Tensor Wheel Decomposition: Theory and Application to Tensor Completion

Zhong-Cheng Wu, Liang-Jian Deng, *Senior Member, IEEE*, Ting-Zhu Huang, *Member, IEEE*, Hong-Xia Dou, Gemine Vivone, *Senior Member, IEEE*, Yu Liu, *Senior Member, IEEE*

***Abstract*—Recently, tensor network (TN) decompositions have gained prominence in computer vision and contributed promising results to tensor recovery for their capability of compactly and efficiently representing high-order tensors. However, current TN topologies are rather being developed towards more intricate structures to pursue incremental improvements, resulting in a drastically increased number of TN ranks, which requires laborious hyper-parameter selection, especially for higher-order cases. In this paper, we propose a novel TN decomposition, dubbed tensor wheel (TW) decomposition, in which a high-order tensor is represented by a set of latent factors mapped into a specific wheel topology. Such a decomposition is constructed starting from analyzing the graph structure, aiming to more accurately characterize the complex interactions inside objectives while maintaining a lower hyper-parameter scale, theoretically alleviating the above deficiencies. The comprehensive analysis of the mathematical properties fully demonstrates that TW decomposition can be more potential in representation capabilities and more flexible in controlling both parameter storage and computational costs. To compute the TW-format decomposition, the sequential singular value decomposition (SVD)-based and the alternating least squares (ALS)-based learning algorithms are developed. Furthermore, to investigate the validity of TW decomposition, we provide its one numerical application, i.e., tensor completion (TC), yet develop an efficient proximal alternating minimization-based solving algorithm with guaranteed convergence. Experimental results on both synthetic and real-world data reveal that TW decomposition significantly outperforms other state-of-the-art tensor decompositions for incomplete-tensor inference, especially under solely few observations, thus substantiating the superiority and reliability of TW decomposition.**

***Index Terms*—Tensor wheel decomposition, tensor network, wheel topology, tensor completion, image inpainting.**

I. Introduction

TENSORS, as a higher-order generalization of matrices, provide the natural representation format for high-dimensional data, e.g., multispectral images (MSIs), color videos (CVs), and light field images (LFIs). Instead of matrices, tensors can inherently preserve the complex interactions inside high-dimensional data, garnering increasing interest in signal processing [1]. Tensor decompositions refer to the representation/approximation of high-order tensor data by using latent small-scale factors, which can be considered as the principal components or features, leading to the reduced storage cost and computational complexity in large-scale and intractable tensor data analysis. Previously, different tensor decompositions have been proposed and extensively applied in various fields, e.g., signal processing [1], [2], [3], [4], [5], [6], [7], machine learning [8], [9], [10], [11], and medical imaging [12]. Among them, tensor singular value decomposition (t-SVD) [13], especially CANDECOMP/PARAFAC (CP) [14] and Tucker [15] decompositions, are the most classical and widely investigated.

More recently, tensor network (TN) decompositions [16], [17], [18], [19] have emerged as a powerful framework to graphically and intuitively construct more sophisticated decomposition formats. They have consecutively been involved in computational mathematics and numerical analysis, exhibiting superior properties, such as super compressing and efficient computing [20], [21]. Particularly, some of TN decompositions have received ubiquitous attention and have productively been introduced to the computer vision community for tensor recovery, e.g., tensor completion (TC) [22], [23], [24], [25] and tensor robust principal component analysis (TRPCA) [26], [27], owing to their excellent characterization capabilities determined by the TN topologies. The popular tensor train (TT) decomposition [28] is one such decomposition, in which an Nth-order tensor is expressed as strictly ordered multi-linear products over one matrix, followed by $N-2$ third-order tensors and another matrix (see Fig. 1(a)). Afterwards, tensor ring (TR) decomposition [29], [30], [31], considered as a generalized form of TT decomposition, instead expands the near-border two matrices in TT factors to two third-order tensors and then constructs the cyclically sequential multi-linear products over N third-order tensor factors (see Fig. 1(b)). Currently, a more sophisticated decomposition, referred fully-connected tensor network (FCTN) decomposition [32], has also been explored by establishing a structure evolved from the complete graph in graph theory, where all decomposed Nth-order factors are connected to each other (see Fig. 1(c)). As of now, TT, TR, and FCTN formats have been properly evaluated and successfully applied to a variety of fields, e.g., classification [33], compressive sensing [34], and image/video recovery [32], [35], [36], [37], [38], showing great performance.

This work was supported in part by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under Grant JYB2025XDXM109, in part by the National Natural Science Foundation of China under Grant 62576132 and Grant U23A20294, and in part by the Project of the Department of Science and Technology of Sichuan Province under Grant 2025YFNH0001. *(Corresponding authors: Liang-Jian Deng; Yu Liu.)*

Z.-C. Wu and Y. Liu are with the School of Instrument Science and Opto-electronics Engineering, Hefei University of Technology, Hefei 230009, China (e-mail: zhongcheng.wu@hfut.edu.cn; yuliu@hfut.edu.cn).

L.-J. Deng and T.-Z. Huang are with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: liangjian.deng@uestc.edu.cn; tingzhuhuang@126.com).

H.-X. Dou is with the School of Science, Xihua University, Chengdu 610039, China (e-mail: hongxia.dou@mail.xhu.edu.cn).

G. Vivone is with the Institute of Methodologies for Environmental Analysis (CNR-IMAA), Tito Scalo 85050, Italy (e-mail: gemine.vivone@imaa.cnr.it).

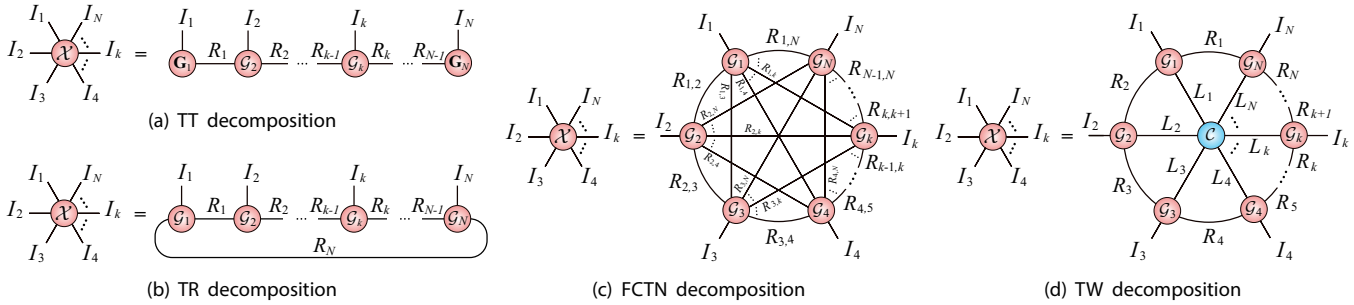


Fig. 1. Graphical illustration of (a) TT decomposition, (b) TR decomposition, (c) FCTN decomposition, and the proposed (d) TW decomposition.

Despite the effectiveness of TT, TR, and FCTN decompositions, three intrinsic limitations are rather reflected in their TN structures. More specifically, (*i*) TT, even TR, models only establish the connection over adjacent latent factors while ignoring other possible interactions between non-adjacent dimensions, leading to an inadequate relation construction. (*ii*) From the perspective of tensor subspace, the intricate features inside an Nth-order tensor can be well-preserved in the same dimensional tensor space without destroying the structures. However, when $N \geq 4$, TT and TR models express an Nth-order tensor employing only third even lower-order factors, which definitely cannot form the desired space. (*iii*) From the number of FCTN-ranks, FCTN model generates $(N^2-N)/2$ hyper-parameters, which scale quadratically with tensor order N. Thus, numerically specifying an optimal collection of FCTN-ranks can be challenging for higher order, making it a costly and tedious task.

Given the above limitations, we argue that a more rational TN model should include three aspects. Namely, (*i*) all factors with dimension modes (i.e., I_k, $k = 1,2,\cdots,N$) are certainly interconnected with their adjacent ones, while remaining as equivalently close as possible to the others. (*ii*) At least one Nth-order factor is required to physically inherit the complex interactions from an Nth-order tensor, thus expecting superior characterization for high-order tensors without sacrificing parameter storage. (*iii*) The TN structure keeps the scale of hyper-parameters lower, i.e., the smaller collection of ranks. Remarkably, the first two folds correspond to the characterization capabilities, which are crucial for TN models in high-order data recovery. Consequently, we propose the tensor wheel (TW) decomposition, which decomposes an Nth-order tensor into both N fourth-order ring factors and an Nth-order core factor, and establishes the multi-linear products by wheel topology (see Fig. 1(d)), aiming to agree above three superior properties.

The main contributions of this paper are as follows.

- We mathematically give several novel TN definitions and investigate the related theoretical properties (Section III-A).
- We analytically propose a novel TN decomposition, called TW decomposition, which decomposes an Nth-order tensor into $N+1$ factors mapped into wheel topology. Such a decomposition allows more expressive characterization for correlation than TT and TR formats, especially preserving the linear scaling for the number of hyper-parameters (TW-ranks) with increased tensor dimension (Section III-B).
- We present two learning algorithms for constructing TW decomposition, i.e., the sequential singular value decomposition (SVD)-based and the alternating least squares (ALS)-based algorithms. Also, we analyze the approximation error bound of the SVD-based algorithm (Section IV).
- We further provide a practical application of TW decomposition, i.e., tensor completion (TC), and then formulate a TW decomposition-based TC model, abbreviated TW-TC. Moreover, we develop an efficient proximal alternating minimization (PAM)-based algorithm to solve the proposed TW-TC model and establish the theoretical proof of the algorithm convergence (Section V).

This paper is an extended version of the earlier conference paper [39]. Since its introduction, TW decomposition has attracted increasing attention in the tensor learning and signal processing communities, and a number of follow-up studies have explored its extensions and applications, e.g., regularized TW models [40], [41], randomized TW algorithms [42], [43], and task-driven TW variants [44], confirming its application-oriented effectiveness. In this context, the journal version aims to provide a more systematic and consolidated treatment of TW decomposition by strengthening its theoretical foundations, clarifying its computational characteristics, and presenting a more comprehensive evaluation of its flexibility, reliability, and universality. Compared with the preliminary version [39], the substantial improvements include

- We investigate the algebraic properties of the generalized tensor k-contraction, which more sufficiently exhibits the computational characteristics (Section III-A).
- We reprove Theorems 1-3 more rigorously by leveraging the above-investigated algebraic properties (Appendix B).
- We present the sequential SVD-based learning algorithm and provide its approximation error analysis (Section IV-A).
- We refine both the parameter storage and computational complexity analysis and clarify the flexibility of TW topology in rank allocation (Remark 1 and Section V-B).
- We perform additional and more comprehensive experiments, including more simulation types, more real-world datasets, more sampling rates, higher-order tensors, storage parameter analysis, topology reliability analysis, and updated comparison methods (Section VI).

The remainder of the paper is organized as follows. The notations and tensor networks are introduced in Section II. The tensor wheel decomposition and theory are developed in Section III. The SVD-based and ALS-based learning algo-

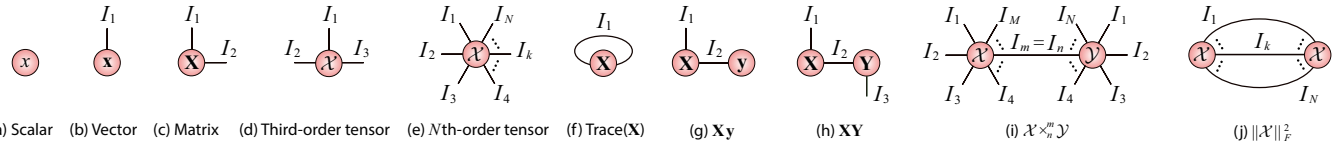


Fig. 2. Tensor network representation of (a) scalar $x \in \mathbb{R}$, (b) vector $\mathbf{x} \in \mathbb{R}^{I_1}$, (c) matrix $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$, (d) third-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, (e) Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, (f) matrix trace $\sum_{i_1} \mathbf{X}(i_1, i_1) \in \mathbb{R}$, (g) matrix-vector product $\mathbf{X}\mathbf{y} \in \mathbb{R}^{I_1}$, (h) matrix-matrix product $\mathbf{X}\mathbf{Y} \in \mathbb{R}^{I_1 \times I_3}$, (i) tensor contraction $\mathcal{X} \times_n^m \mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_{m-1} \times I_{m+1} \times \cdots \times I_M \times J_1 \times \cdots \times J_{n-1} \times J_{n+1} \times \cdots \times J_N}$, and (j) quadratic Frobenius norm $\|\mathcal{X}\|_F^2 \in \mathbb{R}$.

rithms are presented in Section IV. The TW decomposition-based recovery model, algorithm, and analysis are provided in Section V. The numerical experiments are performed in Section VI. Finally, the conclusion is given in Section VII. All proofs are arranged in the supplementary material.

II. Notations and Preliminaries

A. Notations

In general, scalars, vectors, matrices, and tensors are denoted by lowercase letters, e.g., x, lowercase bold letters, e.g., $\mathbf{x}$, uppercase bold letters, e.g., $\mathbf{X}$, and calligraphic letters, e.g., $\mathcal{X}$, respectively. For an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the $(i_1, i_2, \cdots, i_N)$-th element of $\mathcal{X}$ is represented by $\mathcal{X}(i_1, i_2, \cdots, i_N)$. The Frobenius norm of $\mathcal{X}$ is defined as $\|\mathcal{X}\|_F := \sqrt{\sum_{i_1, i_2, \cdots, i_N} \mathcal{X}(i_1, i_2, \cdots, i_N)^2}$. The outer product of vectors $\mathbf{x} \in \mathbb{R}^{I_1}$ and $\mathbf{y} \in \mathbb{R}^{I_2}$ is denoted by $\mathbf{x} \otimes \mathbf{y} \in \mathbb{R}^{I_1 \times I_2}$, whose (i_1, i_2)-th element is $\mathbf{x}(i_1)\mathbf{y}(i_2)$. For integers $p, q \in \mathbb{N}^+$ with $p \leq q$, we especially denote the vector $(p, p+1, \cdots, q)$, the sequence $\{\mathcal{X}_p, \mathcal{X}_{p+1}, \cdots, \mathcal{X}_q\}$, the minimum, and the maximum by $p:q$, $\mathcal{X}_{p:q}$, $p \wedge q$, and $p \vee q$, respectively. Additionally, the top r left singular vectors of any matrix $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ are indicated by $\mathscr{L}_r(\mathbf{X})$.

B. Tensor Networks

Tensor networks graphically consist of vertices and edges, which indicate tensors and modes, respectively. Instead of tensor algebra, TN diagrams can more intuitively represent tensors, tensor contractions, and tensor decomposition models. For tensor representation, the degree (i.e., the number of dangling edges) of vertices corresponds to the order of tensors (see Fig. 2(a)-(e)). The connected edges between two vertices or one vertex with itself represent tensor contractions over the involved tensor modes (see Fig. 2(f)-(j)). When appropriate edges are contracted along vertices, various TN decomposition models, e.g., TT [28], TR [29], and FCTN [32], can be naturally formed (see Fig. 1(a)-(c)). Actually, TN diagrams can be ambiguous for algebraic interpretation by not specifying the arranged order of all tensor modes and not depicting the special tensor modes with a value of one. Nevertheless, any algebraization is acceptable as long as such a criterion is consistent with all computations.

III. Tensor Wheel Decomposition

A. Basic Theory

To numerically construct the TW decomposition, we firstly develop several new tensor definitions and properties.

