

# Nonconvex Robust High-Order Tensor Completion Using Randomized Low-Rank Approximation

Wenjin Qin<sup>ID</sup>, Hailin Wang<sup>ID</sup>, Student Member, IEEE, Feng Zhang<sup>ID</sup>, Weijun Ma<sup>ID</sup>, Jianjun Wang<sup>ID</sup>, Member, IEEE, and Tingwen Huang<sup>ID</sup>, Fellow, IEEE

**Abstract**—Within the *tensor singular value decomposition* (T-SVD) framework, existing robust low-rank tensor completion approaches have made great achievements in various areas of science and engineering. Nevertheless, these methods involve the T-SVD based low-rank approximation, which suffers from high computational costs when dealing with large-scale tensor data. Moreover, most of them are only applicable to third-order tensors. Against these issues, in this article, two efficient low-rank tensor approximation approaches fusing random projection techniques are first devised under the order- $d$  ( $d \geq 3$ ) T-SVD framework. Theoretical results on error bounds for the proposed randomized algorithms are provided. On this basis, we then further investigate the robust high-order tensor completion problem, in which a double nonconvex model along with its corresponding fast optimization algorithms with convergence guarantees are developed. Experimental results on large-scale synthetic and real tensor data illustrate that the proposed method outperforms other state-of-the-art approaches in terms of both computational efficiency and estimated precision.

**Index Terms**—High-order T-SVD framework, robust high-order tensor completion, randomized low-rank tensor approximation, nonconvex regularizers, ADMM algorithm.

## I. INTRODUCTION

MULTIDIMENSIONAL data including medical images, remote sensing images, light field images, color videos, and beyond, are becoming increasingly prevailing in various domains such as neuroscience [1], chemometrics [2], data

Manuscript received 17 May 2023; revised 10 December 2023, 29 February 2024, and 21 March 2024; accepted 29 March 2024. Date of publication 10 April 2024; date of current version 16 April 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFA1008500; in part by the Joint Funds of the Natural Science Innovation-Driven Development of Chongqing, China, under Grant 2023NSCQ-LZX0218; in part by the Fundamental Research Funds for the Central Universities under Grant SWU120078; in part by the National Natural Science Foundation of China under Grant 12071380, Grant 12101512, and Grant 62063028; in part by the Chongqing Talent Project, China, under Grant cstc2021ycjh-bgzxm0015; in part by the Natural Science Foundation of Ningxia under Grant 2023AAC03017; and in part by the China Postdoctoral Science Foundation under Grant 2021M692681. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bart Goossens. (Wenjin Qin and Hailin Wang contributed equally to this work.) (Corresponding author: Jianjun Wang.)

Wenjin Qin and Feng Zhang are with the School of Mathematics and Statistics, Southwest University, Chongqing 400715, China (e-mail: qinwenjin2021@163.com; zfmath@swu.edu.cn).

Hailin Wang is with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: wanghailin97@163.com).

Weijun Ma is with the School of Information Engineering, Ningxia University, Yinchuan 750021, China (e-mail: Weijunma\_2008@sina.com).

Jianjun Wang is with the School of Mathematics and Statistics, and the Research Institute of Intelligent Finance and Digital Economics, Southwest University, Chongqing 400715, China (e-mail: wjj@swu.edu.cn).

Tingwen Huang is with the Department of Mathematics, Texas A&M University at Qatar, Doha, Qatar (e-mail: tingwen.huang@qatar.tamu.edu).

Digital Object Identifier 10.1109/TIP.2024.3385284

mining [3], machine learning [4], image processing [5], [6], [7], and computer vision [8], [9], [10]. Compared to vectors and matrices, tensors possess a more powerful capability to characterize the inherent structural information underlying these data from a higher-order perspective. Nevertheless, due to various factors such as occlusions, abnormalities, software glitches, and sensor failures, the tensorial data faced in practical applications often suffer from elements loss and noise/outliers corruption. Hence, *robust low-rank tensor completion* (RLRTC) has been widely concerned by a large number of scholars [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31].

RLRTC belongs to a canonical inverse problem, which aims to reconstruct the underlying low-rank tensor from partial observations of target tensor corrupted by noise/outliers. Mathematically, the RLRTC model can be formulated as follows:

$$\min_{\mathcal{L}, \mathcal{E}} \Psi(\mathcal{L}) + \lambda \Upsilon(\mathcal{E}), \quad \text{s.t. } \mathbf{P}_\Omega(\mathcal{L} + \mathcal{E}) = \mathbf{P}_\Omega(\mathcal{M}), \quad (1)$$

where  $\Psi(\mathcal{L})$  represents the regularizer measuring tensor low-rankness (also called the relaxation term of the tensor rank),  $\Upsilon(\mathcal{E})$ <sup>1</sup> denotes the noise/outliers regularization,  $\lambda > 0$  is a trade-off parameter that balances these two terms,  $\mathcal{M}$  is the observed tensor, and  $\mathbf{P}_\Omega(\cdot)$  is the projection operator onto the observed index set  $\Omega$  such that

$$(\mathbf{P}_\Omega(\mathcal{M}))_{i_1, \dots, i_d} = \begin{cases} \mathcal{M}_{i_1, \dots, i_d}, & \text{if } (i_1, \dots, i_d) \in \Omega, \\ 0, & \text{otherwise.} \end{cases}$$

If the index set  $\Omega$  is the whole set, i.e., all elements are observed, then the model (1) reduces to the *Tensor Robust Principal Component Analysis* (TRPCA) problem [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42]. If there is no corruption, i.e.,  $\mathcal{E} = 0$ , then the model (1) is equivalent to the *Low-Rank Tensor Completion* (LRTC) problem [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53]. Therefore, RLRTC can be viewed as a generalized form of both LRTC and TRPCA.

Nevertheless, there exist different definitions of tensor rank and its corresponding relaxation term within different tensor factorization frameworks, which makes the optimization problem (1) extremely complicated. The commonly-used frameworks contain *CANDECOMP/PARAFAC* (CP) [54], *Tucker* [55], *tensor train* (TT) [56], *tensor ring* (TR) [57], and *tensor singular value decomposition* (T-SVD) [58], [59]. Among them, T-SVD presents the first closed multiplication operation called *tensor-tensor product* (t-product), and derives a novel tensor tubal rank [60] that well characterizes the

<sup>1</sup>In specific problems, if we assume that the noise/outliers follows the Laplacian distribution or the Gaussian distribution, then  $\Upsilon(\mathcal{E})$  can be chosen as  $\|\mathcal{E}\|_1$  or  $\|\mathcal{E}\|_F^2$ , respectively.

intrinsic low-rank structure of a tensor. In particular, the recent work [61] revolutionarily proved the best representation and compression theories of T-SVD, making it more notable in capturing the “spatial-shifting” correlation and the global structure information underlying tensors. With these advantages, the robust low-tubal-rank tensor completion [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31] modeled by (1) and its variants [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47] have recently caught many scholars’ attention. However, we observe that these approaches are only relevant to third-order tensors. To fix this problem, Qin et al. established an order- $d$  ( $d \geq 3$ ) T-SVD algebraic framework [62] based on a family of invertible linear transforms, and then preliminary investigated the model, algorithm, and theory for the *robust high-order tensor completion* (RHTC) [62], [63], [64], [65]. The RHTC methods generated by other tensor factorization frameworks can be found in [11], [12], [13], [14], [15], [16], [17], [18], [19], and [20].

