intervals, forming a tensor of size  $214 \times 61 \times 144$ . The data is normalized to [0,1]. Due to power outages and damaged road sensors, the dataset naturally exhibits tube-wise missing problems, with an original missing rate of 1.3% (visible as blue strips in Fig. 11). To further test our method, we introduced synthetic tube-wise missing data along the third mode, increasing the missing rate to 70%. Fig. 11 presents the 90th frontal slices of the reconstructed traffic speed data using different methods. The results obtained by the proposed revealFCTN are closer to the original data than the compared methods.

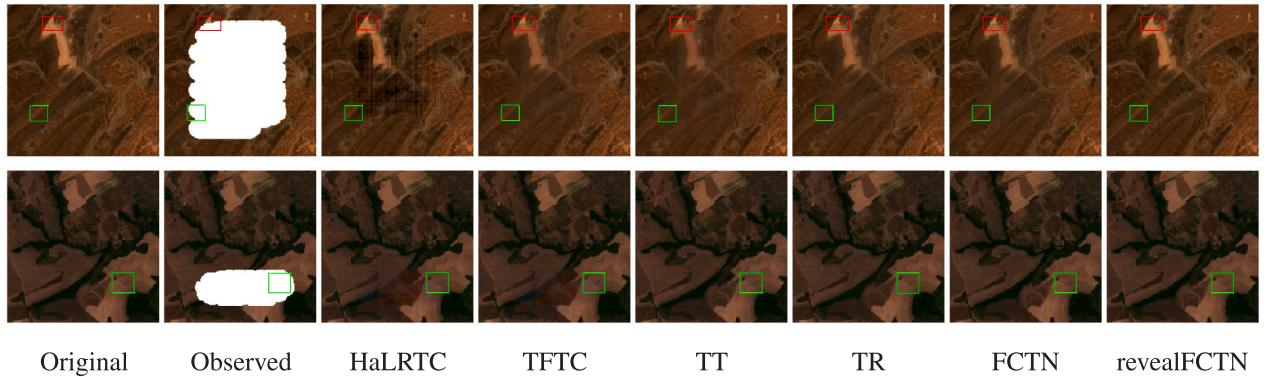
**Remote sensing image cloud removal.** We also test our method on two time-series Sentinel-2 image datasets: Morocco<sup>5</sup> and Brazil<sup>5</sup>. These datasets are resized to  $200 \times 200 \times 4 \times 6$ . The cloud mask is designed as: 1) at different time nodes, clouds are located at different locations and 2) at the same time node, clouds are positioned identically across all spectral bands. Fig. 12 illustrates the pseudo images of the cloud

<sup>4</sup> <https://doi.org/10.5281/zenodo.1205229>.

<sup>5</sup> <https://earthexplorer.usgs.gov/>.



**Fig. 11.** The 90th frontal slice of the recovered traffic speed data using different methods. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 12.** The fourth and fifth time nodes of cloud removal results over Morocco (the first row) and Brazil (the second row) using different methods, respectively.

removal results for Morocco (the first row) and Brazil (the second row) at the fourth and fifth time nodes, respectively. As observed, the results obtained by HaLRTC contain artifacts, while the results obtained by TFTC, TT, TR, and FCTN are over-smoothed and lose local details. In contrast, revealFCTN delivers superior visual results, effectively recovering global structures and preserving local details, especially see the lines and details in the marked boxes.

These experimental results highlight the potential and versatility of the proposed revealFCTN method in effectively addressing real-world applications.

## 5. Conclusions

This paper suggested a novel revealFCTN decomposition, which can efficiently and adaptively determine a suitable FCTN rank for any given data by sufficiently exploiting its correlations. Afterwards, we verify its potential on TC task. We designed a TC model based on the proposed revealFCTN decomposition and utilized a PAM-based algorithm to solve it. Extensive experiments on real data (including third-order, fourth-order, and fifth-order data) substantiated that our method achieves superior results compared to manually constructed TN decompositions in a similar time, while achieving comparable results to that obtained by greedyFCTN in significantly less time.

*Adaptation to other types of data.* Our method can be applied to general tensors, which represent various multidimensional data, such

as videos, remote sensing images, traffic flow data. The proposed revealFCTN adaptively estimates the FCTN rank, requiring only the ranks of single-mode and double-mode unfolding matrices ([Definitions 2](#) and [3](#)) of the data, and then constructs the revealFCTN decomposition. This adaptability makes it practical for handling various real-world data.