Definition 1 (Tensor Permutation). *Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Assume that vector $\boldsymbol{n} = (n_1, n_2, \cdots, n_N)$ is a specific reordering of vector $(1, 2, \cdots, N)$, then we define the vector $\boldsymbol{n}$-based permutation of tensor $\mathcal{X}$ as a tensor $\vec{\mathcal{X}}^{\boldsymbol{n}} \in \mathbb{R}^{I_{n_1} \times I_{n_2} \times \cdots \times I_{n_N}}$, whose elements obey*

$$\vec{\mathcal{X}}^{\boldsymbol{n}}(i_{n_1}, i_{n_2}, \cdots, i_{n_N}) = \mathcal{X}(i_1, i_2, \cdots, i_N). \tag{1}$$

Definition 2 (Generalized Tensor k-Unfolding). *Let $\boldsymbol{n} = (n_1, n_2, \cdots, n_N)$ be any reordering of vector $(1, 2, \cdots, N)$, then for an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the generalized k-unfolding $(0 \leq k \leq N,\ k \in \mathbb{Z})$ of $\mathcal{X}$ is defined as a matrix $\mathbf{X}_{[\boldsymbol{n};k]} \in \mathbb{R}^{\prod_{i=1}^{k} I_{n_i} \times \prod_{j=k+1}^{N} I_{n_j}}$, which requires*

$$\mathbf{X}_{[\boldsymbol{n};k]}(\overline{i_{n_1} i_{n_2} \cdots i_{n_k}}, \overline{i_{n_{k+1}} \cdots i_{n_N}}) = \vec{\mathcal{X}}^{\boldsymbol{n}}(i_{n_1}, i_{n_2}, \cdots, i_{n_N}), \tag{2}$$

where $\vec{\mathcal{X}}^{\boldsymbol{n}}$ is the $\boldsymbol{n}$-based tensor permutation of $\mathcal{X}$, and the multi-indices $\overline{i_{n_1} i_{n_2} \cdots i_{n_k}}$ and $\overline{i_{n_{k+1}} i_{n_{k+2}} \cdots i_{n_N}}$ are defined by $1 + \sum_{i=1}^{k} (i_{n_i} - 1) \prod_{j=1}^{i-1} I_{n_j}$ and $1 + \sum_{i=k+1}^{N} (i_{n_i} - 1) \prod_{j=k+1}^{i-1} I_{n_j}$, respectively. When k is 0 and N, $\mathbf{x}_{[\boldsymbol{n};0]} \in \mathbb{R}^{1 \times \prod_{j=1}^{N} I_{n_j}}$ and $\mathbf{x}_{[\boldsymbol{n};N]} \in \mathbb{R}^{\prod_{i=1}^{N} I_{n_i} \times 1}$ imply two generalized vectorizations. Conversely, the inverse operator of the k-unfolding yields $\mathcal{X} = \texttt{Fold}_{[\boldsymbol{n};k]}(\mathbf{X}_{[\boldsymbol{n};k]})$ or $\mathcal{X} = \texttt{Fold}_{[\boldsymbol{n};k]}(\mathbf{x}_{[\boldsymbol{n};k]})$ for $k = 0, N$.

Following Definition 2 with the associated multi-indices, $\mathbf{X}_{[(k,1:k-1,k+1:N);1]}$ can be specialized as $\mathbf{X}_{(k)}$, which is the classical mode-k unfolding, see [3]. Also, $\mathbf{X}_{[(k:N,1:k-1);1]}$ can be specialized as $\mathbf{X}_{<k>}$, which indicates the cyclic mode-k unfolding in TR decomposition [29]. Correspondingly, the mode-k folding operators of these two special unfoldings can be expressed as $\texttt{Fold}_{(k)}(\cdot)$ and $\texttt{Fold}_{<k>}(\cdot)$, respectively.

Definition 3 (Generalized Tensor k-Contraction). *Given an Mth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ and an Nth-order tensor $\mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ with k common modes $(1 \leq k \leq M \wedge N,\ k \in \mathbb{Z})$. Assume that two vectors $\boldsymbol{m} = (m_1, m_2, \cdots, m_M)$ and $\boldsymbol{n} = (n_1, n_2, \cdots, n_N)$ respectively indicate the reordering of vectors $(1, 2, \cdots, M)$ and $(1, 2, \cdots, N)$, satisfying $I_{m_i} = J_{n_i}$ for $i = 1, 2, \cdots, k$, $m_{k+1} < m_{k+2} < \cdots < m_M$ and $n_{k+1} < n_{k+2} < \cdots < n_N$. Then the generalized k-contraction between $\mathcal{X}$ and $\mathcal{Y}$ along the k modes specifies an $(M + N - 2k)$th-order tensor $\mathcal{X} \times_{n_1, \cdots, n_k}^{m_1, \cdots, m_k} \mathcal{Y} \in \mathbb{R}^{I_{m_{k+1}} \times \cdots \times I_{m_M} \times J_{n_{k+1}} \times \cdots \times J_{n_N}}$, which is given as follows,*

$$\mathcal{X} \times_{n_1, n_2, \cdots, n_k}^{m_1, m_2, \cdots, m_k} \mathcal{Y} = \texttt{Fold}_{[(1:M+N-2k);M-k]}\left(\mathbf{X}_{[\boldsymbol{m};k]}^{\top} \mathbf{Y}_{[\boldsymbol{n};k]}\right), \tag{3}$$

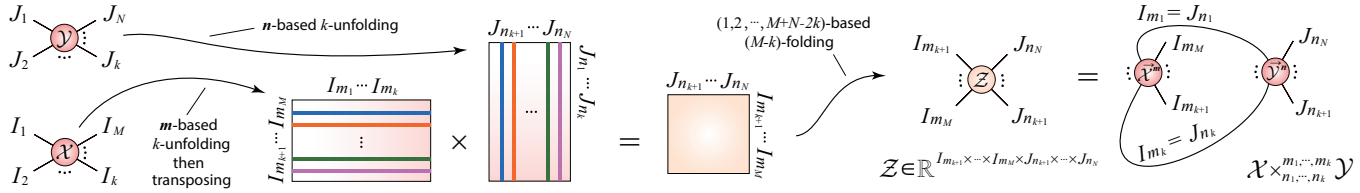


Fig. 3. Graphical illustration of the generalized tensor k-contraction in Definition 3.

where $\mathbf{X}_{[\boldsymbol{m};k]}$ *and* $\mathbf{Y}_{[\boldsymbol{n};k]}$ *are the* $\boldsymbol{m}$*-based and* $\boldsymbol{n}$*-based* k*-unfoldings of* $\mathcal{X}$ *and* $\mathcal{Y}$*, respectively. When* $k = M \wedge N$*, the lowercase symbols, i.e.,* $\mathbf{x}_{[\boldsymbol{m};k]}$ *or* $\mathbf{y}_{[\boldsymbol{n};k]}$*, can be aptly adopted.*

For intuitive understanding, Fig. 3 provides the graphical illustration of the generalized tensor k-contraction. Generally, all contracted modes in Definition 3 are higher than or equal to 2, i.e., $I_{m_i} \geq 2$ and $J_{n_i} \geq 2$, $i = 1, 2, \cdots, k$, but it still works for other exceptional cases. When $\prod_{i=1}^{k} I_{m_i} = \prod_{i=1}^{k} J_{n_i} = 1$, $\prod_{i=k+1}^{M} I_{m_i} \neq 1$, and $\prod_{i=k+1}^{N} J_{n_i} \neq 1$, the matrix-matrix product $\mathbf{X}_{[\boldsymbol{m};k]}^{\top}\mathbf{Y}_{[\boldsymbol{n};k]}$ in formula (3) reduces to the outer product $\mathbf{x}_{[\boldsymbol{m};k]}^{\top} \otimes \mathbf{y}_{[\boldsymbol{n};k]}^{\top}$, appearing as two isolated vertices without the connected edges in TN diagrams. Furthermore, the algebraic properties of the generalized tensor k-contraction are studied for better theoretical analysis.

Property 1. *Let* $\mathcal{X} \times_{n_1,n_2,\cdots,n_k}^{m_1,m_2,\cdots,m_k} \mathcal{Y}$ *be the* k*-contracted tensor in Definition 3, then the* $(i_{m_{k+1}}, \cdots, i_{m_M}, j_{n_{k+1}}, \cdots, j_{n_N})$*-th element of* $\mathcal{X} \times_{n_1,n_2,\cdots,n_k}^{m_1,m_2,\cdots,m_k} \mathcal{Y}$ *can be computed by*

$$\begin{aligned}(\mathcal{X} \times_{n_1,n_2,\cdots,n_k}^{m_1,m_2,\cdots,m_k} \mathcal{Y})(i_{m_{k+1}}, \cdots, i_{m_M}, j_{n_{k+1}}, \cdots, j_{n_N}) \\ = \sum_{i_{m_1}=j_{n_1}=1}^{I_{m_1}} \sum_{i_{m_2}=j_{n_2}=1}^{I_{m_2}} \cdots \sum_{i_{m_k}=j_{n_k}=1}^{I_{m_k}} \\ \big(\mathcal{X}(i_1, i_2, \cdots, i_M)\mathcal{Y}(j_1, j_2, \cdots, j_N)\big).\end{aligned} \tag{4}$$

Property 2. *Given an* M*th-order tensor* $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ *and an* N*th-order tensor* $\mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ *with* k *common modes* ($1 \leq k \leq M \wedge N$, $k \in \mathbb{Z}$)*. Assume that two vectors* $\boldsymbol{m} = (m_1, m_2, \cdots, m_M)$ *and* $\boldsymbol{n} = (n_1, n_2, \cdots, n_N)$ *respectively indicate the reordering of vectors* $(1, 2, \cdots, M)$ *and* $(1, 2, \cdots, N)$*, satisfying* $I_{m_i} = J_{n_i}$ *for* $i = 1, 2, \cdots, k$*,* $m_{k+1} < m_{k+2} < \cdots < m_M$ *and* $n_{k+1} < n_{k+2} < \cdots < n_N$*. Then the computation of the generalized* k*-contraction imposes*

$$\mathcal{X} \times_{n_1,n_2,\cdots,n_k}^{m_1,m_2,\cdots,m_k} \mathcal{Y} = \overrightarrow{(\mathcal{Y} \times_{m_1,m_2,\cdots,m_k}^{n_1,n_2,\cdots,n_k} \mathcal{X})}^{\boldsymbol{v}}, \tag{5}$$

where $\boldsymbol{v} = (N - k + 1 : M + N - 2k, 1 : N - k)$.

Property 2 elaborates that even if the contracted modes are properly specified, the generalized tensor k-contraction operator is still not commutative, i.e., $\mathcal{X} \times_{n_1,\cdots,n_k}^{m_1,\cdots,m_k} \mathcal{Y} \neq \mathcal{Y} \times_{m_1,\cdots,m_k}^{n_1,\cdots,n_k} \mathcal{X}$, just like the matrix multiplication. However, the resulting tensors $\mathcal{X} \times_{n_1,\cdots,n_k}^{m_1,\cdots,m_k} \mathcal{Y}$ and $\mathcal{Y} \times_{m_1,\cdots,m_k}^{n_1,\cdots,n_k} \mathcal{X}$ can be transformed into each other by tensor permutation.

Property 3. *Given an* M*th-order tensor* $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$*, an* N*th-order tensor* $\mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$*, and an* H*th-order tensor* $\mathcal{Z} \in \mathbb{R}^{L_1 \times L_2 \times \cdots \times L_H}$*, where* $\mathcal{X}$ *and* $\mathcal{Y}$*,* $\mathcal{Y}$ *and* $\mathcal{Z}$*, and* $\mathcal{X}$ *and* $\mathcal{Z}$ *have* k_1*,* k_2*, and* k_3 ($2 \leq k_1 + k_3 \leq M$*,* $2 \leq k_1 + k_2 \leq N$*,* $2 \leq k_2 + k_3 \leq H$*,* $k_1, k_2, k_3 \in \mathbb{N}^+$) *common modes, respectively. Assume that three vectors* $\boldsymbol{m} = (m_1, m_2, \cdots, m_M)$*,* $\boldsymbol{n} = (n_1, n_2, \cdots, n_N)$*, and* $\boldsymbol{h} = (h_1, h_2, \cdots, h_H)$ *respectively indicate the reordering of vectors* $(1, 2, \cdots, M)$*,* $(1, 2, \cdots, N)$*, and* $(1, 2, \cdots, H)$*, such that* $I_{m_i} = J_{n_i}$ *for* $i = 1, 2, \cdots, k_1$*,* $J_{n_{k_1+i}} = L_{h_{k_3+i}}$ *for* $i = 1, 2, \cdots, k_2$*,* $I_{m_{k_1+i}} = L_{h_i}$ *for* $i = 1, 2, \cdots, k_3$*,* $m_{k_1+k_3+1} < m_{k_1+k_3+2} < \cdots < m_M$*,* $n_{k_1+k_2+1} < n_{k_1+k_2+2} < \cdots < n_N$*, and* $h_{k_2+k_3+1} < h_{k_2+k_3+2} < \cdots < h_H$*. When the contracted modes are fully and properly specified, the successively contracted tensors with different precedences are equivalent as follows,*

$$\begin{aligned}&\mathcal{X} \times_{1,2,\cdots,k_1,N-k_2+1,\cdots,N-k_2+k_3}^{m_1,m_2,\cdots,m_{k_1},m_{k_1+1},\cdots,m_{k_1+k_3}} \big(\mathcal{Y} \times_{h_{k_3+1},\cdots,h_{k_3+k_2}}^{n_{k_1+1},\cdots,n_{k_1+k_2}} \mathcal{Z}\big) \\ &= \big(\mathcal{X} \times_{n_1,n_2,\cdots,n_{k_1}}^{m_1,m_2,\cdots,m_{k_1}} \mathcal{Y}\big) \times_{h_1,h_2,\cdots,h_{k_3},h_{k_3+1},\cdots,h_{k_3+k_2}}^{1,2,\cdots,k_3,M-k_1+1,\cdots,M-k_1+k_2} \mathcal{Z}.\end{aligned} \tag{6}$$

Property 3 demonstrates the associative property of the generalized tensor k-contraction operator, provided that the contracted modes are properly specified. Otherwise, such an operator is performed by only the left-side precedence.

B. Tensor Wheel Model

Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, then TW decomposition aims to parameterize it by both N fourth-order ring factors $\mathcal{G}_k \in \mathbb{R}^{R_k \times I_k \times L_k \times R_{k+1}}$, $k = 1, 2, \cdots, N$, and an Nth-order core factor $\mathcal{C} \in \mathbb{R}^{L_1 \times L_2 \times \cdots \times L_k \times \cdots \times L_N}$. When $k = N$, R_{N+1} stands for R_1. Mathematically, the TW model establishes the element-wise relation as follows,

$$\begin{aligned}\mathcal{X}(i_1, i_2, \cdots, i_N) = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_N=1}^{R_N} \sum_{l_1=1}^{L_1} \sum_{l_2=1}^{L_2} \cdots \sum_{l_N=1}^{L_N} \\ \big(\mathcal{G}_1(r_1, i_1, l_1, r_2)\mathcal{G}_2(r_2, i_2, l_2, r_3) \\ \cdots \mathcal{G}_k(r_k, i_k, l_k, r_{k+1}) \cdots \\ \mathcal{G}_N(r_N, i_N, l_N, r_1)\mathcal{C}(l_1, l_2, \cdots, l_N)\big).\end{aligned} \tag{7}$$

According to Property 1, when the tensor contractions are firstly executed on the ring factors $\mathcal{G}_k$, $k = 1, 2, \cdots, N$, TW model (7) can be naturally expressed in the tensor form[1] by

$$\mathcal{X} = \mathcal{G}_1 \times_1^4 \mathcal{G}_2 \times_1^6 \cdots \times_1^{2k} \mathcal{G}_k \times_1^{2k+2} \cdots \times_{1,4}^{2N,1} \mathcal{G}_N \times_{1,2,\cdots,N}^{2,4,\cdots,2N} \mathcal{C}, \tag{8}$$

which provides the actual computation for TW decomposition. Symbolically, we employ $\mathrm{TW}[\![\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_N; \mathcal{C}]\!]$ or, more compactly, $\mathrm{TW}[\![\{\mathcal{G}_k\}_{k=1}^{N}; \mathcal{C}]\!]$ to denote TW decomposition. Moreover, the ring factors $\mathcal{G}_k$, $k = 1, 2, \cdots, N$, and the core

[1]For model (7), Property 1 determines that $\mathcal{G}_k$, $k = 1, 2, \cdots, N$, can only be sequentially contracted from $\mathcal{G}_1$ to $\mathcal{G}_N$, but is unconstrained on $\mathcal{C}$. Accordingly, when $\mathcal{G}_k$, $k = 1, 2, \cdots, N$, are firstly contracted, the tensor form of model (7) can be uniquely established in formula (8).

factor $\mathcal{C}$ are collectively called *TW factors*. Apart from the basic tensor form in formula (8), TW decomposition also has numerous generalized tensor forms, but enumerating all the cases is laborious and pointless. Alternatively, Theorems 1 and 2 exhibit two special tensor forms, aiming to illustrate the invariance property of TW decomposition.