Although the above deterministic RLRTC investigations have already made some achievements in real-world applications, they encounter enormous challenges in dealing with those tensorial data characterized by large volumes, multiple modes, high dimensions, etc. This is because they require to perform multiple low-rank approximations based on a specific tensor factorization. Calculating such an approximation generally involves multiple *singular value decompositions* (SVDs) or *alternating least squares* (ALS) problems, which is very time-consuming and inefficient when the data size scales up. Motivated by the fact that randomized algorithms can accelerate the computational speed of their conventional counterparts at the expense of slight accuracy loss (e.g., [66], [67], [68], [69], [70]), many effective methods for *low-rank tensor approximation* (LRTA) which leverage randomized sketching techniques have attracted a great deal of attention in recent years (e.g., [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89]). Among these methods, we obviously find that the ones based on T-SVD framework are only appropriate for third-order tensors. In addition, existing RHTC researches combined with randomized low-rank approximation scheme are relatively lacking. With the rapid development of information technologies, such as Internet of Things, and big data, large-scale high-order tensors encountered in real scenarios are growing explosively, like order-four *color videos* (CVs) and *multi-temporal remote sensing images* (MRSIs), order-five *light field images* (LFIs), order-six *bidirectional texture function images* (BTFIs). Therefore, under the T-SVD framework, it is particularly important to develop fast and efficient large-scale high-order tensor representation and recovery methods in virtue of randomized techniques.

### A. Our Contributions

Main contributions of this work are summarized as follows:

1) Firstly, within the order- $d$  T-SVD algebraic framework [62], two efficient randomized algorithms for low-rank approximation of high-order tensor are devised considering their adaptability to large-scale tensor data. Theoretical results on error bounds for the proposed approximation algorithms are provided. Empirical studies show that the developed approximation methods obtain a significant advantage in computational speed against the optimal  $k$ -term

approximation [62] (also called truncated T-SVD) with a slight loss of precision.

2) Secondly, an effective and scalable model for RHTC is proposed in virtue of novel nonconvex low-rank and noise/outliers regularizers. Based on the proposed randomized low-rank approximation schemes, we then design two fast algorithms with convergence guarantees to solve the formulated model via *alternating direction method of multipliers* (ADMM) framework, through which any low T-SVD rank high-order tensors with simultaneous elements loss and noise/outliers corruption can be reconstructed efficiently and accurately.

3) Thirdly, our proposed RHTC algorithm can be applied to a series of large-scale reconstruction tasks, such as the restoration of fourth-order CVs and MRSIs, and fifth-order LFIs. Experimental results demonstrate that the proposed method achieves competitive performance in estimation accuracy and CPU running time than other state-of-the-art ones. Strikingly, in the case of sacrificing a little precision, our versions combined with randomization ideas decrease the CPU time by about 50%~70% compared with the deterministic version.

### B. Organization

The remainder of the paper is organized as follows. Section II gives a brief summary of related work. The main notations and preliminaries are introduced in Section III, and then we develop two efficient randomized algorithms for low-rank approximation of high-order tensor in Section IV. Section V proposes effective nonconvex model and optimization algorithm for RHTC. In Section VI, extensive experiments are conducted to evaluate the effectiveness of the proposed method. Finally, we conclude our work in Section VII. **What is particularly noteworthy is that all theoretical proofs, related algorithms, and more experimental results of this article are provided in the supplementary material.**

## II. RELATED WORK

Considering that this work investigates the RLRTC problem leveraging fast randomized LRTA strategies, in this section, we first review some representative randomized LRTA methods and then summarize the currently typical approaches for RLRTC problem. Finally, we clearly discuss the distinctions between this work and previous researches.

### A. An Overview of Randomized LRTA Methods

In recent years, several popular sketching techniques (especially random sampling, random projection) have been utilized to develop fast randomized LRTA methods under various factorization schemes. Currently, the representative methods for randomized LRTA can be broadly summarized as follows.

1) *ALS-Based LRTA Methods*: To accelerate the standard *alternating least squares algorithm for CP decomposition* (CP-ALS [54]), Battaglino et al. [72] investigated a practical randomized method through efficient sampling technique and fast Johnson-Lindenstrauss transform. Soon after, Larsen and Kolda [73] proposed applying leverage score-based sketching to the overdetermined least squares problem in CP-ALS. To reduce the computational cost of Tucker-ALS method [54], Malik and Becker [74] designed two randomized algorithms

for fast low-rank Tucker decomposition which incorporate TensorSketch technique. Moreover, Ma and Solomonik [75] proposed fast and accurate sketched ALS algorithms for Tucker decomposition, which mainly solves rank-constrained linear least squares subproblems via TensorSketch and leverage score sampling schemes. In order to avoid forming the large coefficient matrices in TR-ALS problem [57], Malik and Becker [81] provided a fast randomized method based on leverage scores sampling. Furthermore, Yu and Li [82] applied the *Kronecker sub-sampled randomized Fourier transform* (KSRFT) and TensorSketch to the TR-ALS problems to devise more efficient randomized algorithms. With respect to TT framework, a novel randomized proximal ALS algorithm was proposed for low-rank TT decomposition by using TensorSketch [84].

2) *SVD-Based LRTA Methods*: In regard to Tucker framework, combining the strategies of random projection and power iteration, some adaptive randomized versions for the *higher-order SVD* (HOSVD) [54] and *sequentially truncated higher-order SVD* (ST-HOSVD) [90] were designed in [76], [77], and [78]. As for the TR framework, Yuan et al. [79] proposed effective low-rank TR approximation algorithms by using randomized full-mode projection method. Other randomized algorithms related to this framework were summarized by Ahmadi-Asl et al. [80]. With respect to TT framework, Che and Wei [83] put forward an adaptive randomized low-rank TT approximation method. To take advantage of parallelization, Shi et al. [85] proposed several parallelizable sketching algorithms for the computation of low rank TT approximation. Daas et al. proposed parallel algorithms for TT arithmetic [86], and then developed novel randomized algorithms for rounding TT-tensors [87]. Within the T-SVD framework, Zhang et al. [88] proposed an effective randomized algorithm for the low-tubal-rank tensor approximation via Gaussian random projection scheme. In parallel, Tarzanagh et al. [89] developed randomized algorithms for commonly-used tensor operations including LRTA, which sample a small number of lateral and/or horizontal slices of the underlying tensors. However, these approaches are only appropriate for third-order tensors.

## B. An Overview of RLRTC Methods

1) *RLRTC Based on T-SVD Factorization*: Lu et al. [32] rigorously deduced a novel tensor nuclear norm (TNN) corresponding to T-SVD that is proved to be the convex envelope of the tensor average rank. Sequentially, Jiang and Ng [21] conducted a rigorous study for the RLRTC problem, which is modeled by the TNN and  $\ell_1$ -norm penalty terms. Besides, Wang et al. also adopted this novel TNN or slice-weighted TNN plus a sparsity measure inducing  $\ell_1$ -norm to develop the RLRTC methods [22], [23]. Theoretically, the deterministic and non-asymptotic upper bounds on the estimation error are established from a statistical standpoint. However, the previous methods may suffer from disadvantage due to the limitation of Fourier transform [29]. Aiming at this issue, by utilizing the generalized transformed TNN (TTNN) and  $\ell_1$ -norm regularizers, Song et al. [24] proposed an unitary transform method for RLRTC and also analyzed its recovery guarantee. Continuing along this vein, a patched-tubes unitary transform approach for RLRTC was proposed by Ng et al. [25]. Nevertheless, the TTNN is a loose approximation of the tensor tubal rank, which usually leads to the over-penalization of the optimization problem and hence causes some unavoidable biases in real

applications. In addition, as indicated by [91], the  $\ell_1$ -norm might not be statistically optimal in more challenging scenarios. Recently, to break the shortcomings existing in the TNN and  $\ell_1$ -norm regularization terms, some researchers [28], [29], [30], [31] designed new nonconvex low-rank and noise/outliers regularization terms to study the RLRTC problem from the model, algorithm, and theory. But these methods are only limited to the third-order tensor case and face the computation of multiple SVDs of large-sized frontal slices upon processing large-scale tensor data.