*Limitations and future directions.* Although the proposed method does not require any search and evaluation process as the greedy algorithm, estimating the FCTN rank needs to utilize the ranks of unfolding matrices, which still introduces computational costs. To alleviate this problem, random algorithms [[38,39](#)] could be employed to quickly estimate the ranks of the unfolding matrices, thereby further reducing computational costs. Moreover, this paper focuses on handling noise-free data. Noise (e.g., Gaussian or sparse noise) can affect the accuracy of the estimated rank. In future work, we will explore robust rank estimation methods for noisy cases, enhancing the practicality of the proposed method.

## CRediT authorship contribution statement

**Yun-Yang Liu:** Writing – original draft, Visualization. **Xi-Le Zhao:** Supervision, Resources, Methodology, Investigation. **Gemine Vivone:** Writing – review & editing, Supervision.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This research is supported by NSFC (No. 12371456, 12171072, 62131005), Sichuan Science and Technology Program (No. 2024NSFJQ0038, 2023ZYD0007, 2024NSFSC0038), National Key Research and Development Program of China (No. 2020YFA0714001).

## Data availability

Data are publicly available.

## References

- [1] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, *SIAM Rev.* 51 (3) (2009) 455–500.
- [2] M.E. Kilmer, H. Horesh, E. Newman, Tensor-tensor algebra for optimal representation and compression of multiway data, *Proc. Nat. Acad. Sci.* 118 (28) (2021) e2015851118.
- [3] N. Vervliet, O. Debals, L. Sorber, L. De Lathauwer, Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis, *IEEE Signal Process. Mag.* 31 (5) (2014) 71–79.
- [4] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, D.P. Mandic, Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions, *Found. Trends® Mach. Learn.* 9 (4–5) (2016) 249–429.
- [5] H. Wang, J. Peng, W. Qin, J. Wang, D. Meng, Guaranteed tensor recovery fused low-rankness and smoothness, *IEEE Trans. Pattern Anal. Mach. Intell.* (2023).
- [6] Q. Feng, J. Hou, W. Kong, C. Xu, J. Wang, Poisson tensor completion with transformed correlated total variation regularization, *Pattern Recognit.* 156 (2024) 110735.
- [7] W. Qin, H. Wang, F. Zhang, J. Wang, X. Luo, T. Huang, Low-rank high-order tensor completion with applications in visual data, *IEEE Trans. Image Process.* 31 (2022) 2433–2448.
- [8] Y. Liu, F. Shang, W. Fan, J. Cheng, H. Cheng, Generalized higher-order orthogonal iteration for tensor decomposition and completion, *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)* 27 (2014).
- [9] T. Liao, Z. Wu, C. Chen, Z. Zheng, X. Zhang, Tensor completion via convolutional sparse coding with small samples-based training, *Pattern Recognit.* 141 (2023) 109624.
- [10] S. Yuan, K. Huang, A generalizable framework for low-rank tensor completion with numerical priors, *Pattern Recognit.* (2024) 110678.
- [11] W. Wang, Y. Sun, B. Eriksson, W. Wang, V. Aggarwal, Wide compression: Tensor ring nets, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recog., CVPR*, 2018, pp. 9329–9338.
- [12] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, D. Shin, Compression of deep convolutional neural networks for fast and low power mobile applications, in: *Proc. Int. Conf. Learn. Represent., ICLR*, 2016.
- [13] V. Lebedev, Y. Ganin, M. Rakhaba, I. Oseledets, V. Lempitsky, Speeding-up convolutional neural networks using fine-tuned CP-decomposition, in: *Proc. Int. Conf. Learn. Represent., ICLR*, 2015.
- [14] J.A. Bengua, H.N. Phien, H.D. Tuan, M.N. Do, Efficient tensor completion for color image and video recovery: Low-rank tensor train, *IEEE Trans. Image Process.* 26 (5) (2017) 2466–2479.