Theorem 1 (**Core-Centered Circular Invariance**). *Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and its TW decomposition TW$[\![\{\mathcal{G}_k\}_{k=1}^N; \mathcal{C}]\!]$. Assume that $\boldsymbol{n} = (n_1, n_2, \cdots, n_N)$ is the circular reordering of vector $(1, 2, \cdots, N)$, then the core-centered invariance gives $\vec{\mathcal{X}}^{\boldsymbol{n}} =$ TW$[\![\{\mathcal{G}_{n_k}\}_{k=1}^N; \vec{\mathcal{C}}^{\boldsymbol{n}}]\!]$.*

Theorem 2 (**Core-Connected Invariance**). *Assume that the generalized TW decomposition of $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is $\vec{\mathcal{X}}^{\boldsymbol{n}} =$ TW$[\![\{\mathcal{G}_{n_k}\}_{k=1}^N; \vec{\mathcal{C}}^{\boldsymbol{n}}]\!]$, where $\boldsymbol{n} = (n_1, n_2, \cdots, n_N)$ is any vector that circularly shifts vector $(1, 2, \cdots, N)$. Let vector $\boldsymbol{e} = (n_1, n_k, n_2, \cdots, n_{k-1}, n_{k+1}, \cdots, n_N)$ $(3 \le k < N,\ k \in \mathbb{Z})$, then*

$$\vec{\mathcal{X}}^{\boldsymbol{e}} = \mathcal{G}_{n_1} \times_1^3 \vec{\mathcal{C}}^{\boldsymbol{e}} \times_3^4 \mathcal{G}_{n_k} \times_{1,3,\cdots,2k-3,2k-2}^{3,4,\cdots,k+1,N+2} \mathcal{G}_{n_2,\cdots,n_{k-1}} \times_{2(N-k)+2,3,5,\cdots,2(N-k)+1,1}^{1,3,4,\cdots,N-k+2,N-k+4} \mathcal{G}_{n_{k+1},\cdots,n_N}, \tag{9}$$

where $\mathcal{G}_{n_2,\cdots,n_{k-1}} = \mathcal{G}_{n_2} \times_1^4 \cdots \times_1^{2(k-2)} \mathcal{G}_{n_{k-1}}$ and $\mathcal{G}_{n_{k+1},\cdots,n_N} = \mathcal{G}_{n_{k+1}} \times_1^4 \cdots \times_1^{2(N-k)} \mathcal{G}_{n_N}$.

Theorem 2 reveals that any two non-adjacent ring factors with dimension modes (i.e., I_k, $k = 1, 2, \cdots, N$) can be connected by the core factor. Accordingly, TW topology can comprehensively establish all possible mode interactions of a high-order tensor without being impeded by other ring factors, which endows TW decomposition with adequate relation construction over TT and TR decompositions.

Theorem 3 (**Tensor Subwheel Equation**). *Assume that the generalized TW decomposition of $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is $\vec{\mathcal{X}}^{\boldsymbol{n}} =$ TW$[\![\{\mathcal{G}_{n_k}\}_{k=1}^N; \vec{\mathcal{C}}^{\boldsymbol{n}}]\!]$, where $\boldsymbol{n} = (n_1, n_2, \cdots, n_N)$ is any vector that circularly shifts vector $(1, 2, \cdots, N)$. Let vectors $\boldsymbol{u} = (N+1, N+2, 1, 2, \cdots, N)$ and $\boldsymbol{v} = (2, 4, \cdots, 2N, 1, 3, \cdots, 2N-1)$, then there inherently exists the following two tensor subwheel equations,*

$$\mathbf{X}_{<n_N>} = (\mathbf{G}_{n_N})_{(2)} (\mathbf{M}_{\neq n_N})_{[\boldsymbol{u};3]} \tag{10}$$

and

$$\vec{\mathbf{x}}^{\boldsymbol{n}}_{[(1:N);0]} = \vec{\mathbf{c}}^{\boldsymbol{n}}_{[(1:N);0]} (\mathbf{N}_{\neq \mathcal{C}})_{[\boldsymbol{v};N]}, \tag{11}$$

where $\mathcal{M}_{\neq n_N} \in \mathbb{R}^{R_{n_1} \times I_{n_1} \times \cdots \times I_{n_{N-1}} \times R_{n_N} \times L_{n_N}}$ is an $(N+2)$th-order subwheel tensor, which merges all TW factors but $\mathcal{G}_{n_N}$, i.e., $\mathcal{M}_{\neq n_N} = \mathcal{G}_{n_1} \times_1^4 \cdots \times_1^{2k} \mathcal{G}_{n_k} \times_1^{2k+2} \cdots \times_1^{2N-2} \mathcal{G}_{n_{N-1}} \times_{1,2,\cdots,N-1}^{3,5,\cdots,2N-1} \vec{\mathcal{C}}^{\boldsymbol{n}}$, and $\mathcal{N}_{\neq \mathcal{C}} \in \mathbb{R}^{I_{n_1} \times L_{n_1} \times \cdots \times I_{n_N} \times L_{n_N}}$ is another $2N$th-order subwheel tensor obtained by only merging $\{\mathcal{G}_{n_k}\}_{k=1}^N$, i.e., $\mathcal{N}_{\neq \mathcal{C}} = \mathcal{G}_{n_1} \times_1^4 \cdots \times_1^{2k} \mathcal{G}_{n_k} \times_1^{2k+2} \cdots \times_{1,4}^{2N,1} \mathcal{G}_{n_N}$.

Theorem 3 clarifies that all TW factors, i.e., $\mathcal{G}_k$, $k = 1, 2, \cdots, N$, and $\mathcal{C}$, can be individually separated from TW$[\![\{\mathcal{G}_k\}_{k=1}^N; \mathcal{C}]\!]$, which contributes to those alternating direction-based numerical algorithms, e.g., ALS and PAM ones in Sections IV-B and V-A, respectively. Furthermore, $\vec{\mathbf{x}}^{\boldsymbol{n}}_{[(1:N);0]} = \vec{\mathbf{c}}^{\boldsymbol{n}}_{[(1:N);0]} (\mathbf{N}_{\neq \mathcal{C}})_{[\boldsymbol{v};N]}$ in formula (11) is essentially $\vec{\mathcal{X}}^{\boldsymbol{n}} = \vec{\mathcal{C}}^{\boldsymbol{n}} \times_{2,4,\cdots,2N}^{1,2,\cdots,N} \mathcal{N}_{\neq \mathcal{C}}$, appearing as a tensor subspace representation. Since all modes of $\vec{\mathcal{X}}^{\boldsymbol{n}}$ are contained in $\mathcal{N}_{\neq \mathcal{C}}$, $\vec{\mathcal{C}}^{\boldsymbol{n}}$ is purely a coefficient tensor, and $\mathcal{N}_{\neq \mathcal{C}}$ can be regarded as a basis tensor. From such a perspective, the intrinsic structure of a high-order tensor can be well-preserved by the core factor $\mathcal{C}$, leading to a high characterization ability of TW decomposition.

Theorem 4. *Assume that $\mathcal{X} =$ TW$[\![\{\mathcal{G}_k\}_{k=1}^N; \mathcal{C}]\!]$ with N ring factors $\mathcal{G}_k \in \mathbb{R}^{R_k \times I_k \times L_k \times R_{k+1}}$, then for $k = 1, 2, \cdots, N$,*

$$\texttt{Rank}(\mathbf{X}_{(k)}) = \texttt{Rank}(\mathbf{X}_{<k>}) \le L_k \prod_{i=k}^{k+1} R_i. \tag{12}$$

Theorem 4 justifies that R_k and L_k, $k = 1, 2, \cdots, N$, can upper bound the rank of all dimension-mode tensor unfolding and control the low-rankness behavior to a certain extent. Thus, they are assigned as the *TW-ranks*, usually collected by a vector $\boldsymbol{r} = (R_1, R_2, \cdots, R_N, L_1, L_2, \cdots, L_N) \in \mathbb{R}^{2N}$. The TW-ranks essentially determines the actual TN structure, e.g., Tucker topology when $R_k = 1$ and TR topology when $L_k = 1$ with $k = 1, 2, \cdots, N$, graphically proving the superiority of TW decomposition over both Tucker and TR decompositions[2]. Along such a perspective, TT, TR, and FCTN decompositions can be viewed as a special case of the adaptive TN (ATN) decomposition [19], which adaptively identifies TN structures by eliminating the relevant edges from the complete graph topology. Nevertheless, provided that decomposing the same Nth-order tensor, TW decomposition is exactly excluded from the setting of ATN decomposition since the latter cannot fit the structures with an internal core tensor, e.g., Tucker [15], hierarchical Tucker [17], and projected entangled state pairs (PEPS) [16] topologies. Compared with FCTN decomposition with $(N^2 - N)/2$ ranks, TW decomposition forms the collection of $2N$ ranks, which grows only linearly rather than quadratically with the tensor order N, thereby improving its flexibility for higher-order tensor recovery applications.

Remark 1 (**Storage Complexity Analysis**). *For a given Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times \cdots \times I}$, whose TW-ranks $\boldsymbol{r}$ is assumed to involve only R and L, i.e., $\boldsymbol{r} = (R, R, \cdots, R, L, L, \cdots, L)$, TW decomposition requires $\mathcal{O}(NIR^2L + L^N)$ parameter costs for storage. Apparently, the $\mathcal{O}(NIR^2L + L^N)$ scales exponentially with the tensor order N, resulting in a cursed storage complexity, just like Tucker decomposition. Nonetheless, TW topology considers the potential relationship and further establishes valid connections between the adjacent ring factors than Tucker topology. Such a strategy not only allows TW decomposition for more expressive characterization capability but also reduces the loadings of the core factor, which contributes the smaller L in $\mathcal{O}(NIR^2L + L^N)$, thus alleviating the burden of higher storage complexity. For instance, when the SVD-based learning algorithm is applied to decompose an Nth-order tensor, the $\{L_k\}_{k=1}^N$ in $\boldsymbol{r}$ can be arbitrarily specified (e.g., $L_k = 1$, $k = 1, 2, \cdots, N$) without any approximation error (Section IV-A), which is unachievable for Tucker decomposition.*

[2]Instead of the code of the conference paper [39], please see https://github.com/zhongchengwu/code_TWDec, the journal version can actually adjust all values of TW-ranks to 1 in MATLAB code.

IV. Learning Algorithms

To acquire the TW-format representation or approximation, we develop two corresponding learning algorithms, including a sequential singular value decomposition (SVD)-based one and an alternating least squares (ALS)-based one, called TW-SVD algorithm and TW-ALS algorithm, respectively.

A. TW-SVD Algorithm

Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and its predefined TW-ranks, then the TW-SVD algorithm computes the TW factors by N sequential singular value decompositions[3].

More specifically, the tensor contraction-based representation of TW decomposition in formula (8) is

$$\mathcal{X} = \mathcal{G}_1 \times_1^4 \mathcal{G}_2 \times_1^6 \cdots \times_{1,4}^{2N,1} \mathcal{G}_N \times_{1,2,\cdots,N}^{2,4,\cdots,2N} \mathcal{C}. \tag{13}$$

According to Property 3, when the contracted tensor modes are suitably specified, model (13) can be reformatted by

$$\mathcal{X} = \mathcal{G}_1 \times_{N+1,N+2,1}^{1,3,4} \mathcal{M}_{\neq 1}, \tag{14}$$

where $\mathcal{M}_{\neq 1} = \mathcal{G}_2 \times_1^4 \mathcal{G}_3 \times_1^6 \cdots \times_1^{2N-2} \mathcal{G}_N \times_{2,3,\cdots,N}^{3,5,\cdots,2N-1} \mathcal{C}$. Likewisely, the subwheel $\mathcal{M}_{\neq 1}$ can also be reformatted and further unified into the recursive equation as follows,

$$\mathcal{M}_{\neq(1:k-1)} = \mathcal{G}_k \times_{N+2,1}^{3,4} \mathcal{M}_{\neq(1:k)}, \ k = 2, 3, \cdots, N-1, \tag{15}$$

where $\mathcal{M}_{\neq(1:N-1)} = \mathcal{G}_N \times_N^3 \mathcal{C}$. Following Definition 3, the contraction-based (14)-(15) can be equivalently rewritten as

$$\mathbf{X}_{<1>} = (\mathbf{G}_1)_{(2)} (\mathbf{M}_{\neq 1})_{[(N+1,N+2,1:N);3]}, \tag{16}$$

and

$$\left(\mathbf{M}_{\neq(1:k-1)}\right)_{[(1:N+2);2]} = (\mathbf{G}_k)_{[(1:4);2]} \left(\mathbf{M}_{\neq(1:k)}\right)_{[z;2]}, \tag{17}$$

where

$$\left(\mathbf{M}_{\neq(1:N-1)}\right)_{[(1:N+2);3]} = (\mathbf{G}_N)_{(3)}^\top \mathbf{C}_{(N)}, \tag{18}$$

and vector $\boldsymbol{z} = (N+2, 1:N+1)$. Let the compact SVD of $\mathbf{X}_{<1>}$ in (16) be

$$\mathbf{X}_{<1>} = \begin{bmatrix} \mathbf{U} & \mathbf{U}_\epsilon \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma} & \\ & \mathbf{\Sigma}_\epsilon \end{bmatrix} \begin{bmatrix} \mathbf{V}^\top \\ \mathbf{V}_\epsilon^\top \end{bmatrix}, \tag{19}$$

where $\mathbf{U} = \mathscr{L}_{R_1 L_1 R_2}(\mathbf{X}_{<1>})$, then the $\mathbf{U}$ and $\mathbf{\Sigma}\mathbf{V}^\top$ can be designated as the $(\mathbf{G}_1)_{(2)}$ and $(\mathbf{M}_{\neq 1})_{[(N+1,N+2,1:N);3]}$, respectively. Afterwards, by sequentially applying the truncated SVD to $\left(\mathbf{M}_{\neq(1:k-1)}\right)_{[(1:N+2);2]}$, $k = 2, 3, \cdots, N-1$, and $\left(\mathbf{M}_{\neq(1:N-1)}\right)_{[(1:N+2);3]}$, the $(\mathbf{G}_k)_{[(1:4);2]}$, $(\mathbf{G}_N)_{(3)}$, and $\mathbf{C}_{(N)}$ can be obtained. Ultimately, the TW factors $\mathcal{G}_k$, $k = 1, 2, \cdots, N$, and $\mathcal{C}$, can be formed by the folding operators. The TW-SVD algorithm is summarized in Algorithm 1.

Remark 2. *From the requirements of the truncated SVDs, the TW-ranks $\boldsymbol{r}$ is upper bounded, i.e., $R_1 L_1 R_2 \leq I_1 \wedge \prod_{i=2}^N I_i$, $L_k R_{k+1} \leq R_k I_k \wedge \prod_{i=k+1}^N I_i R_1 \prod_{j=1}^{k-1} L_j$, $k = 2, 3, \cdots, N-1$, and $L_N \leq R_N I_N R_1 \wedge \prod_{j=1}^{N-1} L_j$. Interestingly, Proposition 1 implies that when the given R_k and L_k, $k = 1, 2, \cdots, N$, violate the above inequalities, TW decomposition can still be indirectly generated by the TW-SVD algorithm by adopting the smaller $\bar{R}_k$ and $\bar{L}_k$, where $\bar{R}_k = R_k - p_k$, $\bar{L}_k = L_k - q_k$, and $p_k, q_k \in \mathbb{N}$. Therefore, the TW-ranks $\boldsymbol{r}$ in Algorithm 1 can be freely given as long as $R_k \geq 1$ and $L_k \geq 1$.*

[3]Typically, the sequence is closely related to the tensor contraction-based representation of TW decomposition (e.g., that of Theorems 1-2). For different representations, the TW-SVD algorithm can generate TW factors following different sequences. Here, we design to sequentially yield $\mathcal{G}_1$, $\mathcal{G}_2$, $\cdots$, $\mathcal{G}_N$, and $\mathcal{C}$. The other cases can be similarly deduced.

Proposition 1. *Let TW$[\![\{\mathcal{G}_k\}_{k=1}^N; \mathcal{C}]\!]$ be the TW decomposition of tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, where $\mathcal{G}_k \in \mathbb{R}^{R_k \times I_k \times L_k \times R_{k+1}}$, $k = 1, 2, \cdots, N$, and $\mathcal{C} \in \mathbb{R}^{L_1 \times L_2 \times \cdots \times L_k \times \cdots \times L_N}$. Assume that for $\forall k$, $\bar{R}_k = R_k + p_k$ and $\bar{L}_k = L_k + q_k$, where $p_k, q_k \in \mathbb{N}$, and denote the factors $\mathcal{G}_k$, $\mathcal{C}$, $\bar{\mathcal{G}}_k$, and $\bar{\mathcal{C}}$ by $\bar{\mathcal{G}}_k^{(0)} \in \mathbb{R}^{R_k \times I_k \times L_k \times R_{k+1}}$, $\bar{\mathcal{C}}^{(0)} \in \mathbb{R}^{L_1 \times L_2 \times \cdots \times L_k \times \cdots \times L_N}$, $\bar{\mathcal{G}}_k^{(4)} \in \mathbb{R}^{\bar{R}_k \times I_k \times \bar{L}_k \times \bar{R}_{k+1}}$, and $\bar{\mathcal{C}}^{(N)} \in \mathbb{R}^{\bar{L}_1 \times \bar{L}_2 \times \cdots \times \bar{L}_k \times \cdots \times \bar{L}_N}$. When $\bar{\mathcal{G}}_k^{(i)}$, $i = 1, 2, 3, 4$, and $\bar{\mathcal{C}}^{(j)}$, $j = 1, 2, \cdots, N$, are recursively defined by*

$$\left(\bar{\mathbf{G}}_k^{(i)}\right)_{(i)} = \begin{bmatrix} \left(\bar{\mathbf{G}}_k^{(i-1)}\right)_{(i)} \\ \mathbf{0} \end{bmatrix}, \ k = 1, 2, \cdots, N, \tag{20}$$

and

$$\left(\bar{\mathbf{C}}^{(j)}\right)_{(j)} = \begin{bmatrix} \left(\bar{\mathbf{C}}^{(j-1)}\right)_{(j)} \\ \mathbf{0} \end{bmatrix}, \tag{21}$$

where $\mathbf{0}$ implies the all-zeros matrix with appropriate size, the factors $\bar{\mathcal{G}}_k$, $k = 1, 2, \cdots, N$, and $\bar{\mathcal{C}}$ satisfy $\mathcal{X} =$ TW$[\![\{\bar{\mathcal{G}}_k\}_{k=1}^N; \bar{\mathcal{C}}]\!]$.