### 2) *RLRTC Based on Other Factorization Schemes*:

Liu et al. [49] primitively developed a new Tucker nuclear norm, i.e., *Sum-of-Nuclear-Norms of unfolding matrices of a tensor* (SNN), as the convex relaxation of the tensor tucker rank. Then, relevant RLRTC approaches within the Tucker format was investigated in [11] and [12] via combining the SNN regularization with  $\ell_1$ -norm loss function. Zhao et al. [13] proposed a variational Bayesian inference framework for CP rank determination and applied it to the RLRTC problem. Within the TT factorization, Bengua et al. [50] proposed a novel TT nuclear norm as the convex surrogate of the TT rank. Furthermore, in virtue of an auto-weighted mechanism, Chen et al. [16] studied a new RLRTC method modeled by the TT nuclear norm and  $\ell_1$ -norm regularizers. Under the TR decomposition, by utilizing the TR nuclear norm and  $\ell_1$ -norm regularizers, the model, algorithm, and theoretical analysis for RLRTC were developed by Huang et al. [15]. To be more robust against both missing entries and noise/outliers, an effective iterative  $\ell_p$ -regression ( $0 < p < 2$ ) TT completion method was developed in [17]. Subsequently, Li and So [18] suggested replacing the  $\ell_p$ -norm with a novel  $\ell_{p,\epsilon}$ -norm ( $0 < p \leq 1$ ), and then integrated it with TR factorization to develop a new RLRTC approach. He and Atia [20] further developed a two-stage RLRTC approach through a novel coarse-to-fine framework, which employs a global coarse completion result to guide a local patch refinement process. Recently, in combination with TR decomposition and a novel capped Frobenius norm, Li et al. [19] devised a novel RLRTC method without the need to tune any parameter. However, these deterministic methods experience the calculation of multiple SVDs of large-sized unfolding matrices (or multiple large-sized ALS problems) when dealing with large-scale high-order tensor data.

## C. The Distinctions of Our Work and Previous Researches

Note that this article can be regarded as an expanded version of our conference paper [63]. Built off the conference version, this paper makes the following changes: 1) another high-order LRTA algorithm that fuses the randomized blocked strategy is added; 2) the original low-rank and noise/outliers regularizers are further enhanced with more flexible regularizers; 3) two accelerated algorithms for solving the newly formulated non-convex model are designed via the proposed LRTA strategies; 4) a large number of experiments concerning with high-order tensor approximation and restoration are added. Below, we discuss the distinctions between our work and other researches.

1) *The Differences in Model Construction*: This paper first proposes a new nonconvex low-rank regularizer (see Definition 1), and then combines it with nonconvex noise/outlier regularizer to study the RLRTC problem. Comparing with the previous low-rank regularizer, e.g., HTNN [62] that treats the different rank components equally and thus results in the suboptimality of the obtained solution to the original problem,

TABLE I  
THE MAIN NOTIONS AND PRELIMINARIES FOR ORDER- $d$  TENSOR

Notations	Descriptions	Notations	Descriptions
$\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$	matrix	$\mathbf{I}_n \in \mathbb{R}^{n \times n}$	$n \times n$ identity matrix
trace( $\mathbf{A}$ )	matrix trace	$\mathbf{A}^H(\mathbf{A}^T)$	conjugate transpose (transpose)
$\langle \mathbf{A}, \mathbf{B} \rangle = \text{trace}(\mathbf{A}^H \cdot \mathbf{B})$	matrix inner product	$\ \mathbf{A}\ _{w,S_p} = (\sum_i w_i  \sigma_i(\mathbf{A}) ^p)^{\frac{1}{p}}$	matrix weighted Schatten- $p$ norm
$\mathbf{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$	order- $d$ tensor	$\mathbf{A}_{i_1 \cdots i_d}$ or $\mathcal{A}(i_1, \dots, i_d)$	$(i_1, \dots, i_d)$ -th entry
$\mathbf{A}_{(k)} \in \mathbb{R}^{n_k \times \prod_{j \neq k} n_j}$	mode- $k$ unfolding of $\mathbf{A}$	$\ \mathbf{A}\ _\infty = \max_{i_1 \cdots i_d}  \mathbf{A}_{i_1 \cdots i_d} $	tensor infinity norm
$\ \mathbf{A}\ _{W,\ell_q} = (\sum w_i  \mathbf{A}_{i_1 \cdots i_d} ^q)^{\frac{1}{q}}$	tensor weighted $\ell_q$ -norm	$\ \mathbf{A}\ _F = (\sum_{i_1 \cdots i_d}  \mathbf{A}_{i_1 \cdots i_d} ^2)^{\frac{1}{2}}$	tensor Frobenius norm
$\mathcal{C} = \mathbf{A} *_{\mathcal{L}} \mathbf{B}$	order- $d$ t-product under linear transform $\mathcal{L}$	$\mathbf{A}^H(\mathbf{A}^T)$	conjugate transpose (transpose)
$\langle \mathbf{A}, \mathbf{B} \rangle$	the inner product between order- $d$ tensors $\mathbf{A}$ and $\mathbf{B}$ , i.e., $\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{j=1}^{n_3 n_4 \cdots n_d} \langle \mathbf{A}^{<j>}, \mathbf{B}^{<j>} \rangle$ .		
$\mathbf{A}^{<j>} \in \mathbb{R}^{n_1 \times n_2}$	the matrix frontal slice of $\mathbf{A}$ , $\mathbf{A}^{<j>} = \mathbf{A}(:, :, i_3, \dots, i_d)$ , $j = \sum_{a=4}^d (i_a - 1) \Pi_{b=3}^{a-1} n_b + i_3$ .		
$\mathbf{A} \times_n \mathbf{M}$	the mode- $n$ product of tensor $\mathbf{A}$ with matrix $\mathbf{M}$ , $\mathbf{B} = \mathbf{A} \times_n \mathbf{M} \iff \mathbf{B}_{(n)} = \mathbf{M} \cdot \mathbf{A}_{(n)}$ .		
bdiag( $\mathbf{A}$ ) $\in \mathbb{R}^{n_1 n_3 \cdots n_d \times n_2 n_3 \cdots n_d}$	bdiag( $\mathbf{A}$ ) is a block diagonal matrix whose $i$ -th block equals to $\mathbf{A}^{<i>}$ , $\forall i \in \{1, 2, \dots, n_3 \cdots n_d\}$ .		
f-diagonal/f-upper triangular tensor $\mathbf{A}$	frontal slice $\mathbf{A}^{<j>}$ of $\mathbf{A}$ is a diagonal matrix (an upper triangular matrix), $\forall j \in \{1, 2, \dots, n_3 \cdots n_d\}$ .		
identity tensor $\mathbf{I} \in \mathbb{R}^{n \times n \times n_3 \times \cdots \times n_d}$	identity tensor $\mathbf{I}$ is defined to be a tensor such that $\mathcal{L}(\mathbf{I})^{<j>} = \mathbf{I}_n$ , $\forall j \in \{1, 2, \dots, n_3 \cdots n_d\}$ .		
Gaussian random tensor $\mathbf{G}$	the entries of $\mathcal{L}(\mathbf{G})^{<j>}$ follow the standard normal distribution, $\forall j \in \{1, 2, \dots, n_3 \cdots n_d\}$ .		
orthogonal tensor $\mathbf{Q}$	orthogonal tensor satisfies: $\mathbf{Q}^T *_{\mathcal{L}} \mathbf{Q} = \mathbf{I}$ , while partially orthogonal tensor satisfies: $\mathbf{Q}^T *_{\mathcal{L}} \mathbf{Q} = \mathbf{I}$ .		
H-TSVD( $\mathbf{A}, \mathcal{L}$ )	order- $d$ T-SVD factorization, i.e., $\mathbf{A} = \mathbf{U} *_{\mathcal{L}} \mathbf{S} *_{\mathcal{L}} \mathbf{V}^T$ , where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal, $\mathbf{S}$ is f-diagonal.		
H-TQR( $\mathbf{A}, \mathcal{L}$ )	order- $d$ tensor QR-type factorization, i.e., $\mathbf{A} = \mathbf{Q} *_{\mathcal{L}} \mathbf{R}$ , where $\mathbf{Q}$ is orthogonal while $\mathbf{R}$ is f-upper triangular.		
rank <sub>tsvd</sub> ( $\mathbf{A}$ )	rank <sub>tsvd</sub> ( $\mathbf{A}$ ) = $\sum_i \mathbf{1}[\mathbf{S}(i, i, \dots, i) \neq 0]$ , where $\mathbf{S}$ originates from the middle component of $\mathbf{A} = \mathbf{U} *_{\mathcal{L}} \mathbf{S} *_{\mathcal{L}} \mathbf{V}^T$ .		