- [15] L. Xu, L. Cheng, N. Wong, Y.-C. Wu, Tensor train factorization under noisy and incomplete data with automatic rank estimation, *Pattern Recognit.* 141 (2023) 109650.
- [16] I.V. Oseledets, Tensor-train decomposition, *SIAM J. Sci. Comput.* 33 (5) (2011) 2295–2317.
- [17] D. Bigoni, A.P. Engsig-Karup, Y.M. Marzouk, Spectral tensor-train decomposition, *SIAM J. Sci. Comput.* 38 (4) (2016) A2405–A2439.
- [18] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, *Ann. Phys.* 326 (1) (2011) 96–192.
- [19] F. Verstraete, J.I. Cirac, Matrix product states represent ground states faithfully, *Phys. Rev. B* 73 (9) (2006) 094423.
- [20] Q. Zhao, G. Zhou, S. Xie, L. Zhang, A. Cichocki, Tensor ring decomposition, 2016, arXiv preprint arXiv:1606.05535.
- [21] W. Wang, V. Aggarwal, S. Aeron, Efficient low rank tensor ring completion, in: *Proc. IEEE Int. Conf. Comput. Vis., ICCV*, 2017, pp. 5697–5705.
- [22] W. He, Y. Chen, N. Yokoya, C. Li, Q. Zhao, Hyperspectral super-resolution via coupled tensor ring factorization, *Pattern Recognit.* 122 (2022) 108280.
- [23] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, Q. Zhao, T.-X. Jiang, Fully-connected tensor network decomposition and its application to higher-order tensor completion, in: *Proc. AAAI Conf. Artif. Intell., AAAI*, Vol. 35, 2021, pp. 11071–11078.
- [24] C.-Y. Lyu, X.-L. Zhao, B.-Z. Li, H. Zhang, T.-Z. Huang, Multi-dimensional image recovery via fully-connected tensor network decomposition under the learnable transforms, *J. Sci. Comput.* 93 (2) (2022) 49.
- [25] J. Yu, Z. Li, G. Ma, J. Wang, T. Zou, G. Zhou, Low-rank sparse fully-connected tensor network for tensor completion, *Pattern Recognit.* 158 (2025) 111000.
- [26] M. Kodryan, D. Kropotov, D. Vetrov, Mars: Masked automatic ranks selection in tensor decompositions, in: *International Conference on Artificial Intelligence and Statistics, PMLR*, 2023, pp. 3718–3732.
- [27] Y. Liu, J. Chen, Y. Lu, W. Ou, Z. Long, C. Zhu, Adaptively topological tensor network for multi-view subspace clustering, *IEEE Trans. Knowl. Data Eng.* (2024).
- [28] Z. Cheng, B. Li, Y. Fan, Y. Bao, A novel rank selection scheme in tensor ring decomposition based on reinforcement learning for deep neural networks, in: *Proc. IEEE Int. Conf. Acoust. Speech Signal Process., ICASSP*, 2020, pp. 3292–3296.
- [29] M. Hashemizadeh, M. Liu, J. Miller, G. Rabusseau, Adaptive learning of tensor network structures, 2020, arXiv preprint arXiv:2008.05437.
- [30] C. Li, Z. Sun, Evolutionary topology search for tensor network decomposition, in: *Proc. Int. Conf. Mach. Learn., ICML*, 2020, pp. 5947–5957.
- [31] C. Li, J. Zeng, Z. Tao, Q. Zhao, Permutation search of tensor network structures via local sampling, in: *Proc. Int. Conf. Mach. Learn., ICML*, 2022, pp. 13106–13124.
- [32] C. Hawkins, Z. Zhang, Bayesian tensorized neural networks with automatic rank selection, *Neurocomputing* 453 (2021) 172–180.
- [33] K. Hayashi, T. Yamaguchi, Y. Sugawara, S. Maeda, Exploring unexplored tensor network decompositions for convolutional neural networks, *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)* 32 (2019).
- [34] H. Attouch, J. Bolte, B.F. Svaiter, Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods, *Math. Program.* 137 (1) (2013) 91–129.
- [35] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (1) (2013) 208–220.
- [36] P. Zhou, C. Lu, Z. Lin, C. Zhang, Tensor factorization for low-rank tensor completion, *IEEE Trans. Image Process.* 27 (3) (2018) 1152–1163.
- [37] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (4) (2004) 600–612.
- [38] P.-G. Martinsson, V. Rokhlin, M. Tygert, A randomized algorithm for the decomposition of matrices, *Appl. Comput. Harmon. Anal.* 30 (1) (2011) 47–68.
- [39] M. Meier, Y. Nakatsukasa, Fast randomized numerical rank estimation for numerically low-rank matrices, *Linear Algebra Appl.* 686 (2024) 1–32.