Theorem 5 (Approximation Error). *Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and its predefined TW-ranks. Assume that the factors $\mathcal{G}_k \in \mathbb{R}^{R_k \times I_k \times L_k \times R_{k+1}}$, $k = 1, 2, \cdots, N$, and $\mathcal{C} \in \mathbb{R}^{L_1 \times L_2 \times \cdots \times L_k \times \cdots \times L_N}$ are generated by Algorithm 1. Then, the TW-format tensor, i.e., TW$[\![\{\mathcal{G}_k\}_{k=1}^N; \mathcal{C}]\!]$, satisfies*

$$\begin{aligned} \left\| \mathcal{X} - \text{TW}[\![\{\mathcal{G}_k\}_{k=1}^N; \mathcal{C}]\!] \right\|_F \leq & \sqrt{\sum_{i=R_1 L_1 R_2 + 1}^{I_1 \wedge \prod_{i=2}^N I_i} \sigma_i^2 \left(\mathbf{X}_{<1>}\right)} \\ & + \sum_{k=2}^{N-1} \sqrt{\sum_{i=L_k R_{k+1}+1}^{R_k I_k \wedge \prod_{i=k+1}^N I_i R_1 \prod_{j=1}^{k-1} L_j} \sigma_i^2 \left(\left(\mathbf{M}_{\neq(1:k-1)}\right)_{[(1:N+2);2]} \right)} \\ & + \sqrt{\sum_{i=L_N+1}^{R_N I_N R_1 \wedge \prod_{j=1}^{N-1} L_j} \sigma_i^2 \left(\left(\mathbf{M}_{\neq(1:N-1)}\right)_{[(1:N+2);3]} \right)}, \end{aligned}$$

where $\mathcal{M}_{\neq 1} = \texttt{Fold}_{[(N+1,N+2,1:N);3]}((\mathbf{G}_1)_{(2)}^\top \mathbf{X}_{<1>})$, $\mathcal{M}_{\neq(1:k)} = \texttt{Fold}_{[z;2]}((\mathbf{G}_k)_{[(1:4);2]}^\top \left(\mathbf{M}_{\neq(1:k-1)}\right)_{[(1:N+2);2]})$, $k = 2, 3, \cdots, N-1$, and $\sigma_i(\cdot)$ indicates the i-th singular value.

By Theorem 5, provided that $\|\mathcal{X} - \text{TW}[\![\{\mathcal{G}_k\}_{k=1}^N; \mathcal{C}]\!]\|_F = 0$, the TW-SVD algorithm can more flexibly allocate each L_k, $k = 1, 2, \cdots, N$, e.g., the smallest-value collection $\{L_k\}_{k=1}^N$ with $L_k = 1$ can be assigned for Algorithm 1 without error. Consequently, variable L in Remark 1 can be significantly reduced compared with Tucker decomposition, thus relieving the burden of exponential increases in parameter storage.

Algorithm 1 TW-SVD.

Input: An Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the predefined TW-ranks $\boldsymbol{r} = (R_1, R_2, \cdots, R_N, L_1, L_2, \cdots, L_N)$, and the permutation vector $\boldsymbol{z} = (N+2, 1:N+1)$.

Trim R_k and L_k, $k = 1, 2, \cdots, N$, to smaller $\bar{R}_k$ and $\bar{L}_k$, if the upper bounds for truncated SVDs are violated.
Generate the unfolding matrix $\mathbf{X}_{<1>} \in \mathbb{R}^{I_1 \times \prod_{i=2}^{N} I_i}$.
$(\mathbf{G}_1)_{(2)} \leftarrow \mathscr{L}_{\bar{R}_1 \bar{L}_1 \bar{R}_2}(\mathbf{X}_{<1>})$.
$(\mathbf{M}_{\neq 1})_{[(N+1,N+2,1:N);3]} \leftarrow (\mathbf{G}_1)_{(2)}^{\top} \mathbf{X}_{<1>}$.
for $k = 2$ to $N-1$ **do**
  $(\mathbf{G}_k)_{[(1:4);2]} \leftarrow \mathscr{L}_{\bar{L}_k \bar{R}_{k+1}} \left(\left(\mathbf{M}_{\neq(1:k-1)} \right)_{[(1:N+2);2]} \right)$.
  $\left(\mathbf{M}_{\neq(1:k)} \right)_{[\boldsymbol{z};2]} \leftarrow (\mathbf{G}_k)_{[(1:4);2]}^{\top} \left(\mathbf{M}_{\neq(1:k-1)} \right)_{[(1:N+2);2]}$.
end for
$(\mathbf{G}_N)_{(3)}^{\top} \leftarrow \mathscr{L}_{\bar{L}_N} \left(\left(\mathbf{M}_{\neq(1:N-1)} \right)_{[(1:N+2);3]} \right)$.
$\mathbf{C}_{(N)} \leftarrow (\mathbf{G}_N)_{(3)} \left(\mathbf{M}_{\neq(1:N-1)} \right)_{[(1:N+2);3]}$.
Zero-pad $\mathcal{G}_k$, $k = 1, 2, \cdots, N$, and $\mathcal{C}$ to intended sizes.

Output: TW factors $\mathcal{G}_k$, $k = 1, 2, \cdots, N$, and $\mathcal{C}$.

B. TW-ALS Algorithm

Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and its predefined TW-ranks, then the TW-ALS algorithm mainly aims to optimize the latent TW factors $\mathcal{G}_k$, $k = 1, 2, \cdots, N$, and $\mathcal{C}$ such that a relative error between $\mathcal{X}$ and $\text{TW}[\![\{\mathcal{G}_k\}_{k=1}^{N}; \mathcal{C}]\!]$ is minimum, i.e.,

$$\min_{\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_N, \mathcal{C}} \ \|\mathcal{X} - \text{TW}[\![\{\mathcal{G}_k\}_{k=1}^{N}; \mathcal{C}]\!]\|_F. \tag{22}$$

Based on Theorem 3, the minimization problem (22) can be arguably optimized under the ALS framework by alternatively solving $N+1$ least-squares subproblems, i.e., $\min_{(\mathbf{G}_k)_{(2)}} \|\mathbf{X}_{<k>} - (\mathbf{G}_k)_{(2)} (\mathbf{M}_{\neq k})_{[\boldsymbol{u};3]}\|_F$, $k = 1, 2, \cdots, N$, and $\min_{\mathbf{c}_{[(1:N);0]}} \|\mathbf{x}_{[(1:N);0]} - \mathbf{c}_{[(1:N);0]} (\mathbf{N}_{\neq \mathcal{C}})_{[\boldsymbol{v};N]}\|_F$. Compared to formula (10), since we utilize the simplest case without permutation for updating $\mathcal{C}$, $\mathbf{x}_{[(1:N);0]}$ and $\mathbf{c}_{[(1:N);0]}$ imply $\bar{\mathbf{x}}^{\boldsymbol{n}}_{[(1:N);0]}$ and $\bar{\mathbf{c}}^{\boldsymbol{n}}_{[(1:N);0]}$ with $\boldsymbol{n} = (1, 2, \cdots, N)$, respectively. The numerical procedure is detailed in Algorithm 2.

V. Numerical Application to Completion

To validate the rationality and superiority of TW decomposition, we employ it to one classical tensor recovery problem, i.e., tensor completion (TC), whose objective is recovering the missing entries from a partially observed tensor, then formulating a TW decomposition-based TC (TW-TC) model. The proposed TW-TC method is committed to exploring latent TW factors from the partially observed tensor, then predicting the missing entries using these constructed factors. This essentially evaluates the potential capability of TW decomposition to characterize the high-order tensors.

A. Model and Algorithm

Given a partially observed tensor $\mathcal{F} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ at location set $\Omega \subseteq \{1, \cdots, I_1\} \times \{1, \cdots, I_2\} \times \cdots \times \{1, \cdots, I_N\}$, then the proposed TW-TC model formally aims to identify an optimum-TW-ranks approximation $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$

Algorithm 2 TW-ALS.

Input: An Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the manually given TW-ranks $\boldsymbol{r}$, and the threshold $\epsilon > 0$.

Initialization: The randomized $\mathcal{G}_k \in \mathbb{R}^{R_k \times I_k \times L_k \times R_{k+1}}$, $k = 1, 2, \cdots, N$, and the randomized $\mathcal{C} \in \mathbb{R}^{L_1 \times L_2 \times \cdots \times L_N}$.

while *not converged* **do**
  Record the last-update result: $\mathcal{T} \leftarrow \text{TW}[\![\{\mathcal{G}_k\}_{k=1}^{N}; \mathcal{C}]\!]$.
  for $k = 1$ to N **do**
    Determine the subwheel tensor $\mathcal{M}_{\neq k}$ and vector $\boldsymbol{u}$ by Theorem 3.
    $(\mathbf{G}_k)_{(2)} \leftarrow \underset{(\mathbf{G}_k)_{(2)}}{\arg\min} \|\mathbf{X}_{<k>} - (\mathbf{G}_k)_{(2)} (\mathbf{M}_{\neq k})_{[\boldsymbol{u};3]}\|_F$.
  end for
  Determine the subwheel tensor $\mathcal{N}_{\neq \mathcal{C}}$ and vector $\boldsymbol{v}$ by Theorem 3.
  $\mathbf{c}_{[(1:N);0]} \leftarrow \underset{\mathbf{c}_{[(1:N);0]}}{\arg\min} \|\mathbf{x}_{[(1:N);0]} - \mathbf{c}_{[(1:N);0]} (\mathbf{N}_{\neq \mathcal{C}})_{[\boldsymbol{v};N]}\|_F$.
  Check the convergence criterion: $\|\text{TW}[\![\{\mathcal{G}_k\}_{k=1}^{N}; \mathcal{C}]\!] - \mathcal{T}\|_F / \|\mathcal{T}\|_F < \epsilon$.
end while

Output: TW factors $\mathcal{G}_k$, $k = 1, 2, \cdots, N$, and $\mathcal{C}$.

of the tensor $\mathcal{F}$. Mathematically, the TW-TC model can be formulated as the unconstrained problem as follows,

$$\min_{\mathcal{X}, \mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_N, \mathcal{C}} \ \frac{1}{2} \|\mathcal{X} - \text{TW}[\![\{\mathcal{G}_k\}_{k=1}^{N}; \mathcal{C}]\!]\|_F^2 + \iota(\mathcal{X}) \tag{23}$$

with

$$\iota(\mathcal{X}) := \begin{cases} 0, & \mathcal{X} \in \{\mathcal{L} : \mathscr{P}_\Omega(\mathcal{L}) = \mathscr{P}_\Omega(\mathcal{F})\}; \\ \infty, & \text{otherwise}, \end{cases} \tag{24}$$

where $\mathscr{P}_\Omega(\cdot)$ is a projection function keeping the entries in Ω while forcing all the others to zeros.

Obviously, the object function in (23) is not jointly but independently convex for all optimization variables. Thus, we develop a proximal alternating minimization (PAM) [45] based algorithm to alternately and recursively optimize each variable, leading to the following constructive procedure for $k = 1, 2, \cdots, N$,

$$\begin{cases} \mathcal{G}_k^{(t+1)} \in \underset{\mathcal{G}_k}{\arg\min} \Big\{ \frac{\rho}{2} \|\mathcal{G}_k - \mathcal{G}_k^{(t)}\|_F^2 \\ \qquad + \frac{1}{2} \|\mathcal{X}^{(t)} - \text{TW}[\![\mathcal{G}_{1:k-1}^{(t+1)}, \mathcal{G}_k, \mathcal{G}_{k+1:N}^{(t)}; \mathcal{C}^{(t)}]\!]\|_F^2 \Big\}, \\ \mathcal{C}^{(t+1)} \in \underset{\mathcal{C}}{\arg\min} \Big\{ \frac{1}{2} \|\mathcal{X}^{(t)} - \text{TW}[\![\mathcal{G}_{1:N}^{(t+1)}; \mathcal{C}]\!]\|_F^2 \\ \qquad + \frac{\rho}{2} \|\mathcal{C} - \mathcal{C}^{(t)}\|_F^2 \Big\}, \\ \mathcal{X}^{(t+1)} \in \underset{\mathcal{X}}{\arg\min} \Big\{ \frac{1}{2} \|\mathcal{X} - \text{TW}[\![\mathcal{G}_{1:N}^{(t+1)}; \mathcal{C}^{(t+1)}]\!]\|_F^2 \\ \qquad + \frac{\rho}{2} \|\mathcal{X} - \mathcal{X}^{(t)}\|_F^2 + \iota(\mathcal{X}) \Big\}, \end{cases} \tag{25}$$

where $\rho > 0$ is a proximal parameter. Following Theorem 3 again, each univariate minimization problem in formula (25) can certainly be reduced to a linear least-squares problem. More specifically,

1) Updating $\mathcal{G}_k$, $k = 1, 2, \cdots, N$: The unfolding form of the $\mathcal{G}_k$-subproblem can be given by

$$(\mathbf{G}_k^{(t+1)})_{(2)} \in \underset{(\mathbf{G}_k)_{(2)}}{\arg\min} \Big\{ \frac{\rho}{2} \|(\mathbf{G}_k)_{(2)} - (\mathbf{G}_k^{(t)})_{(2)}\|_F^2 + \frac{1}{2} \|\mathbf{X}_{<k>}^{(t)} - (\mathbf{G}_k)_{(2)} (\mathbf{M}_{\neq k}^{(t)})_{[\boldsymbol{u};3]}\|_F^2 \Big\}, \tag{26}$$

where $\mathcal{M}_{\neq k}^{(t)}$ and $\boldsymbol{u}$ are obtained by relying upon Theorem 3. Then, we have the following closed-form solution

$$\mathcal{G}_k^{(t+1)} = \texttt{Fold}_{(2)} \Big\{ \big(\mathbf{X}_{<k>}^{(t)} (\mathbf{M}_{\neq k}^{(t)})_{[\boldsymbol{u};3]}^\top + \rho (\mathbf{G}_k^{(t)})_{(2)}\big) \Big((\mathbf{M}_{\neq k}^{(t)})_{[\boldsymbol{u};3]} (\mathbf{M}_{\neq k}^{(t)})_{[\boldsymbol{u};3]}^\top + \rho \mathbf{I} \Big)^{-1} \Big\}. \tag{27}$$

2) Updating $\mathcal{C}$: Similarly, the $\mathcal{C}$-subproblem can be rewritten as

$$\mathbf{c}_{[(1:N);0]}^{(t+1)} \in \underset{\mathbf{c}_{[(1:N);0]}}{\arg\min} \Big\{ \frac{\rho}{2} \|\mathbf{c}_{[(1:N);0]} - \mathbf{c}_{[(1:N);0]}^{(t)}\|_F^2 + \frac{1}{2} \|\mathbf{x}_{[(1:N);0]}^{(t)} - \mathbf{c}_{[(1:N);0]} (\mathbf{N}_{\neq \mathcal{C}}^{(t)})_{[\boldsymbol{v};N]}\|_F^2 \Big\}, \tag{28}$$

where $\mathcal{N}_{\neq \mathcal{C}}^{(t)}$ and $\boldsymbol{v}$ are determined by Theorem 3. The above problem has a minimum solution, appearing as

$$\mathcal{C}^{(t+1)} = \texttt{Fold}_{[(1:N);0]} \Big\{ \big(\mathbf{x}_{[(1:N);0]}^{(t)} (\mathbf{N}_{\neq \mathcal{C}}^{(t)})_{[\boldsymbol{v};N]}^\top + \rho \mathbf{c}_{[(1:N);0]}^{(t)}\big) \Big((\mathbf{N}_{\neq \mathcal{C}}^{(t)})_{[\boldsymbol{v};N]} (\mathbf{N}_{\neq \mathcal{C}}^{(t)})_{[\boldsymbol{v};N]}^\top + \rho \mathbf{I} \Big)^{-1} \Big\}. \tag{29}$$

3) Updating $\mathcal{X}$: The $\mathcal{X}$-subproblem also reduces to a least-squares problem, thus yielding the closed-form solution

$$\mathcal{X}^{(t+1)} = \mathscr{P}_{\Omega^c} \Big(\frac{\text{TW}[\![\{\mathcal{G}_k^{(t+1)}\}_{k=1}^N; \mathcal{C}^{(t+1)}]\!] + \rho \mathcal{X}^{(t)}}{1+\rho} \Big) + \mathscr{P}_{\Omega}(\mathcal{F}), \tag{30}$$

where Ω^c indicates the complementary set of Ω.