the proposed one is tighter and more feasible, which assigns different weight values to different singular components in the transform domain. Although other novel nonconvex regularization terms have been proposed in existing RLRTC studies [28], [29], [30], [31], they can only be applied to third-order tensors.

*2) The Differences in Algorithm Design:* In previous work [62], we proposed utilizing T-SVT operator to estimate the low-rank component in high-order LRTC model. However, the T-SVT operator shrinks all the singular values in the transform domain with the same threshold (i.e., it does not consider the importance of different singular components), and hence causes some unavoidable biases in real applications. In addition, the calculation of T-SVT operator involves high-order T-SVD, which faces high computational costs of multiple SVDs in the transform domain when dealing with large-scale tensor data. Similarly, existing RLRTC algorithms [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31] also suffer from this computational bottlenecks. Aiming at addressing these critical issues, in this manuscript, we have made the following improvements: 1) Two efficient LRTA approaches (i.e., randQB and FPB-randQB in the supplementary material) fusing random projection techniques are devised under the order- $d$  T-SVD framework. On this basis, we put forward two effective randomized algorithms for calculating the high-order T-SVD (see Section IV); 2) Furthermore, we put forward a fast randomized GTSVT operator (see Section V—Subsection V-B) to estimate the low-rank component in the RHTC model, which automatically updates the weight values in virtue of reweighting strategy and shrinks the singular values in the transform domain to different degrees.

### III. NOTATIONS AND PRELIMINARIES

For brevity, the main notations and preliminaries utilized in the whole paper are summarized in Table I, most of which originate from the literature [62].

In this work, we let  $\mathcal{L}(\mathbf{A})$  represent the result of invertible linear transforms  $\mathcal{L}$  on  $\mathbf{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ , i.e.,

$$\mathcal{L}(\mathbf{A}) = \mathbf{A} \times_3 \mathbf{U}_{n_3} \times_4 \mathbf{U}_{n_4} \cdots \times_d \mathbf{U}_{n_d}, \quad (2)$$

where the transform matrices  $\mathbf{U}_{n_i} \in \mathbb{C}^{n_i \times n_i}$  of  $\mathcal{L}$  satisfies:

$$\mathbf{U}_{n_i} \cdot \mathbf{U}_{n_i}^H = \mathbf{U}_{n_i}^H \cdot \mathbf{U}_{n_i} = \alpha_i \mathbf{I}_{n_i}, \forall i \in \{3, \dots, d\}, \quad (3)$$

in which  $\alpha_i > 0$  is a constant. The inverse operator of  $\mathcal{L}(\mathbf{A})$  is defined as  $\mathcal{L}^{-1}(\mathbf{A}) = \mathbf{A} \times_d \mathbf{U}_{n_d}^{-1} \times_{d-1} \mathbf{U}_{n_{d-1}}^{-1} \cdots \times_3 \mathbf{U}_{n_3}^{-1}$ , and  $\mathcal{L}^{-1}(\mathcal{L}(\mathbf{A})) = \mathbf{A}$ .

*Definition 1 (Order- $d$  WTSN):* Let  $\mathcal{L}$  be any invertible linear transform in (2) and it satisfies (3),  $\mathbf{S}$  be from the middle component of  $\mathbf{A} = \mathbf{U} *_{\mathcal{L}} \mathbf{S} *_{\mathcal{L}} \mathbf{V}^T$ . Then, the weighted tensor Schatten- $p$  norm (WTSN) of  $\mathbf{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$  is defined as

$$\begin{aligned} \|\mathbf{A}\|_{W,S_p} &:= \left( \frac{1}{\rho} \| \text{bdiag}(\mathcal{L}(\mathbf{A})) \|_{w,S_p}^p \right)^{1/p} \\ &= \left( \frac{1}{\rho} \sum_{j=1}^{n_3 n_4 \cdots n_d} \| \mathcal{L}(\mathbf{A})^{<j>} \|_{w^{(j)},S_p}^p \right)^{1/p} \\ &= \left( \frac{1}{\rho} \sum_{j=1}^{n_3 n_4 \cdots n_d} \text{trace}(\mathbf{W}^{<j>} \cdot |\mathcal{L}(\mathbf{S})^{<j>}|^p) \right)^{1/p}, \end{aligned}$$

where  $\mathbf{W}$  is a nonnegative order- $d$  f-diagonal weighting tensor,  $\mathbf{w} = \text{diag}(\text{bdiag}(\mathbf{W}))$ ,  $\mathbf{w}^{(j)} = \text{diag}(\mathbf{W}^{<j>})$ , and  $\rho = \alpha_3 \cdots \alpha_d > 0$  is a constant determined by  $\mathcal{L}$ .

*Remark 1:* The high-order WTSN (HWTSN) assigns different weight values to different singular values in the transform domain: the larger one is multiplied by a smaller weight while the smaller one is multiplied by a larger weight. That is, the weight values should be inversely proportional to the singular values in the transform domain. In particular, the HWTSN I is equivalent to the high-order tensor Schatten- $p$  norm (HTSN) when weighting is not taken into account, 2) reduces to the high-order weighted TNN (HWTNN) [64] when  $p = 1$ , and 3) simplifies to the high-order TNN (HTNN) [62] when  $p = 1$ , and  $\mathbf{W}$  is not considered.

*Theorem 1 (Optimal  $k$ -term approximation [62]):* Suppose that the T-SVD of  $\mathbf{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$  is  $\mathbf{A} = \mathbf{U} *_{\mathcal{L}} \mathbf{S} *_{\mathcal{L}} \mathbf{V}^T$  and define  $\mathbf{A}_k = \sum_{i=1}^k \mathbf{U}(:, i, :, \dots, :) *_{\mathcal{L}} \mathbf{S}(:, i, :, \dots, :) *_{\mathcal{L}} \mathbf{V}(:, i, :, \dots, :)^T$  for some  $k < \min(n_1, n_2)$ . Let  $\Theta = \{\mathbf{X} *_{\mathcal{L}} \mathbf{Y} | \mathbf{X} \in \mathbb{R}^{n_1 \times k \times n_3 \times \cdots \times n_d}, \mathbf{Y} \in \mathbb{R}^{k \times n_2 \times n_3 \times \cdots \times n_d}\}$ . (a) Then,  $\mathbf{A}_k = \arg \min_{\tilde{\mathbf{A}} \in \Theta} \|\mathbf{A} - \tilde{\mathbf{A}}\|_F$ , which implies that  $\|\mathbf{A} - \mathbf{A}_k\|_F$  is the theoretical minimal error, given by

$$\|\mathbf{A} - \mathbf{A}_k\|_F^2 = \frac{1}{\rho} \sum_{v=1}^N \sum_{j>k} (\hat{\sigma}_j^{(v)})^2. \quad (4)$$

**Algorithm 1** The Basic randQB Approximation.

**Input:**  $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , invertible linear transform:  $\mathfrak{L}$ , target T-SVD rank:  $k$ , oversampling parameter:  $s$ .

**Output:**  $\mathcal{Q}, \mathcal{B}$ .

- 1 Set  $\hat{l} = k + s$ , and generate a Gaussian random tensor  $\mathcal{G} \in \mathbb{R}^{n_2 \times \hat{l} \times n_3 \times \dots \times n_d}$ ;
- 2 Construct a random projection of  $\mathcal{A}$  as  $\mathcal{Y} = \mathcal{A} *_{\mathfrak{L}} \mathcal{G}$ ;
- 3 Form the partially orthogonal tensor  $\mathcal{Q}$  by computing the T-QR factorization of  $\mathcal{Y}$ ;
- 4  $\mathcal{B} = \mathcal{Q}^T *_{\mathfrak{L}} \mathcal{A}$ .

(b) Then,  $\mathcal{A}_k = \arg \min_{\tilde{\mathcal{A}} \in \mathcal{Q}} \|\mathcal{A} - \tilde{\mathcal{A}}\|$ , which implies that  $\|\mathcal{A} - \mathcal{A}_k\|$  is the theoretical minimal error, given by

$$\|\mathcal{A} - \mathcal{A}_k\| = \max_{v=1, \dots, N} \hat{\sigma}_{k+1}^{(v)}, \quad (5)$$

where  $N = n_3 \cdots n_d$ ,  $\hat{\sigma}_j^{(v)} = \mathfrak{L}(\mathcal{S})^{<v>}_{(j,j)}$ ,  $\rho = \alpha_3 \alpha_4 \cdots \alpha_d > 0$  is a constant determined by  $\mathfrak{L}$ .

#### IV. RANDOMIZED TECHNIQUES BASED HIGH-ORDER TENSOR APPROXIMATION

The optimal  $k$ -term approximation presented in Theorem 1 is time-consuming for large-scale tensors. To tackle this issue, an efficient QB approximation for high-order tensor is developed in virtue of randomized projection techniques. On this basis, we put forward an effective randomized algorithm for calculating the high-order T-SVD (abbreviated as **R-TSVD**). To be slightly more specific, the calculation of R-TSVD can be subdivided into the following two steps:

- **Step I (Randomized Step):** Compute an approximate basis for the range of the target tensor  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_d}$  via randomized projection techniques. That is to say, we require an orthogonal subspace basis tensor  $\mathcal{Q} \in \mathbb{R}^{n_1 \times l \times n_3 \times \dots \times n_d}$  which satisfies

$$\mathcal{A} \approx \mathcal{Q} *_{\mathfrak{L}} \mathcal{B} = \mathcal{Q} *_{\mathfrak{L}} \mathcal{Q}^T *_{\mathfrak{L}} \mathcal{A}. \quad (6)$$

The approximation presented by (6) can also be regarded as a kind of low-rank factorization/approximation of  $\mathcal{A}$ , called QB factorization or QB approximation in our work. A basic randomized technique for computing the QB factorization is shown in Algorithm 1, which is denoted as the basic randomized QB (randQB) approximation.

- **Step II (Deterministic Step):** Perform the deterministic T-QR factorization on the reduced tensor  $\mathcal{B}^T$ , i.e.,  $\mathcal{B}^T = \mathcal{Q}_1 *_{\mathfrak{L}} \mathcal{R}$ . Then, execute the deterministic T-SVD on the smaller tensor  $\mathcal{R}$ , i.e.,  $\mathcal{R} = \hat{\mathcal{U}} *_{\mathfrak{L}} \mathcal{S} *_{\mathfrak{L}} \hat{\mathcal{V}}^T$ . Thus,

$$\begin{aligned} \mathcal{A} &\approx \mathcal{Q} *_{\mathfrak{L}} \hat{\mathcal{V}} *_{\mathfrak{L}} \mathcal{S} *_{\mathfrak{L}} \hat{\mathcal{U}}^T *_{\mathfrak{L}} \mathcal{Q}_1^T \\ &= \mathcal{U} *_{\mathfrak{L}} \mathcal{S} *_{\mathfrak{L}} \mathcal{V}^T \quad (\text{Let } \mathcal{U} = \mathcal{Q} *_{\mathfrak{L}} \hat{\mathcal{V}}, \mathcal{V} = \mathcal{Q}_1 *_{\mathfrak{L}} \hat{\mathcal{U}}). \end{aligned} \quad (7)$$

*Remark 2 (Power Iteration Strategy):* To further improve the accuracy of randQB approximation of  $\mathcal{A}$ , we can additionally apply the power iteration scheme, which multiplies alternately with  $\mathcal{A}$  and  $\mathcal{A}^T$ , i.e.,  $(\mathcal{A} *_{\mathfrak{L}} \mathcal{A}^T)^t *_{\mathfrak{L}} \mathcal{A}$ , where  $t$  is a nonnegative integer. Besides, to avoid the rounding error of float point arithmetic obtained from performing the power iteration, the reorthogonalization step is required. Thus, the randQB approximation algorithm incorporating power iteration strategy can be obtained by adding the following

**Algorithm 2** Transform Domain Version: **R-TSVD**.

**Input:**  $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , transform:  $\mathfrak{L}$ , target T-SVD rank:  $k$ , oversampling parameter:  $s$ , power iteration:  $t$ .

**Output:**  $\mathcal{U}, \mathcal{S}, \mathcal{V}$  such that  $\hat{\mathcal{A}}_k \approx \mathcal{U} *_{\mathfrak{L}} \mathcal{S} *_{\mathfrak{L}} \mathcal{V}^T$ .

- 1 Set  $\hat{l} = k + s$  and initialize a Gaussian random tensor  $\mathcal{G} \in \mathbb{R}^{n_2 \times \hat{l} \times n_3 \times \dots \times n_d}$ ;
- 2 Compute the results of  $\mathfrak{L}$  on  $\mathcal{A}$  and  $\mathcal{G}$ , i.e.,  $\mathfrak{L}(\mathcal{A}), \mathfrak{L}(\mathcal{G})$ ;
- 3 **for**  $v = 1, 2, \dots, n_3 \cdots n_d$  **do**
- 4    $[\mathfrak{L}(\mathcal{Q})^{<v>}, \sim] = \text{qr}(\mathfrak{L}(\mathcal{A})^{<v>} \cdot \mathfrak{L}(\mathcal{G})^{<v>})$ ;
- 5   **for**  $j = 1, 2, \dots, t$  **do**
- 6      $[\mathfrak{L}(\mathcal{Q}_1)^{<v>}, \sim] = \text{qr}((\mathfrak{L}(\mathcal{A})^{<v>})^T \cdot \mathfrak{L}(\mathcal{Q})^{<v>})$ ;
- 7      $[\mathfrak{L}(\mathcal{Q})^{<v>}, \sim] = \text{qr}(\mathfrak{L}(\mathcal{A})^{<v>} \cdot \mathfrak{L}(\mathcal{Q}_1)^{<v>})$ ;
- 8   **end**
- 9    $[\mathfrak{L}(\mathcal{Q}_1)^{<v>}, \mathfrak{L}(\mathcal{R})^{<v>}] = \text{qr}((\mathfrak{L}(\mathcal{A})^{<v>})^T \cdot \mathfrak{L}(\mathcal{Q})^{<v>})$ ;
- 10    $[\mathfrak{L}(\mathcal{U})^{<v>}, \mathfrak{L}(\mathcal{S})^{<v>}, \mathfrak{L}(\hat{\mathcal{V}})^{<v>}] = \text{svd}(\mathfrak{L}(\mathcal{R})^{<v>})$ ;
- 11    $\mathfrak{L}(\mathcal{V})^{<v>} = \mathfrak{L}(\mathcal{Q}_1)^{<v>} \cdot \mathfrak{L}(\hat{\mathcal{U}})^{<v>};$
- 12    $\mathfrak{L}(\mathcal{U})^{<v>} = \mathfrak{L}(\mathcal{Q})^{<v>} \cdot \mathfrak{L}(\hat{\mathcal{V}})^{<v>};$
- 13 **end**
- 14  $\mathcal{U} \leftarrow \mathfrak{L}^{-1}(\mathfrak{L}(\mathcal{U})), \mathcal{S} \leftarrow \mathfrak{L}^{-1}(\mathfrak{L}(\mathcal{S})), \mathcal{V} \leftarrow \mathfrak{L}^{-1}(\mathfrak{L}(\mathcal{V}))$ .