In numerical TC experiments, the variable $\mathcal{C}$ typically tends to be stable when it exceeds 200 iterations (see Fig. 4). Therefore, we utilize the modulo operation with a step size to intermittently update $\mathcal{C}$ after 200 iterations, aiming to save computing resources. The PAM-based solving procedure for the developed TW-TC model is summarized in Algorithm 3.

Algorithm 3 The PAM-Based Solver for TW-TC Model.

Input: The observed tensor $\mathcal{F} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the location set Ω, the TW-ranks $\boldsymbol{r}$, the step size $s = 20$, $\rho = 10^{-1}$, $t_{\max} = 1000$, and $\epsilon = 10^{-5}$.

Initialization: The iteration step $t = 0$, $\mathcal{X}^{(0)} = \mathcal{F}$, the randomized $\mathcal{G}_k^{(0)} \in \mathbb{R}^{R_k \times I_k \times L_k \times R_{k+1}}$, $k = 1, 2, \cdots, N$, and the randomized $\mathcal{C}^{(0)} \in \mathbb{R}^{L_1 \times L_2 \times \cdots \times L_N}$.

1: **while** *not converged* and $t < t_{\max}$ **do**
2: Update $\mathcal{G}_k^{(t+1)}$, $k = 1, 2, \cdots, N$, by (27).
3: **if** $t > 200$ and $\texttt{mod}(t, s) \neq 0$ **then**
4: $\mathcal{C}^{(t+1)} = \mathcal{C}^{(t)}$.
5: **else**
6: Update $\mathcal{C}^{(t+1)}$ by (29).
7: **end if**
8: Update $\mathcal{X}^{(t+1)}$ by (30).
9: Check the convergence criterion: $\|\mathcal{X}^{(t+1)} - \mathcal{X}^{(t)}\|_F / \|\mathcal{X}^{(t)}\|_F < \epsilon$.
10: $t \leftarrow t + 1$.
11: **end while**

Output: The recovered tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$.

B. Computational Complexity Analysis

For Algorithm 3, the computational complexity is mainly involved in updating $\{\mathcal{G}_k\}_{k=1}^N$, $\mathcal{C}$, and $\mathcal{X}$, which concretely are $\mathcal{O}(\sum_{k=1}^N (R_k R_{\vartheta(N+k-1,N)} \prod_{j=k}^{N+k-2} I_{\vartheta(j,N)} \prod_{l=1}^N L_l) + \sum_{k=1}^N \sum_{i=k+1}^{N+k-2} (R_k R_{\vartheta(i,N)} R_{\vartheta(i+1,N)} \prod_{j=k}^{i} I_{\vartheta(j,N)} L_{\vartheta(j,N)}) + \sum_{k=1}^N (R_k R_{\vartheta(N+k-1,N)} L_{\vartheta(N+k-1,N)} \prod_{l=1}^N I_l) + \sum_{k=1}^N (R_k^2 R_{\vartheta(N+k-1,N)}^2 L_{\vartheta(N+k-1,N)}^2 \prod_{j=k}^{N+k-2} I_{\vartheta(j,N)}) + \sum_{k=1}^N (R_k^3 R_{\vartheta(N+k-1,N)}^3 L_{\vartheta(N+k-1,N)}^3))$, $\mathcal{O}(\prod_{l=1}^N I_l L_l^2 + \sum_{i=2}^{N-1} (R_1 R_i R_{i+1} \prod_{j=1}^{i} I_j L_j) + R_1 R_N \prod_{l=1}^N I_l L_l + \prod_{l=1}^N L_l^3)$, and $\mathcal{O}(\sum_{i=2}^{N-1} (R_1 R_i R_{i+1} \prod_{j=1}^{i} I_j L_j) + R_1 R_N \prod_{l=1}^N I_l L_l)$, respectively. Consequently, the overall computational complexity for each nonintermittent iteration (i.e., $t \in \{t : 0 \leq t \leq 200 \text{ or } \texttt{mod}(t, s) = 0\}$) in Algorithm 3 is $\mathcal{O}(\sum_{k=1}^N (R_k R_{\vartheta(N+k-1,N)} \prod_{j=k}^{N+k-2} I_{\vartheta(j,N)} \prod_{l=1}^N L_l) + \sum_{k=1}^N \sum_{i=k+1}^{N+k-2} (R_k R_{\vartheta(i,N)} R_{\vartheta(i+1,N)} \prod_{j=k}^{i} I_{\vartheta(j,N)} L_{\vartheta(j,N)}) + \sum_{k=1}^N (R_k R_{\vartheta(N+k-1,N)} L_{\vartheta(N+k-1,N)} \prod_{l=1}^N I_l) + \prod_{l=1}^N I_l L_l^2 + \sum_{k=1}^N (R_k^2 R_{\vartheta(N+k-1,N)}^2 L_{\vartheta(N+k-1,N)}^2 \prod_{j=k}^{N+k-2} I_{\vartheta(j,N)}) + \sum_{k=1}^N (R_k^3 R_{\vartheta(N+k-1,N)}^3 L_{\vartheta(N+k-1,N)}^3) + R_1 R_N \prod_{l=1}^N I_l L_l + \prod_{l=1}^N L_l^3)$. In the above expressions, $\vartheta(k, \epsilon)$ is an indicator function that returns k for $1 \leq k \leq \epsilon$ but $\texttt{mod}(k, \epsilon)$ for $k > \epsilon$.

Assume that $R_k = R$ and $I_k \geq R^2 \vee N$, $k = 1, 2, \cdots, N$, then the total computational complexity of Algorithm 3 can be simplified as $\mathcal{O}(R^2 \prod_{k=1}^N I_k L_k + \prod_{k=1}^N I_k L_k^2 + \prod_{k=1}^N L_k^3)$. Akin to the storage complexity in Remark 1, although the computational complexity also increases exponentially with dimensionality N, a smaller-value collection $\{L_k\}_{k=1}^N$ can be intentionally followed to alleviate the computational burden, thus facilitating the applicability of TW decomposition for higher-order and larger-scale tensor recovery tasks.

C. Convergence Analysis

This section consists in theoretically proving the convergence of Algorithm 3. For brevity, we employ $\mathcal{Z}$ and $\Phi(\mathcal{Z})$ to denote the $(\mathcal{X}, \mathcal{G}_{1:N}, \mathcal{C})$ and the objective function in (23), respectively. The main results are explicated as follows.

Lemma 1. *The $\Phi(\mathcal{Z})$ is a Kurdyka-Łojasiewicz (KŁ) function.*

Lemma 2. *Let $\{\mathcal{Z}^{(t)}\}_{t \in \mathbb{N}}$ be the sequence generated by Algorithm 3. Then, the sequence $\{\Phi(\mathcal{Z}^{(t)})\}_{t \in \mathbb{N}}$ sufficiently decreases, i.e., $\Phi(\mathcal{Z}^{(t)}) - \Phi(\mathcal{Z}^{(t+1)}) \geq \rho/2 \|\mathcal{Z}^{(t+1)} - \mathcal{Z}^{(t)}\|_F^2$, where $\|\mathcal{Z}^{(t+1)} - \mathcal{Z}^{(t)}\|_F^2 = \|\mathcal{X}^{(t+1)} - \mathcal{X}^{(t)}\|_F^2 + \sum_{k=1}^N \|\mathcal{G}_k^{(t+1)} - \mathcal{G}_k^{(t)}\|_F^2 + \|\mathcal{C}^{(t+1)} - \mathcal{C}^{(t)}\|_F^2$.*

Lemma 3. *Let $\{\mathcal{Z}^{(t)}\}_{t \in \mathbb{N}}$ be the sequence generated by Algorithm 3. Then, there exists $\|\partial \Phi(\mathcal{Z}^{(t+1)})\|_F \leq (L_\Phi + (N+2)\rho) \|\mathcal{Z}^{(t+1)} - \mathcal{Z}^{(t)}\|_F$, where L_Φ is the sum of the Lipschitz constants of $\partial_{\mathcal{G}_k} \Phi(\mathcal{Z})$ and $\partial_{\mathcal{C}} \Phi(\mathcal{Z})$, i.e., $L_\Phi = \sum_{k=1}^N L_{\mathcal{G}_k} + L_{\mathcal{C}}$.*

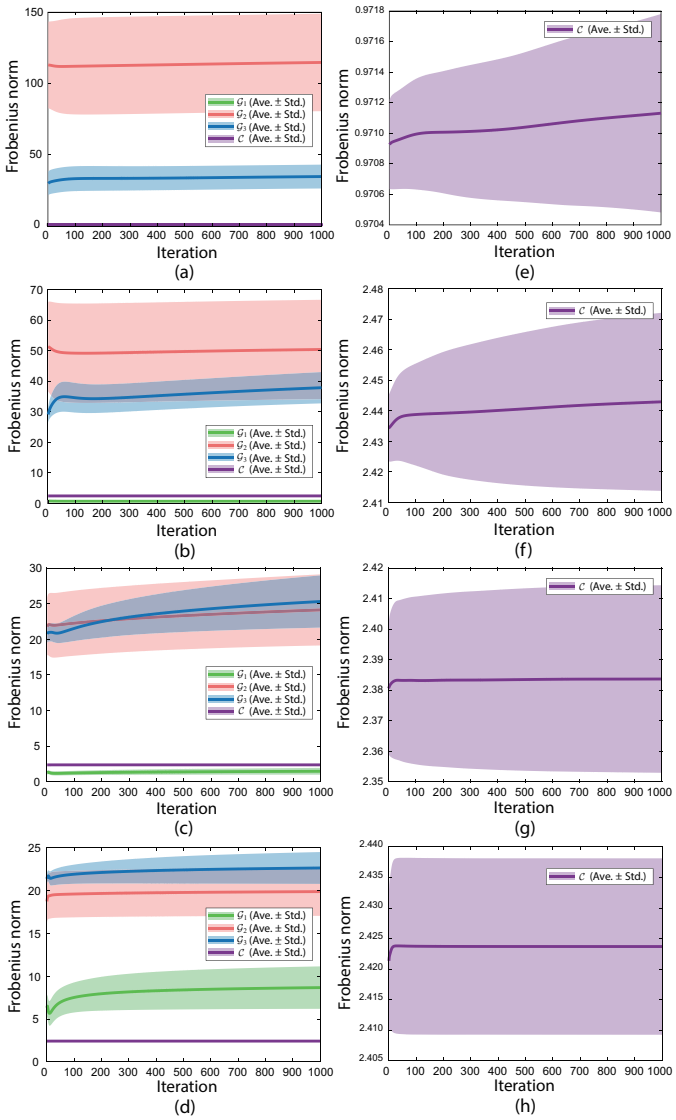


Fig. 4. The Frobenius norm of TW factors (i.e., $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{G}_3$, and $\mathcal{C}$) against the iteration when the PAM-based TW-TC algorithm is performed over 1000 third-order synthetic tensors. The (a), (b), (c), and (d) illustrate the fluctuations of Frobenius norm when the sampling rate is 5%, 10%, 20%, and 40%, respectively. Furthermore, the (e), (f), (g), and (h) present the closeups regarding factor $\mathcal{C}$ in (a), (b), (c), and (d), respectively. The experimental tensors, whose sizes are $60 \times 60 \times 60$ and Tucker-ranks are $(6, 6, 6)$, are synthesized in Tucker format by using Tucker factors sampled from the uniform distribution $U(0, 1)$ for 1000 different seeds.

Lemma 4. *Let $\{\mathcal{Z}^{(t)}\}_{t\in\mathbb{N}}$ be the sequence generated by Algorithm 3, then it is bounded.*

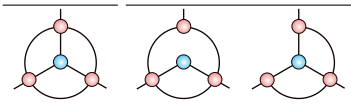
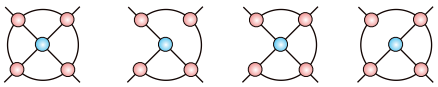
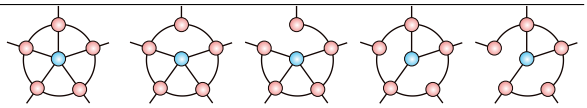
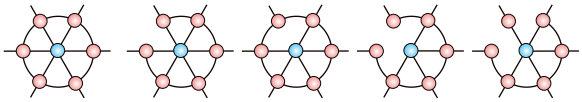
According to the finite length property (see [46, Theorem 1]), since Lemmas 1-4 are satisfied, the convergence guarantee of Algorithm 3 can be similarly established in Theorem 6.

Theorem 6 (Local Convergence). *Let $\{\mathcal{Z}^{(t)}\}_{t\in\mathbb{N}}$ be the sequence generated by Algorithm 3, then it globally converges to a critical point (i.e., local minimum point) of $\Phi(\mathcal{Z})$.*

VI. Numerical Experiments

In this section, we design substantial numerical experiments on synthetic and real-world data to verify the superiority of the proposed TW-TC method over the other comparators, which are constructed based on several commonly used tensor decompositions. All the experiments are implemented in MATLAB (R2021a) on a computer with an Intel(R) Core(TM) i9-10900KF 3.70GHz CPU and 64GB memory.

TABLE I
Statistics of the actual topological structures for TW decomposition and the corresponding WFLR values. The WFLR is the average value of $3 \times Count$ independent experiments used in the TW-ranks tuning.

		Wheel	Subwheel			
Third order						
	Count	3	2	1	–	–
	WFLR	0.00%	3.92%	3.35%	–	–
Fourth order						
	Count	2	2	1	1	–
	WFLR	0.00%	4.47%	2.46%	0.43%	–
Fifth order						
	Count	1	2	1	1	1
	WFLR	0.00%	0.79%	2.67%	3.39%	4.38%
Sixth order						
	Count	1	2	1	1	1
	WFLR	0.00%	1.98%	1.10%	3.45%	3.24%

A. Synthetic Data Experiments

To testify the superiority of TW decomposition compared with four related tensor decompositions, i.e., TT [28], TR [29], FCTN [32], and Tucker [15] ones, we firstly conduct the numerical experiments on synthetic tensor data by horizontally comparing five PAM-based TC algorithms, i.e., TT-TC (PAM), TR-TC (PAM), FCTN-TC (PAM), Tucker-TC (PAM), and TW-TC (PAM) ones. Since the Tucker factors comprise an underlying high-order structure, which may be closer to reality, we construct the synthetic tensor $\mathcal{X}_r \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ by the Nth-order Tucker decomposition using Tucker factors sampled from the uniform distribution $U(0, 1)$. Mainly, the synthetic Tucker-format data includes two types, i.e.,

- Low-Tucker-rank simulation (LTS): The LTS consists of 125 third-order, 144 fourth-order, 108 fifth-order, and 144 sixth-order tensors, whose sizes are $\{I_1 \times I_2 \times I_3 : I_1, I_2, I_3 \in \{40, 45, 50, 55, 60\}\}$, $\{I_1 \times I_2 \times I_3 \times I_4 : I_1, I_3 \in \{14, 16, 18, 20\}, I_2, I_4 \in \{16, 18, 20\}\}$, $\{I_1 \times I_2 \times I_3 \times I_4 \times I_5 : I_1, I_3, I_5 \in \{6, 8, 10\}, I_2, I_4 \in \{8, 10\}\}$, and $\{I_1 \times I_2 \times I_3 \times I_4 \times I_5 \times I_6 : I_1, I_4 \in \{3, 4, 5\}, I_2, I_5 \in \{3, 4\}, I_3, I_6 \in \{4, 5\}\}$, respectively. The corresponding Tucker-ranks are $(5, 5, 5)$, $(3, 3, 3, 3)$, $(2, 2, 2, 2, 2)$, and $(2, 2, 2, 2, 2, 2)$, respectively.