steps after the third step of Algorithm 1, i.e.,

- ```

for  $j = 1, 2, \dots, t$  do
   $[\mathcal{Q}_1, \sim] = \text{H-TQR}(\mathcal{A}^T *_{\mathfrak{L}} \mathcal{Q}, \mathfrak{L});$ 
   $[\mathcal{Q}, \sim] = \text{H-TQR}(\mathcal{A} *_{\mathfrak{L}} \mathcal{Q}_1, \mathfrak{L});$ 
end

```

Empirical studies show that when the number of power iterations  $t$  is equal to 1 or 2, the randQB algorithm is sufficient to achieve similar performance to the truncated T-SVD.

According to the above analysis, the computational procedure of R-TSVD is shown in Algorithm 2. To obtain high performance of linear algebraic computation, we further investigate the blocked version of basic randQB approximation, and then derive a randomized blocked algorithm for computing high-order T-SVD (abbreviated as **RB-TSVD**, see Algorithm 3). Owing to the space limitations of this paper, the detailed derivation of blocked randQB approximation and its induced RB-TSVD are given in the supplementary material. Below, we provide the expected error of the probabilistic part of Algorithm 2 and Algorithm 3 under the Frobenius norm and spectral norm representation, respectively.

*Theorem 2:* Let  $\mathfrak{L}$  be any invertible linear transform in (2) and it satisfies (3). Suppose that  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_d}$  and  $\mathcal{G} \in \mathbb{R}^{n_2 \times \hat{l} \times n_3 \times \dots \times n_d}$  is the Gaussian random tensor with  $\hat{l} < \min(n_1, n_2)$ . If  $\mathcal{Q}, \mathcal{B}$  are obtained from Algorithm 2, then

$$\begin{aligned} \mathbb{E} \|\mathcal{A} - \mathcal{Q} *_{\mathfrak{L}} \mathcal{B}\| &\leq \max_{v=1, \dots, N} \left[ \left(1 + \sqrt{\frac{k}{s-1}}\right) (\hat{\sigma}_{k+1}^{(v)})^{2t+1} \right. \\ &\quad \left. + \frac{e\sqrt{k+s}}{s} \left( \sum_{j>k} (\hat{\sigma}_j^{(v)})^{4t+2} \right)^{1/2} \right]^{1/(2t+1)}, \\ &\quad \mathbb{E} \|\mathcal{A} - \mathcal{Q} *_{\mathfrak{L}} \mathcal{B}\|_F^2 \\ &\leq \frac{1}{\rho} \sum_{v=1}^N \left( 1 + \frac{k}{s-1} (\varrho_k^{(v)})^{(4t)} \right) \left( \sum_{j>k} (\hat{\sigma}_j^{(v)})^2 \right), \end{aligned}$$

where  $\varrho_k^{(v)} = \hat{\sigma}_{k+1}^{(v)} / \hat{\sigma}_k^{(v)} \ll 1$ , the parameters  $N, \hat{\sigma}_j^{(v)}$  and  $\rho$  are equivalent to the ones appeared in Theorem 1, and  $e$  is the basis of the natural exponent.

**Algorithm 3** Transform Domain Version: **RB-TSVD**.

---

**Input:**  $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , transform:  $\mathcal{L}$ , target T-SVD Rank:  $k$ , block size:  $b$ , power iteration:  $t$ .  
**Output:**  $\mathcal{U}, \mathcal{S}, \mathcal{V}$  such that  $\hat{\mathcal{A}}_k \approx \mathcal{U} *_{\mathcal{L}} \mathcal{S} *_{\mathcal{L}} \mathcal{V}^T$ .

```

1 Let  $\hat{l}$  be a number slightly larger than  $k$ , and generate
   a Gaussian random tensor  $\mathcal{G} \in \mathbb{R}^{n_2 \times \hat{l} \times n_3 \times \dots \times n_d}$ ;
2 Compute the results of  $\mathcal{L}$  on  $\mathcal{A}$  and  $\mathcal{G}$ , i.e.,
    $\mathcal{L}(\mathcal{A}), \mathcal{L}(\mathcal{G})$ ;
3 for  $v = 1, 2, \dots, n_3 \dots n_d$  do
4    $\mathcal{L}(\mathcal{Q})^{<v>} = [ ]$ ;  $\mathcal{L}(\mathcal{B})^{<v>} = [ ]$ ;
5   for  $j = 1, 2, \dots, t$  do
6      $[\mathcal{L}(\mathcal{W})^{<v>}, \sim] = \text{qr}(\mathcal{L}(\mathcal{A})^{<v>} \cdot \mathcal{L}(\mathcal{G})^{<v>})$ ;
7      $[\mathcal{L}(\mathcal{G})^{<v>}, \sim] = \text{qr}((\mathcal{L}(\mathcal{A})^{<v>})^T \cdot \mathcal{L}(\mathcal{W})^{<v>})$ ;
8   end
9    $\mathcal{L}(\mathcal{W})^{<v>} = \mathcal{L}(\mathcal{A})^{<v>} \cdot \mathcal{L}(\mathcal{G})^{<v>};$ 
10   $\mathcal{L}(\mathcal{H})^{<v>} = (\mathcal{L}(\mathcal{A})^{<v>})^T \cdot \mathcal{L}(\mathcal{W})^{<v>};$ 
11  for  $i = 1, 2, \dots, \lfloor \frac{\hat{l}}{b} \rfloor$  do
12     $\mathcal{L}(\mathcal{G}^{(i)})^{<v>} = \mathcal{L}(\mathcal{G})^{<v>}(:, (i-1)b+1 : ib)$ ;
13     $\mathcal{L}(\mathcal{Y}^{(i)})^{<v>} = -\mathcal{L}(\mathcal{Q})^{<v>} \cdot \mathcal{L}(\mathcal{B})^{<v>}.$ 
14     $\mathcal{L}(\mathcal{G}^{(i)})^{<v>} + \mathcal{L}(\mathcal{W})^{<v>}(:, (i-1)b+1 : ib) ;$ 
15     $[\mathcal{L}(\mathcal{Q}^{(i)})^{<v>}, \mathcal{L}(\mathcal{R}^{(i)})^{<v>}] = \text{qr}(\mathcal{L}(\mathcal{Y}^{(i)})^{<v>})$ ;
16     $[\mathcal{L}(\mathcal{Q}^{(i)})^{<v>}, (\mathcal{L}(\hat{\mathcal{R}}^{(i)}))^{<v>}] =$ 
       $\text{qr}(\mathcal{L}(\mathcal{Q}^{(i)})^{<v>} - \mathcal{L}(\mathcal{Q})^{<v>} \cdot (\mathcal{L}(\mathcal{Q})^{<v>})^T \cdot$ 
       $\mathcal{L}(\mathcal{Q}^{(i)})^{<v>});$ 
17     $\mathcal{L}(\mathcal{R}^{(i)})^{<v>} = (\mathcal{L}(\hat{\mathcal{R}}^{(i)}))^{<v>} \cdot \mathcal{L}(\mathcal{R}^{(i)})^{<v>};$ 
18     $\mathcal{L}(\mathcal{B}^{(i)})^{<v>} = (\mathcal{L}(\mathcal{R}^{(i)})^{<v>})^{-T} \cdot$ 
       $[(\mathcal{L}(\mathcal{H})^{<v>}(:, (i-1)b+1 : ib))^T -$ 
       $(\mathcal{L}(\mathcal{Y}^{(i)})^{<v>})^T \cdot \mathcal{L}(\mathcal{Q})^{<v>} \cdot \mathcal{L}(\mathcal{B})^{<v>} -$ 
       $(\mathcal{L}(\mathcal{G}^{(i)})^{<v>})^T \cdot (\mathcal{L}(\mathcal{B})^{<v>})^T \cdot \mathcal{L}(\mathcal{B})^{<v>}];$ 
19     $\mathcal{L}(\mathcal{Q})^{<v>} = [\mathcal{L}(\mathcal{Q})^{<v>}, \mathcal{L}(\mathcal{Q}^{(i)})^{<v>}];$ 
20     $\mathcal{L}(\mathcal{B})^{<v>} = [\mathcal{L}(\mathcal{B})^{<v>}, \mathcal{L}(\mathcal{B}^{(i)})^{<v>}]^T;$ 
21  end
22   $[\mathcal{L}(\mathcal{Q}_1)^{<v>}, \mathcal{L}(\mathcal{R}_1)^{<v>}] = \text{qr}((\mathcal{L}(\mathcal{B})^{<v>})^T);$ 
23   $[\mathcal{L}(\hat{\mathcal{U}})^{<v>}, \mathcal{L}(\mathcal{S})^{<v>}, \mathcal{L}(\hat{\mathcal{V}})^{<v>}] =$ 
     $\text{svd}(\mathcal{L}(\mathcal{R}_1)^{<v>});$ 
24   $\mathcal{L}(\mathcal{V})^{<v>} = \mathcal{L}(\mathcal{Q}_1)^{<v>} \cdot \mathcal{L}(\hat{\mathcal{U}})^{<v>};$ 
25   $\mathcal{L}(\mathcal{U})^{<v>} = \mathcal{L}(\mathcal{Q})^{<v>} \cdot \mathcal{L}(\hat{\mathcal{V}})^{<v>};$ 
26 end
26  $\mathcal{U} \leftarrow \mathcal{L}^{-1}(\mathcal{L}(\mathcal{U})), \mathcal{S} \leftarrow \mathcal{L}^{-1}(\mathcal{L}(\mathcal{S})), \mathcal{V} \leftarrow \mathcal{L}^{-1}(\mathcal{L}(\mathcal{V})).$ 

```

---

From the Theorem 2, we can find that as the number of power iteration  $t$  increases, the effect of the residual terms  $\frac{k}{s-1}(\varrho_k(v))^{(4t)}$  in the average Frobenius error decreases, which indicates that the power iteration scheme nearly achieves the optimal error of the truncated T-SVD (see Theorem 1 for details). Note that for  $t = 0$ , when the singular values of  $\mathcal{L}(\mathcal{A})^{<v>}$  decay gradually,  $\sum_{j>k}(\hat{\sigma}_j^{(v)})^2$  can be large, which means that the multiplicative factor  $\frac{k}{s-1}$  will greatly magnify the error so the bound will not be tight. The corresponding analysis for the average spectral error bound is similar to the previous results.

*Remark 3:* As the blocked randQB algorithm is mathematically equivalent to the basic randQB algorithm, Algorithm 3 inherits the theoretical error bounds of Algorithm 2 (if ignoring the round-off error). Note that the above algorithms and corresponding theoretical error bounds can degenerate to the

results existed in matrix/third-order tensor cases. For example, when the FFT is utilized as the linear transforms  $\mathcal{L}$  and  $d$  is set to 3, R-TSVD algorithm and its error bounds reduce to the ones in [36] and [88]. The theoretical results of R-TSVD algorithm is equivalent to the ones in [66] if we consider the case:  $d = 2$ . There is other situation where the randomized blocked algorithms proposed in [92] and [93] are the equivalent version of RB-TSVD algorithm in the matrix case.

## V. ROBUST HIGH-ORDER TENSOR COMPLETION

## A. Proposed Model

In this subsection, we formally introduce the double non-convex model for RHTC, in which the low-rank component is constrained by the HWTSN (see Definition 1), while the noise/outlier component is regularized by its weighted  $\ell_q$ -norm (see Table I). Specifically, suppose that we are given a low T-SVD rank tensor  $\mathcal{L} \in \mathbb{R}^{n_1 \times \dots \times n_d}$  corrupted by the noise or outliers. The corrupted part can be represented by the tensor  $\mathcal{E} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ . Here, both  $\mathcal{L}$  and  $\mathcal{E}$  are of arbitrary magnitude. We do not know the T-SVD rank of  $\mathcal{L}$ . Moreover, we have no idea about the locations of the nonzero entries of  $\mathcal{E}$ , not even how many there are. Then, the goal of the RHTC problem is to achieve the reconstruction (either exactly or approximately) of low-rank component  $\mathcal{L}$  from an observed subset of corrupted tensor  $\mathcal{M} = \mathcal{L} + \mathcal{E}$ . Mathematically, the proposed RHTC model can be formulated as follows:

$$\min_{\mathcal{L}, \mathcal{E}} \|\mathcal{L}\|_{\mathcal{W}_1, \mathcal{S}_p}^p + \lambda \|\mathcal{E}\|_{\mathcal{W}_2, \ell_q}^q, \quad \mathbf{P}_{\Omega}(\mathcal{L} + \mathcal{E}) = \mathbf{P}_{\Omega}(\mathcal{M}), \quad (8)$$

where  $0 < p, q < 1, \lambda > 0$  is the penalty parameter,  $\|\mathcal{L}\|_{\mathcal{W}_1, \mathcal{S}_p}$  denotes the HWTSN of  $\mathcal{L}$  while  $\|\mathcal{E}\|_{\mathcal{W}_2, \ell_q}$  represents the weighted  $\ell_q$ -norm of  $\mathcal{E}$ , and  $\mathcal{W}_1, \mathcal{W}_2$  are the weight tensors which will be updated automatically in the subsequent ADMM optimization, see V-C for details.

*Remark 4:* Recently,  $\ell_q$ -norm regularization term was employed to enhance the robustness against noise/outliers by selecting appropriate positive  $q$  in tensor recovery problem (e.g., [9], [28], [29], [63]). Numerical results given by these investigations showed that non-convex  $\ell_q$ -norm regularizer outperforms convex  $\ell_1$ -norm regularizer. However, small  $q$  will lead to the insensitivity to tensor elements, for example,  $1000^q \approx 0.1^q$  when  $q$  is small enough, and larger elements will be misclassified as small entries in subsequent optimization process (i.e., subproblem (16)). Inspired by the idea that sparsity enhancement by using the weighted  $\ell_1$ -norm minimization [94], [95], [96] in signal recovery, we propose utilizing the weighted  $\ell_q$ -norm as the non-convex noise/outliers regularization. Compared with the  $\ell_q$ -norm that treats each element in noise/outliers component equally, the so-called weighted  $\ell_q$ -norm adaptively assigns different weights to different entries in noise/outliers component.