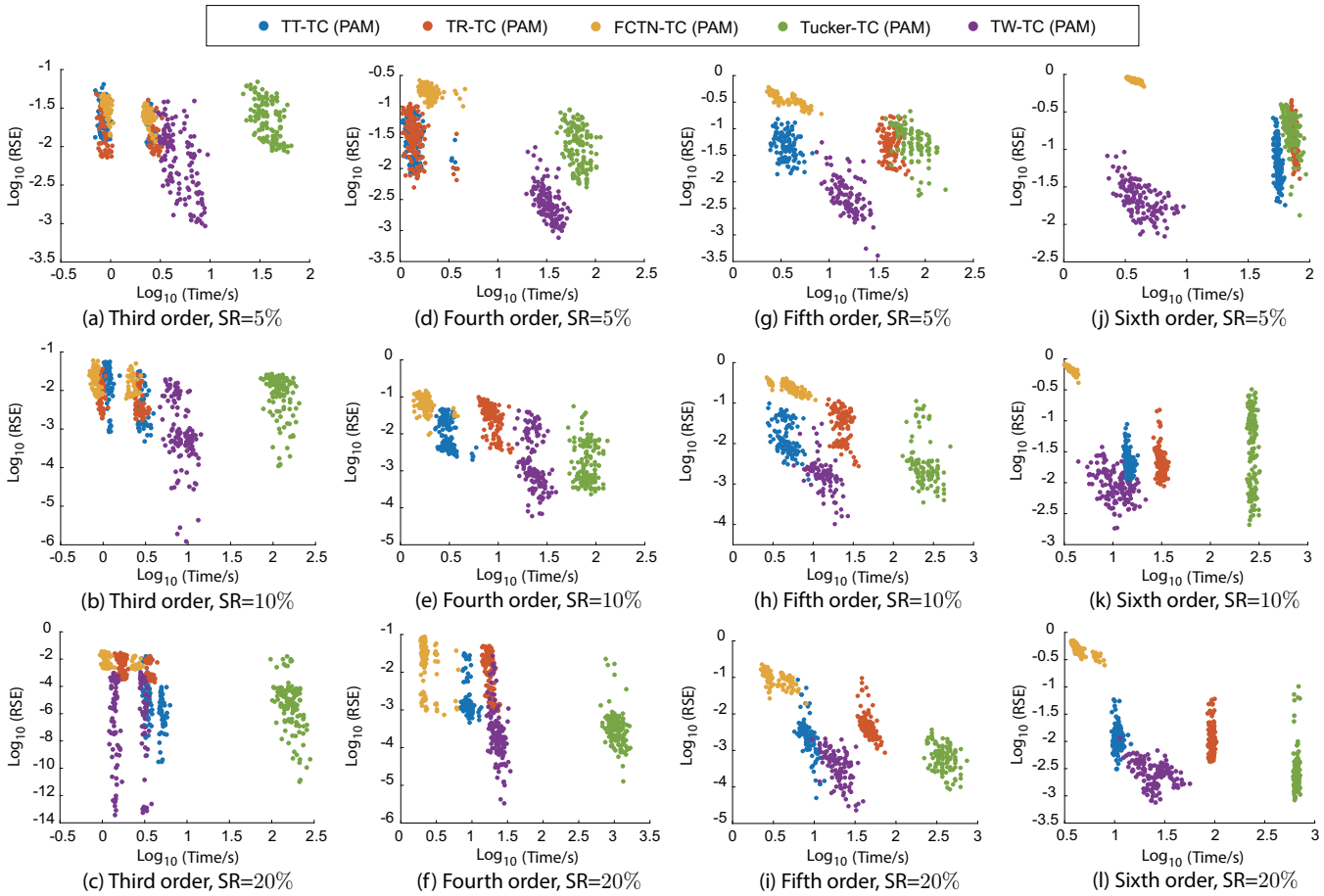


Fig. 5. Recovery performance (i.e., Log_{10}(RSE)) against computing cost (i.e., Log_{10}(Time/s)) on LTS. The TT-TC, TR-TC, FCTN-TC, Tucker-TC, and TW-TC methods are evaluated under 12 different cases. The first column: 125 third-order tensor experiments with SR $= 5\%$, 10%, and 20%; The second column: 144 fourth-order tensor experiments with SR $= 5\%$, 10%, and 20%; The third column: 108 fifth-order tensor experiments with SR $= 5\%$, 10%, and 20%; The fourth column: 144 sixth-order tensor experiments with SR $= 5\%$, 10%, and 20%. The smaller Log_{10}(RSE) value implies higher accuracy.

TABLE II
QUALITY METRICS AND AVERAGE COMPUTATIONAL TIMES (/S: SECONDS) OF ALL COMPARED METHODS UNDER DIFFERENT SRS FOR MSIS. THE MPSNR AND MSSIM ARE THE AVERAGES OF 31 INDEPENDENT MSI EXPERIMENTS. (**BOLD**: BEST; UNDERLINE: SECOND BEST)

Method	SR $= 0.1\%$		SR $= 0.5\%$		SR $= 1\%$		SR $= 5\%$		SR $= 10\%$		SR $= 20\%$		SR $= 40\%$		Time/s
	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	
Observed	16.07	0.209	16.09	0.212	16.11	0.217	16.29	0.249	16.53	0.286	17.04	0.354	18.29	0.473	–
HaLRTC [47]	16.07	0.209	16.09	0.213	16.31	0.267	25.63	0.767	30.23	0.854	35.47	0.930	42.32	0.979	8.76
t-SVD [48]	16.09	0.220	16.53	0.341	17.64	0.394	25.11	0.608	35.14	0.908	40.54	0.963	47.65	0.990	31.31
TMacTT [37]	16.05	0.204	15.82	0.277	22.61	0.539	25.64	0.705	32.92	0.876	35.22	0.927	43.71	0.977	193.32
LRTFR [49]	19.40	0.347	22.46	0.382	25.18	0.592	**34.54**	0.846	**40.64**	**0.954**	44.90	0.984	48.45	0.992	9.16
RR-FBTC [50]	17.90	0.332	21.39	0.481	25.12	**0.646**	32.77	0.846	34.23	0.869	35.46	0.887	36.16	0.896	569.26
TT-TC	17.17	0.347	20.75	0.467	22.98	0.540	28.01	0.749	32.94	0.869	35.59	0.917	47.26	0.981	72.23
TR-TC	16.85	0.325	19.68	0.431	22.80	0.523	31.02	0.820	35.54	0.904	41.78	0.964	49.52	0.986	64.29
FCTN-TC	16.89	0.326	19.73	0.430	22.39	0.503	30.84	0.814	35.56	0.908	41.52	0.963	49.93	0.988	60.00
TW-TC	**19.59**	**0.401**	**23.27**	**0.504**	**25.86**	0.606	34.22	**0.877**	39.89	0.951	**46.83**	**0.986**	**55.33**	**0.997**	189.98

- High-Tucker-rank simulation (HTS): The HTS adopts the identical tensor configurations with that in LTS. The only difference is that the Tucker-ranks are increased to $(10, 10, 10)$, $(6, 6, 6, 6)$, $(4, 4, 4, 4, 4)$, and $(4, 4, 4, 4, 4, 4)$.

All synthetic data are numerically renormalized into the interval [0, 1]. Subsequently, the partially observed tensors are generated by random sampling with three sampling rates (SRs): 5%, 10%, and 20%. The performance is assessed by the residual standard error (RSE), i.e., $\|\mathcal{X} - \mathcal{X}_r\|_F / \|\mathcal{X}_r\|_F$, where $\mathcal{X}$ is the recovered tensor.

1) Result analysis: The RSE values and computing times on LTS and HTS under various SRs are depicted in Fig. 5 and Fig. 6, respectively. Compared with the TT-TC, TR-TC, and FCTN-TC models, the TW-TC method statistically achieves the optimal RSE values for 93.51% of all experiments by using competitive computing costs, which evidently demonstrates the advantageability of TW decomposition. Furthermore, when compared with the Tucker-TC model, the TW-TC model acquires higher recovery accuracy for 91.62% of the experiments and RSE stability (i.e., smaller discretization), even if the

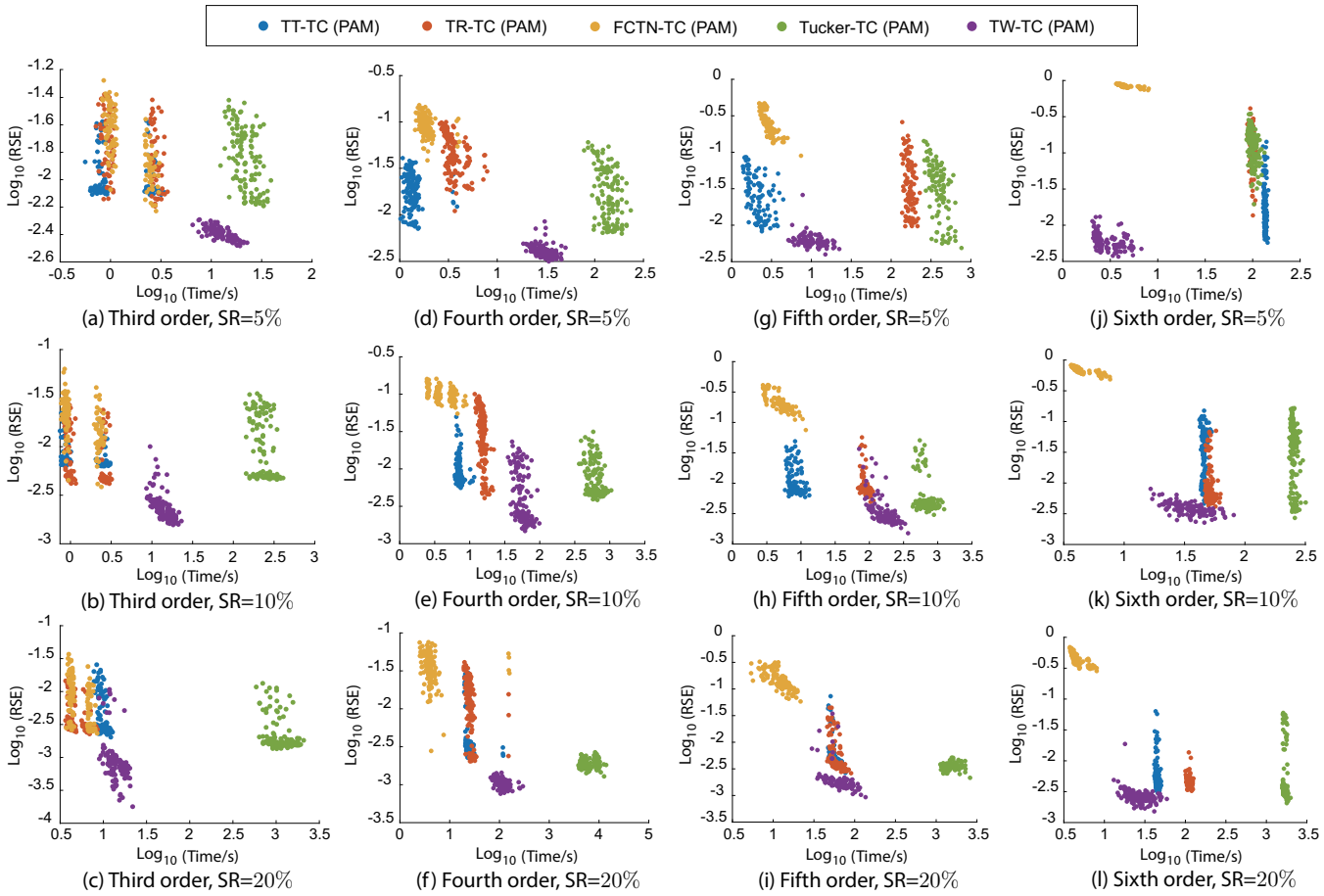


Fig. 6. Recovery performance (i.e., Log_{10}(RSE)) against computing cost (i.e., Log_{10}(Time/s)) on HTS. The TT-TC, TR-TC, FCTN-TC, Tucker-TC, and TW-TC methods are evaluated under 12 different cases. The first column: 125 third-order tensor experiments with SR $= 5\%$, 10%, and 20%; The second column: 144 fourth-order tensor experiments with SR $= 5\%$, 10%, and 20%; The third column: 108 fifth-order tensor experiments with SR $= 5\%$, 10%, and 20%; The fourth column: 144 sixth-order tensor experiments with SR $= 5\%$, 10%, and 20%. The smaller Log_{10}(RSE) value implies higher accuracy.

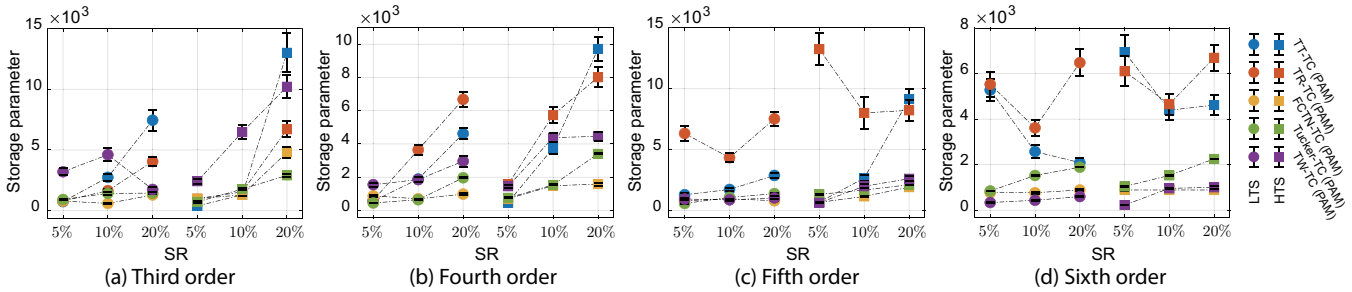


Fig. 7. Storage parameters of the TT-TC, TR-TC, FCTN-TC, Tucker-TC, and TW-TC models on LTS and HTS for (a) third-order, (b) fourth-order, (c) fifth-order, and (d) sixth-order tensor experiments with SR $= 5\%$, 10%, and 20%. Each plotted point in (a), (b), (c), and (d) represents the Ave. $\pm$ Std. of the storage parameters for 125 third-order, 144 fourth-order, 108 fifth-order, and 144 sixth-order independent tensor experiments, respectively.

experimental data is synthesized in Tucker format. Especially, the TW-TC model exhibits higher computing efficiency over the Tucker-TC model, which implies that TW decomposition really alleviates the curse of computational burden, thus achieving greater applicability than Tucker decomposition.

From a comparative perspective, the TW-TC model typically realizes the more significant performance for HTS (e.g., see Fig. 5(a) versus Fig. 6(a)), lower SRs (e.g., see Fig. 5(d) versus Fig. 5(e)), and higher-order tensors (e.g., see Fig. 5(a) versus Fig. 5(d)), which prominently validates the superior characterization ability of TW decomposition, making it easier to reconstruct the underlying high-order tensors under more challenging recovery conditions. Moreover, benefiting from the terrific rank flexibility of TW topology, the TW-TC model also exemplifies the relatively faster computation for HTS (e.g., see Fig. 5(g) versus Fig. 6(g)), higher SRs (e.g., see Fig. 5(b) versus Fig. 5(c)), and higher-order tensors (e.g., see Fig. 5(g) versus Fig. 5(j)). This experimentally supports the previous theoretical analysis of TW decomposition.

Notably, for Nth-order ($N \geq 4$) synthetic tensor experiments, the FCTN-TC model consistently yields small optimal rank configurations, thus showing lower computational times

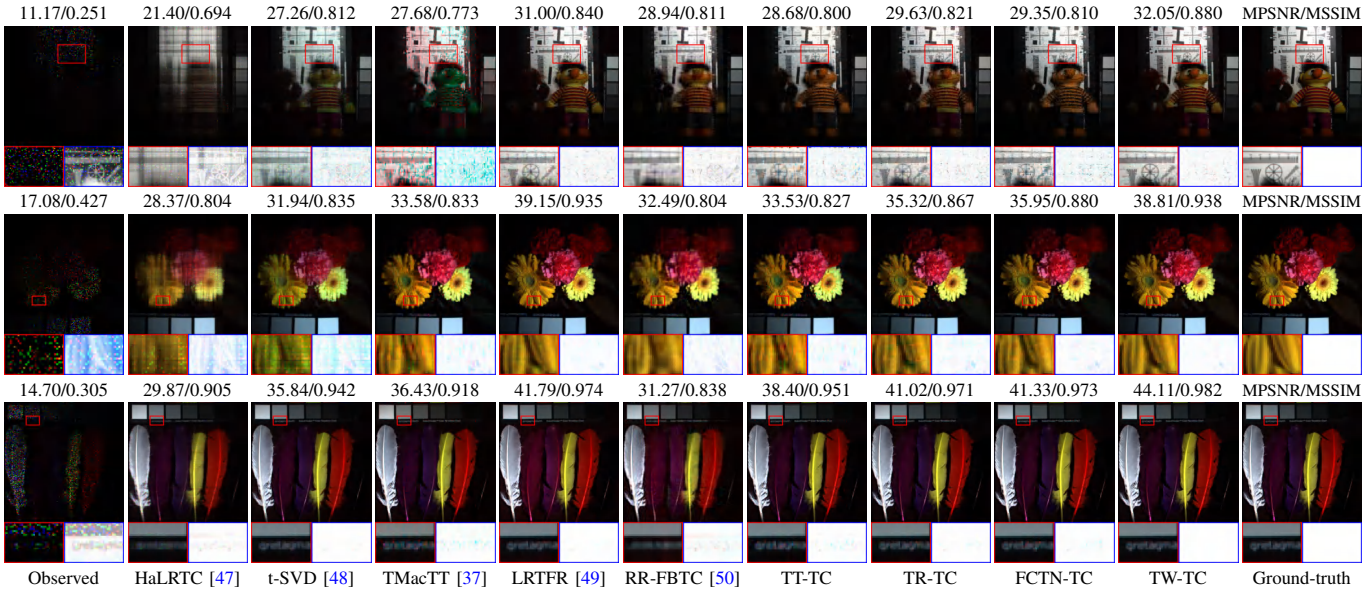


Fig. 8. Visualization of the recovered results on third-order MSIs. From top to bottom: the visual compositions (31-th, 20-th, and 10-th spectral bands as R, G, and B channels, respectively) and residual images for MSI *Toy* (SR $= 5\%$), *Flowers* (SR $= 10\%$), and *Feathers* (SR $= 20\%$), sequentially.

TABLE III
QUALITY METRICS AND AVERAGE COMPUTATIONAL TIMES (/S: SECONDS) OF ALL COMPARED METHODS UNDER DIFFERENT SRS FOR CVS. THE MPSNR AND MSSIM ARE THE AVERAGES OF 10 INDEPENDENT CV EXPERIMENTS. (**BOLD**: BEST; UNDERLINE: SECOND BEST)

Method	SR $= 0.1\%$		SR $= 0.5\%$		SR $= 1\%$		SR $= 5\%$		SR $= 10\%$		SR $= 20\%$		SR $= 40\%$		Time/s
	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	
Observed	6.39	0.002	6.41	0.003	6.43	0.005	6.61	0.014	6.84	0.022	7.35	0.038	8.60	0.073	–
HaLRTC [47]	6.39	0.002	6.42	0.005	6.56	0.020	18.48	0.531	22.25	0.660	26.35	0.804	31.66	0.922	7.02
t-SVD [48]	6.46	0.011	7.63	0.057	10.78	0.134	28.79	0.821	31.73	0.883	34.81	0.931	38.79	0.969	25.99
TMacTT [37]	6.37	0.002	16.65	0.291	19.74	0.431	26.13	0.746	30.38	0.859	33.19	0.914	35.73	0.948	134.49
LRTFR [49]	15.92	0.143	**21.17**	**0.504**	22.79	0.545	**30.29**	0.834	33.41	0.900	36.13	0.945	37.78	0.961	7.56
RR-FBTC [50]	10.91	0.162	18.43	0.503	19.86	0.499	25.62	0.730	26.10	0.744	27.92	0.796	29.06	0.824	457.37
TT-TC	9.61	0.069	17.14	0.283	20.70	0.469	24.21	0.694	29.81	0.844	32.75	0.906	34.91	0.939	61.34
TR-TC	9.15	0.054	15.49	0.186	20.11	0.411	28.62	0.802	32.07	0.883	35.15	0.924	39.74	0.970	166.93
FCTN-TC	7.89	0.034	14.38	0.150	19.24	0.375	27.99	0.765	32.45	0.884	36.33	0.941	40.06	0.974	314.56
TW-TC	**16.48**	**0.272**	20.33	0.411	**22.92**	**0.562**	30.13	**0.872**	**33.77**	**0.913**	**37.08**	**0.953**	**40.42**	**0.976**	330.58

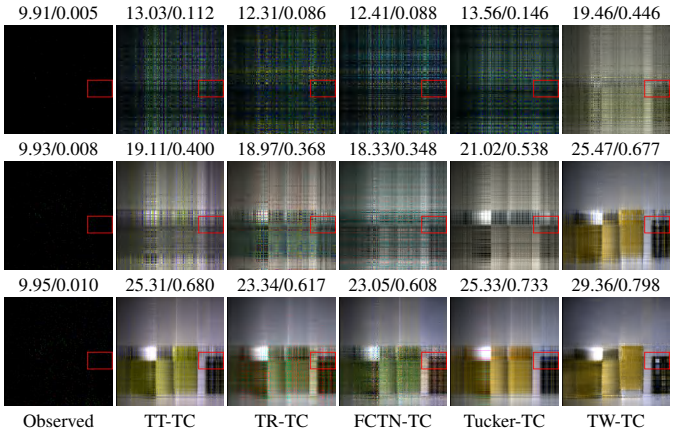


Fig. 9. Recovery performance of the TT-TC, TR-TC, FCTN-TC, Tucker-TC, and TW-TC models on MSI *Beers*. From top to bottom: SRs are 0.1%, 0.5%, and 1%, respectively. The $(31, 20, 10)$-th bands are applied for RGB rendering, and the MPSNR/MSSIM is appended. The average computational times of TT-TC, TR-TC, FCTN-TC, Tucker-TC, and TW-TC are 44.88, 47.63, 42.14, 215.81, and 95.72 seconds, respectively.

TABLE IV
QUALITY METRICS AND AVERAGE COMPUTATIONAL TIMES (/M: MINUTES) OF ALL COMPARED METHODS FOR HSV UNDER THREE SRS: 0.1%, 0.5%, AND 1%. (**BOLD**: BEST; UNDERLINE: SECOND BEST)

Method	SR $= 0.1\%$		SR $= 0.5\%$		SR $= 1\%$		Time /m
	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	
Observed	9.74	0.001	9.76	0.003	9.78	0.004	–
HaLRTC [47]	9.74	0.001	9.77	0.003	9.82	0.007	0.10
t-SVD [48]	9.74	0.001	12.80	0.165	18.19	0.451	0.67
TMacTT [37]	7.86	0.003	22.68	0.577	25.97	0.734	12.34
LRTFR [49]	17.68	0.294	21.57	0.594	26.72	0.801	0.16
RR-FBTC [50]	9.74	0.001	21.06	0.502	22.75	0.612	3.70
TT-TC	14.20	0.186	20.63	0.479	25.58	0.723	2.51
TR-TC	13.94	0.164	20.94	0.479	25.46	0.710	2.62
FCTN-TC	12.85	0.098	19.93	0.376	24.07	0.643	2.65
Tucker-TC	14.68	0.211	21.65	0.524	23.82	0.626	70.54
TW-TC	**19.72**	**0.383**	**25.38**	**0.697**	**28.57**	**0.828**	2.23

over the other methods. Instead of the real-world data, the synthetic data reflects the more random and unpredictable characteristics, in which not all non-adjacent dimensions have a direct relationship, thus leading to the authentic TN ranks between some non-adjacent factors being 1 (i.e., without connection). However, to maintain the fully-connected topology,

Fig. 10. Visualization of the recovered results on fourth-order CVs. From top to bottom: the visual compositions and residual images for CV *Akiyo* (SR $= 5\%$, 11-th frame), *Hall* (SR $= 10\%$, 18-th frame), and *Container* (SR $= 20\%$, 11-th frame), sequentially.

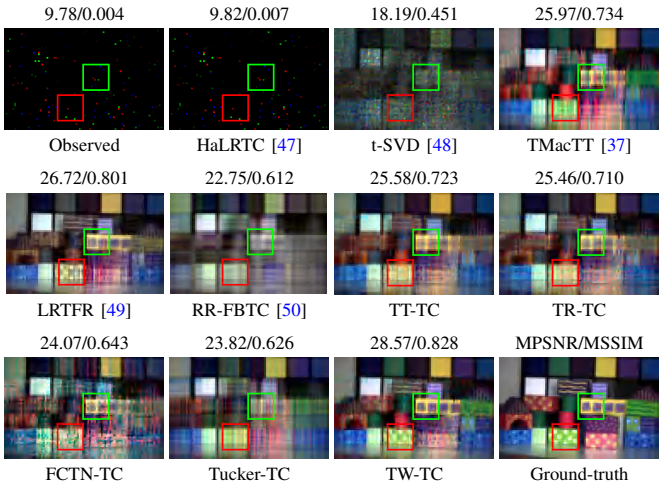


Fig. 11. Recovery performance of all competing methods on HSV when SR $= 1\%$. The $(20, 15, 10)$-th spectral bands of the 11-th frame are used for RGB rendering, and the notable regions are highlighted.

the minimum values of the configured FCTN-ranks among several non-adjacent factors are 2 rather than 1, which inevitably reduces the performance of the FCTN-TC model for possible overfitting. Accordingly, the FCTN-TC model performs inferiorly to the TT-TC and TR-TC methods, despite its more TN topological connections. In particular, when applied to higher-order tensors, the number of ineffective structures in FCTN topology also grows nearly quadratically, resulting in a progressively more significant RSE gap between the FCTN-TC and other methods (e.g., see Fig. 5(h) versus Fig. 5(k)). Nevertheless, the TW-TC model is barely affected, even though TW decomposition has more TN topological connections than FCTN one under Nth-order ($N \leq 4$) cases, which corroborates the validity of TW topology. Regarding the topological analysis, further investigations are presented in Section VI-A3.

TABLE V
QUALITY METRICS AND AVERAGE COMPUTATIONAL TIMES (/M: MINUTES) OF ALL COMPARED METHODS UNDER DIFFERENT SRS FOR LFIS. THE MPSNR AND MSSIM ARE THE AVERAGES OF 10 INDEPENDENT LFI EXPERIMENTS. (**BOLD**: BEST; UNDERLINE: SECOND BEST)

Method	SR $= 0.1\%$		SR $= 0.5\%$		SR $= 1\%$		Time
	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	/m
Observed	8.79	0.053	8.80	0.055	8.82	0.059	–
HaLRTC [47]	8.79	0.053	8.81	0.059	8.87	0.075	0.08
t-SVD [48]	8.79	0.053	8.80	0.055	10.33	0.216	0.16
TMacTT [37]	8.76	0.049	11.10	0.134	17.86	0.408	3.12
LRTFR [49]	12.79	0.118	17.00	**0.371**	17.83	0.380	0.13
RR-FBTC [50]	8.78	0.052	11.10	0.137	11.97	0.174	1.37
TT-TC	10.87	0.133	15.20	0.239	17.67	0.385	1.17
TR-TC	10.51	0.117	14.84	0.225	17.25	0.341	1.31
FCTN-TC	9.17	0.081	11.31	0.107	14.04	0.186	5.01
Tucker-TC	11.25	0.144	15.05	0.260	17.36	0.379	26.68
TW-TC	**15.13**	**0.245**	**17.90**	0.366	**19.72**	**0.490**	2.37
Method	SR $= 5\%$		SR $= 10\%$		SR $= 20\%$		Time
	MPSNR	MSSIM	MPSNR	MSSIM	MPSNR	MSSIM	/m
Observed	9.00	0.083	9.24	0.113	9.75	0.173	–
HaLRTC [47]	11.09	0.313	16.54	0.508	20.88	0.689	0.18
t-SVD [48]	20.22	0.538	24.32	0.747	27.71	0.849	0.33
TMacTT [37]	21.98	0.627	23.85	0.720	25.83	0.803	2.88
LRTFR [49]	21.72	0.645	26.19	0.783	29.23	0.854	0.13
RR-FBTC [50]	13.93	0.247	15.71	0.346	17.51	0.421	1.70
TT-TC	21.56	0.606	23.79	0.718	25.91	0.798	3.34
TR-TC	22.86	0.643	26.21	0.774	29.53	0.873	6.83
FCTN-TC	23.19	0.647	26.48	0.772	29.96	0.873	44.29
TW-TC	**24.23**	**0.723**	**27.34**	**0.809**	**30.65**	**0.891**	30.08

2) Storage parameter comparison: For each experimental case, e.g., (LTS, third order, SR $= 5\%$), the hyper-parameters[4] of five decomposition-based models, i.e., the TT-ranks, TR-ranks, FCTN-ranks, Tucker-ranks, and TW-ranks, are determined by respectively rounding the average of the fine-tuned rank collections on three randomly selected experiments. Afterwards, the storage parameters of all models can be computed by tensor sizes and ranks, e.g., $\mathcal{O}(\sum_{k=1}^{N} R_k I_k L_k R_{k+1} +$

[4]Here only the tensor ranks are considered as the hyper-parameters, since $\rho = 10^{-1}$ is uniformly designated in advance.

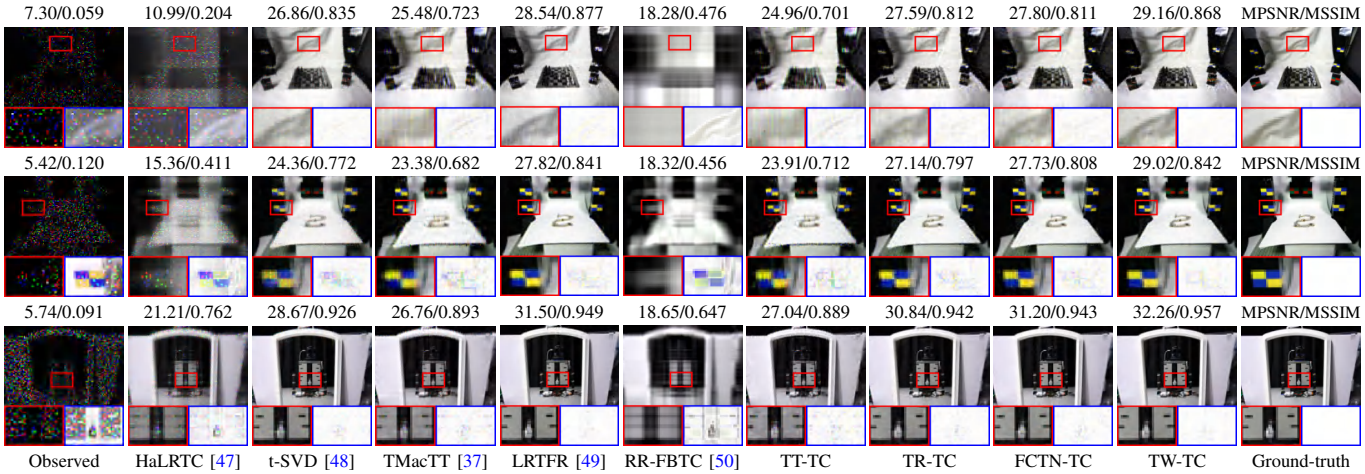


Fig. 12. Visualization of the recovered results on fifth-order LFIs. From top to bottom: the visual compositions and residual images for LFI *Chess* (SR $= 5\%$, $(4, 4)$-th grid), *Bracelet* (SR $= 10\%$, $(4, 4)$-th grid), and *Gantry Self Portrait* (SR $= 20\%$, $(4, 4)$-th grid), sequentially.

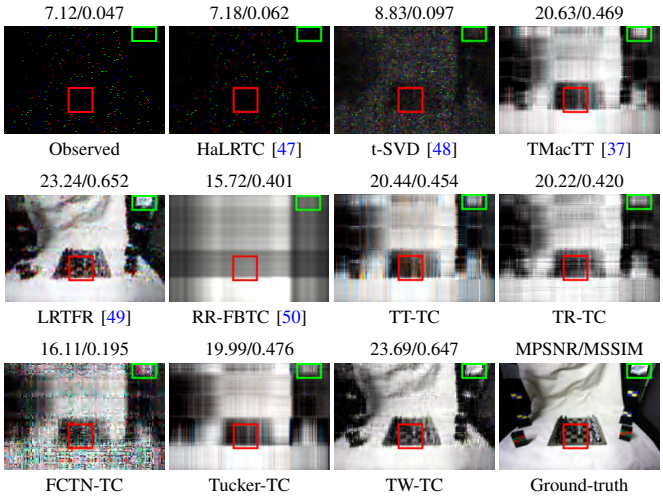


Fig. 13. Recovery performance of all competing methods on LFI *Chess* when SR $= 1\%$. The $(4, 4)$-th grid of all recovered images is adopted for visual presentation, and the notable regions are highlighted.

$\prod_{k=1}^{N} L_k$) of TW decomposition.

To clearly understand the model complexity, Fig. 7 provides the storage parameters of all models on both LTS and HTS with 5%, 10%, and 20% SRs. According to Fig. 7(a)-(d), the TW-TC model requires gradually fewer storage parameters than other models under varying SRs as the tensor order increases, thus validating that TW decomposition can avoid the potential impact of exponentially growing storage burden in practical tensor recovery applications.

More significantly, although the storage parameters of the TW-TC model exceed those of the Tucker-TC model for Nth-order ($N \leq 4$) cases, the TW-TC model is remarkably more computationally efficient than the Tucker-TC model, e.g., $10^{1.5} \sim 10^2$ times faster on both LTS and HTS, with SR being 20%. This results from the flexible TW-ranks allocation of TW decomposition for sets $\{R_k\}_{k=1}^N$ and $\{L_k\}_{k=1}^N$, as described in the computational complexity analysis.

3) Topology reliability: For tensor completion problem, TW decomposition is intrinsically utilized as the predictor or inferrer, whose topology uniquely determines the characterization or fitting capability. Instead of tensor compression, tensor completion obtains optimal performance only by employing the appropriate TN ranks, which reflect the actual effective TN structures.

Relying upon the optimal TW-ranks in synthetic data experiments, Table I statistics the actual topological structures of TW decomposition, including 7 wheel topologies and 17 subwheel topologies with few eliminations (i.e., few R_k/L_k being 1 in TW-ranks). According to Table I, the subwheel topologies do not include any of the TT, TR, and Tucker topologies, implying that the latter are not the optimal TN structures. That is why the TW-TC model always performs better than the TT-TC, TR-TC, and Tucker-TC models.

Moreover, Table I further provides the wheel-forced loss rate[5] (WFLR), which indicates the RSE loss rate of the TW-TC model when the complete-shape wheel topology is forcefully required (i.e., $R_k, L_k \geq 2$, $k = 1, 2, \cdots, N$). The lower WFLR means that TW topology can already achieve almost optimal performance without overly relying upon its subwheel topologies, which signifies the higher validity and anti-overfitting of TW topology. From Table I, the WFLRs of all subwheel topologies are lower than 4.50%, thus proving the rationality and reliability of TW topology.

B. Real-World Data Experiments

Furthermore, we investigate the TW-TC method on real-world data experiments by comparing it with several related methods, including HaLRTC [47], t-SVD [48], TMacTT [37], LRTFR [49], RR-FBTC [50], TT-TC (PAM), TR-TC (PAM), FCTN-TC (PAM), and Tucker-TC (PAM)[6]. All hyper-parameters involved in these competitors are optimally assigned or finely tuned within a specific range suggested by the reference papers. The real-world data mainly comprises three types, i.e., third-order multispectral images (MSIs), fourth-order videos, and fifth-order light field images (LFIs). Analogously, all experimental data are prenormalized into the inter-

[5]The WFLR is defined by $\frac{\text{Log}_{10}(\text{RSE}_1)-\text{Log}_{10}(\text{RSE}_2)}{\text{Log}_{10}(\text{RSE}_1)}$, where RSE_1 and RSE_2 are the optimal RSE values of the TW-TC model by imposing $R_k, L_k \geq 1$ and $R_k, L_k \geq 2$, $k = 1, 2, \cdots, N$, respectively.

[6]Since the Tucker-TC model is computationally consumptive, we only perform it on partial experiments under lower SRs, e.g., 0.1%.

val [0, 1]. The peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [51] metrics are employed for quantitative evaluation. When applied to Nth-order ($N \geq 3$) data, the means of PSNRs and SSIMs across the latter $N-2$ dimensions are measured, termed MPSNR and MSSIM, respectively. Regarding the TW-ranks, R_k and L_k, $k = 1, 2, \cdots, N$, are firstly initialized to 1 for all experiments. Subsequently, with the objective of maximizing the MPSNR, a block-wise grid search is performed sequentially and respectively on $\{R_k\}_{k=1}^{N}$ and $\{L_k\}_{k=1}^{N}$, yielding the final values of TW-ranks.

1) MSI data experiments: The tested MSI data consists of 31 MSIs sized $256 \times 256 \times 31$ (i.e., height $\times$ width $\times$ spectral), which are resized from the CAVE dataset[7]. The partially observed tensors are generated by random sampling with SR $= 0.1\%, 0.5\%, 1\%, 5\%, 10\%, 20\%$, and 40%. Numerically, Table II reports the average values of MPSNR and MSSIM for 31 MSIs under various SRs and the average computing costs (/s: seconds) of all compared methods. Moreover, Fig. 8 presents the visual performances of partial MSIs (i.e., *Toy*, *Flowers*, and *Feathers*) when SR $= 5\%$, 10%, and 20%. From the quantitative and qualitative results, we observe that the TW-TC model notably achieves higher MPSNR and MSSIM values and better visual reconstruction than the others, even the deep learning-based LRTFR method, under most SRs.

Considering the expensive computational burden of the Tucker-TC model, we further include the Tucker-TC model for comparison under only extremely low SRs (i.e., 0.1%, 0.5%, and 1%). From Fig. 9, the TW-TC model manifestly realizes a relatively finer recovery, while all other competing methods can hardly capture the inherent features especially visually invalid when SR $= 0.1\%$, confirming the effectiveness of TW decomposition for real-world third-order data.

2) Video data experiments: The tested video data contains 10 color videos[8] (CVs) and one hyperspectral video[9] (HSV) [52]. The former is of size $144 \times 176 \times 3 \times 20$ (i.e., height $\times$ width $\times$ spectral $\times$ frame), and the latter is of size $60 \times 94 \times 20 \times 20$ (i.e., height $\times$ width $\times$ spectral $\times$ frame). When tested on all CVs, we also empirically exclude the Tucker-TC model from the comparison for its unaffordable time consumption. Identically to the MSI experiments, the SRs in the CV experiments are designated as 0.1%, 0.5%, 1%, 5%, 10%, 20%, and 40%. The numerical results of all compared methods under all SR levels are tabulated in Table III. Also, the graphical results on CV *Akiyo*, *Hall*, and *Container*, with SR respectively being 5%, 10%, and 20%, are exhibited in Fig. 10. From the results, the TW-TC method invariably contributes the near-optimal metrics, particularly under lower SRs (e.g., $6 \sim 9$ dB gain of MPSNR compared to traditional TN decomposition methods when SR $= 0.1\%$), and the recovered images are closest to the ground truths over most other comparators.

Furthermore, we compare the TW-TC method with the other comparators (including the Tucker-TC) on HSV data under three SRs, i.e., 0.1%, 0.5%, and 1%. The experimental metrics are provided in Table IV, and the recovered images using SR $= 1\%$ are adopted in Fig. 11 for visual presentation. At any considered SR, the TW-TC method compares favorably to the TT-TC, TR-TC, FCTN-TC, and Tucker-TC models, which distinctly illustrates the remarkable superiority of the TW topology for fourth-order tensor recovery.

3) LFI data experiments: The LFI data experiments use 10 LFIs[10] of size $108 \times 162 \times 3 \times 5 \times 5$ (i.e., height $\times$ width $\times$ spectral $\times$ vertical grid $\times$ horizontal grid). All compared methods are conducted under 0.1%, 0.5%, 1%, 5%, 10%, and 20% SRs, but the Tucker-TC is selectively conducted only under 0.1%, 0.5%, and 1% SRs. The numerical and visual results are given in Table V and Figs. 12-13, respectively. For the fifth-order LFI recovery, the TW-TC steadily outshines most others, which is consistent with the performance in both third-order and fourth-order tensor experiments. This reveals the potential of the TW-TC model or, more internally, TW decomposition for higher-order tensor characterization.

According to the computational times reported in Fig. 9 and Tables II-V, the TW-TC typically delivers the comparable computational efficiency with the TT-TC, TR-TC, and FCTN-TC methods. Such a characteristic visibly distinguishes TW decomposition from Tucker decomposition, which allows TW decomposition to be workable under higher SRs and even higher-rank recovery, just like in synthetic data experiments, rather than highly computationally intractable.

VII. Conclusion

In this paper, we proposed a wheel topology-supported TN decomposition, i.e., TW decomposition, and analyzed its relevant theoretical properties. Afterwards, we presented two different learning algorithms, i.e., the non-iterative TW-SVD and iterative TW-ALS algorithms, to acquire the TW-format representation. The TW decomposition was verified, both theoretically and numerically, to have the potential to more accurately characterize the complex interactions inside high-order tensors using only linear hyper-parameter scaling, significantly contributing to the advancement of TNs in machine learning. In particular, TW decomposition conferred greater flexibility in rank allocation, which relatively alleviates the burdens of parameter storage and computation, thus promoting the applicability of TW decomposition. Furthermore, we provided a TW decomposition-based TC model, i.e., TW-TC, to investigate the validity of TW decomposition for high-order tensor recovery. Moreover, we correspondingly developed an efficient PAM-based solving algorithm, which enjoys a convergence guarantee. Substantial experiments on synthetic and real-world data confirmed that the TW-TC model markedly outperforms other state-of-the-art tensor decomposition-based methods, especially under extremely low SRs, provably demonstrating the reliability and superiority of TW decomposition.

References

[1] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, 2015.

[7] https://www.cs.columbia.edu/CAVE/databases/multispectral/

[8] https://media.xiph.org/video/derf/

[9] https://openremotesensing.net/knowledgebase/

[10] http://lightfield.stanford.edu/lfs.html

[2] A. Cichocki, R. Zdunek, A. H. Phan, and S. I. Amari, *Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation*, Wiley, 2009.

[3] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[4] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 925–938, 2019.

[5] Y. Zhou and Y. M. Cheung, "Bayesian low-tubal-rank robust tensor factorization with multi-rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 62–76, 2019.

[6] F. Zhang, J. Wang, W. Wang, and C. Xu, "Low-tubal-rank plus sparse tensor recovery with prior subspace information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3492–3507, 2020.

[7] X. Zhang and M. K. Ng, "Low rank tensor completion with Poisson observations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4239–4251, 2021.

[8] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models," *J. Mach. Learn. Res.*, vol. 15, pp. 2773–2832, 2014.

[9] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, 2017.

[10] W. Wang, Y. Sun, B. Eriksson, W. Wang, and V. Aggarwal, "Wide compression: Tensor ring nets," in *Proc. CVPR*, 2018, pp. 9329–9338.

[11] C. Llosa and R. Maitra, "Reduced-rank tensor-on-tensor regression and tensor-variate analysis of variance," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.

[12] T. Yokota, K. Kawai, M. Sakata, Y. Kimura, and H. Hontani, "Dynamic PET image reconstruction using nonnegative matrix factorization incorporated with deep image prior," in *Proc. ICCV*, 2019, pp. 3126–3135.

[13] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, 2013.

[14] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Math. Phys.*, vol. 6, no. 1-4, pp. 164–189, 1927.

[15] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[16] F. Verstraete, V. Murg, and J. I. Cirac, "Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems," *Adv. Phys.*, vol. 57, no. 2, pp. 143–224, 2008.

[17] L. Grasedyck, "Hierarchical singular value decomposition of tensors," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 4, pp. 2029–2054, 2010.

[18] T. Huckle, K. Waldherr, and T. Schulte-Herbrüggen, "Computations in quantum tensor networks," *Linear Alg. Appl.*, vol. 438, no. 2, pp. 750–781, 2013.

[19] M. Hashemizadeh, M. Liu, J. Miller, and G. Rabusseau, "Adaptive tensor learning with tensor networks," in *Proc. NeurIPS 1st Workshop on Quantum Tensor Networks in Machine Learning*, 2020.

[20] U. Schollwöck, "The density-matrix renormalization group in the age of matrix product states," *Ann. Phys.*, vol. 326, no. 1, pp. 96–192, 2011.

[21] B. Khavari and G. Rabusseau, "Lower and upper bounds on the pseudo-dimension of tensor network models," in *Proc. NeurIPS*, 2021, vol. 34.

[22] L. Grasedyck, M. Kluge, and S. Kramer, "Variants of alternating least squares tensor completion in the tensor train format," *SIAM J. Sci. Comput.*, vol. 37, no. 5, pp. A2424–A2450, 2015.

[23] W. Wang, V. Aggarwal, and S. Aeron, "Efficient low rank tensor ring completion," in *Proc. ICCV*, 2017, pp. 5697–5705.

[24] L. Yuan, C. Li, D. Mandic, J. Cao, and Q. Zhao, "Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion," in *Proc. AAAI*, 2019, pp. 9151–9158.

[25] C. Y. Ko, K. Batselier, L. Daniel, W. Yu, and N. Wong, "Fast and accurate tensor completion with total variation regularized tensor trains," *IEEE Trans. on Image Process.*, vol. 29, pp. 6918–6931, 2020.

[26] H. Huang, Y. Liu, Z. Long, and C. Zhu, "Robust low-rank tensor ring completion," *IEEE Trans. Comput. Imaging*, vol. 6, pp. 1117–1126, 2020.

[27] Y. Qiu, G. Zhou, Z. Huang, Q. Zhao, and S. Xie, "Efficient tensor robust PCA under hybrid model of Tucker and tensor train," *IEEE Signal Process. Lett.*, vol. 29, pp. 627–631, 2022.

[28] I. V. Oseledets, "Tensor train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011.

[29] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, "Tensor ring decomposition," *arXiv Preprint:1606.05535*, 2016.

[30] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac, "Matrix product state representations," *Quantum Inform. Comput.*, vol. 7, no. 5-6, pp. 401–430, 2007.

[31] R. Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states," *Ann. Phys.*, vol. 349, pp. 117–158, 2014.

[32] Y. B. Zheng, T. Z. Huang, X. L. Zhao, Q Zhao, and T. X. Jiang, "Fully-connected tensor network decomposition and its application to higher-order tensor completion," in *Proc. AAAI*, 2021, pp. 11071–11078.

[33] Y. Yang, D. Krompass, and V. Tresp, "Tensor-train recurrent neural networks for video classification," in *Proc. ICML*, 2017, pp. 3891–3900.

[34] Y. Chen, T. Z. Huang, W. He, N. Yokoya, and X. L. Zhao, "Hyperspectral image compressive sensing reconstruction using subspace-based nonlocal tensor ring decomposition," *IEEE Trans. on Image Process.*, vol. 29, pp. 6813–6828, 2020.

[35] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Probl.*, vol. 27, no. 2, pp. 025010, 2011.

[36] Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, and S. I. Amari, "Bayesian robust tensor factorization for incomplete multiway data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 736–748, 2016.

[37] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do, "Efficient tensor completion for color image and video recovery: Low-rank tensor train," *IEEE Trans. on Image Process.*, vol. 26, no. 5, pp. 2466–2479, 2017.

[38] W. He, Y. Chen, N. Yokoya, C. Li, and Q. Zhao, "Hyperspectral super-resolution via coupled tensor ring factorization," *Pattern Recognit.*, vol. 122, pp. 108–280, 2022.

[39] Z. C. Wu, T. Z. Huang, L. J. Deng, H. X. Dou, and D. Meng, "Tensor wheel decomposition and its tensor completion application," in *Proc. NeurIPS*, 2022.

[40] P. Yang, Y. Qiu, Z. Huang, G. Zhou, and Q. Zhao, "Efficient and compact tensor wheel decomposition for tensor completion," *Pattern Recognit.*, vol. 172, pp. 112377, 2026.

[41] X. Tian, K. Xie, J. Wen, and Q. Feng, "Tensor wheel completion with low-rank factor prior and adaptive graph regularizer for hyperspectral image recovery," *IEEE Trans. Multimedia*, vol. 28, pp. 1174–1187, 2026.

[42] M. Wang, Y. Yu, and H. Li, "Randomized tensor wheel decomposition," *SIAM J. Sci. Comput.*, vol. 46, no. 3, pp. A1714–A1746, 2024.

[43] R. Zdunek, "Pair-mode tensor wheel decomposition with randomized block Krylov iteration," *Neurocomputing*, vol. 650, pp. 130820, 2025.

[44] X. Tian, K. Xie, and J. Wen, "Tensor wheel completion with parallel matrix factorization and group smoothness for hyperspectral image recovery," *Pattern Recognit.*, vol. 172, pp. 112508, 2026.

[45] H. Attouch, J. Bolte, and 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.*, vol. 137, no. 1, pp. 91–129, 2013.

[46] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Math. Program.*, vol. 146, no. 1, pp. 459–494, 2014.

[47] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, 2013.

[48] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, 2017.

[49] Y. Luo, X. Zhao, Z. Li, M. K. Ng, and D. Meng, "Low-rank tensor function representation for multi-dimensional data recovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 3351–3369, 2024.

[50] S. Li, S. Fang, L. Cheng, F. Yin, Y. Wu, P. Gerstoft, and S. Theodoridis, "When Bayesian tensor completion meets multioutput Gaussian processes: Functional universality and rank learning," *IEEE Trans. Signal Process.*, vol. 73, pp. 5319–5335, 2025.

[51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[52] A. Mian and R. Hartley, "Hyperspectral video restoration using optical flow and sparse coding," *Opt. Express*, vol. 20, no. 10, pp. 10658–10673, 2012.