*Remark 5:* In the model (8), the HWTSN not only gives better approximation to the original low-rank assumption, but also considers the importance of different singular components. Comparing with the previous regularizer, e.g., HTNN [62] that treats the different rank components equally, the proposed one is tighter and more feasible. Besides, the weighted  $\ell_q$ -norm has a superior potential to be sparsity-promoting in comparison with the  $\ell_1$ -norm and  $\ell_q$ -norm. Therefore, the joint HWTSN and weighted  $\ell_q$ -norm enable the underlying low-rank structure in the observed tensor  $\mathcal{M}$  to be well captured, and the robustness against noise/outliers to

be well enhanced. The proposed two nonconvex regularizers mainly involve several key ingredients: 1) flexible order  $d$  and linear transforms  $\mathcal{L}$ ; 2) adjustable parameters  $p$  and  $q$ ; 3) automatically updated weight tensors  $\mathcal{W}_1$  and  $\mathcal{W}_2$ . Their various combinations can degenerate to many existing low-rank tensor recovery models (e.g., [21], [22], [23], [24], [25], [32], [33], [34], [35], [36], [62], [63], [64]).

### B. HWTSN Minimization Problem

In this subsection, we mainly present the solution method of HWTSN minimization problem, that is, the method of solving

$$\arg \min_{\mathcal{X}} \tau \|\mathcal{X}\|_{\mathcal{W}, \mathcal{S}_p}^p + \frac{1}{2} \|\mathcal{X} - \mathcal{Z}\|_F^2. \quad (9)$$

Before providing the solution to problem (9), we first introduce the key lemma and definition.

*Lemma 1:* [97] For the given  $p$  ( $0 < p < 1$ ) and  $w > 0$ , the optimal solution of the following optimization problem

$$\min_x w|x|^p + \frac{1}{2}(x-s)^2 \quad (10)$$

is given by the generalized soft-thresholding (GST) operator:

$$\hat{x} = \text{GST}(s, w, p) = \begin{cases} 0, & \text{if } |s| \leq \delta, \\ \text{sign}(s)\hat{\alpha}^* & \text{if } |s| > \delta, \end{cases}$$

where  $\delta = [2w(1-p)]^{\frac{1}{2-p}} + wp[2w(1-p)]^{\frac{p-1}{2-p}}$  is a threshold value,  $\text{sign}(s)$  denotes the signum function, and  $\hat{\alpha}^*$  can be obtained by solving  $\alpha + wp\alpha^{p-1} - |s| = 0$  ( $\alpha > 0$ ).

*Definition 2 (GTSVT operator):* Let  $\mathcal{A} = \mathcal{U} *_{\mathcal{L}} \mathcal{S} *_{\mathcal{L}} \mathcal{V}^T$  be the T-SVD of  $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ . For any  $\tau > 0$ ,  $0 < p < 1$ , then the generalized Tensor Singular Value Thresholding (GTSVT) operator of  $\mathcal{A}$  is defined as follows

$$\mathcal{D}_{\mathcal{W}, p, \tau}(\mathcal{A}) = \mathcal{U} *_{\mathcal{L}} \mathcal{S}_{\mathcal{W}, p, \tau} *_{\mathcal{L}} \mathcal{V}^T, \quad (11)$$

where  $\mathcal{S}_{\mathcal{W}, p, \tau} = \mathcal{L}^{-1}(\text{GST}(\mathcal{L}(\mathcal{S}), \tau \mathcal{W}, p))$ ,  $\mathcal{W} \in \mathbb{R}^{n_1 \times \dots \times n_d}$  is the weight parameter composed of an order- $d$   $f$ -diagonal tensor, and GST denotes the element-wise shrinkage operator.

*Remark 6:* Since the larger singular values usually carry more important information than the smaller ones, the GTSVT operator requires to satisfy: the larger singular values in the transform domain should be shrunk less, while the smaller ones should be shrunk more. In other words, the weights are selected inversely to the singular values in the transform domain. Thus, the original components corresponding to the larger singular values will be less affected. Compared with the T-SVT operator proposed in [62] (it does not consider the importance of different singular components: shrinks all singular values with the same threshold), the GTSVT operator is more flexible and provides more degree of freedom for the approximation to the original problem.

*Theorem 3:* Let  $\mathcal{L}$  be any invertible linear transform in (2) and it satisfies (3),  $m = \min(n_1, n_2)$ . For any  $\tau > 0$  and  $\mathcal{Z} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , if the weight parameter satisfies

$$0 \leq \mathcal{W}^{<j>}(1, 1) \leq \dots \leq \mathcal{W}^{<j>}(m, m), \quad \forall j \in [n_3 \dots n_d],$$

then the GTSVT operator (11) obeys

$$\mathcal{D}_{\mathcal{W}, p, \tau}(\mathcal{Z}) = \arg \min_{\mathcal{X}} \tau \|\mathcal{X}\|_{\mathcal{W}, \mathcal{S}_p}^p + \frac{1}{2} \|\mathcal{X} - \mathcal{Z}\|_F^2. \quad (12)$$

*Proposition 1:* Let  $\mathcal{A} = \mathcal{Q} *_{\mathcal{L}} \mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ , where  $\mathcal{Q} \in \mathbb{R}^{n_1 \times k \times \dots \times n_d}$  is partially orthogonal and  $\mathcal{B} \in$

---

### Algorithm 4 GST Algorithm [97].

---

```

Input:  $s, w, p, J = 3$  or  $4$ .
Output:  $\text{GST}(s, w, p)$ .
1  $\delta_p^{\text{GST}}(w) = [2w(1-p)]^{\frac{1}{2-p}} + wp[2w(1-p)]^{\frac{p-1}{2-p}};$ 
2 if  $|s| \leq \delta_p^{\text{GST}}(w)$  then
3    $\text{GST}(s, w, p) = 0$  ;
4 else
5    $j = 0, x^{(j)} = |s|$ ;
6   for  $j = 0, 1, \dots, J$  do
7      $x^{(j+1)} = |s| - wp(x^{(j)})^{p-1};$ 
8      $j = j + 1$ ;
9   end
10   $\text{GST}(s, w, p) = \text{sgn}(s)x^{(j)}$ ;
11 end

```

---

### Algorithm 5 Fast Randomized GTSVT, $\mathcal{F}_{\mathcal{W}, p, \tau}$ ( $\mathcal{A}, \mathcal{L}$ ).

---

```

Input:  $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , transform:  $\mathcal{L}$ , target T-SVD
Rank:  $k$ , weight tensor:  $\mathcal{W} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , block
size:  $b$ ,  $0 < p < 1$ ,  $\tau > 0$ , power iteration:  $t$ .
1 Let  $\hat{l}$  be a number slightly larger than  $k$  and generate a
Gaussian random tensor  $\mathcal{G} \in \mathbb{R}^{n_2 \times \hat{l} \times n_3 \times \dots \times n_d}$ ;
2 Compute the results of  $\mathcal{L}$  on  $\mathcal{A}$  and  $\mathcal{G}$ , i.e.,
 $\mathcal{L}(\mathcal{A}), \mathcal{L}(\mathcal{G})$ ;
3 for  $v = 1, 2, \dots, n_3 n_4 \dots n_d$  do
4   if utilize the unblocked randomized technique then
5     Execute Lines 4-12 of Algorithm 2 to obtain
 $\mathcal{L}(\mathcal{U})^{<v>}, \mathcal{L}(\mathcal{S})^{<v>}$ , and  $\mathcal{L}(\mathcal{V})^{<v>};$ 
6   end
7   else if utilize the blocked randomized technique
then
8     Execute Lines 4-24 of Algorithm 3 to obtain
 $\mathcal{L}(\mathcal{U})^{<v>}, \mathcal{L}(\mathcal{S})^{<v>}$ , and  $\mathcal{L}(\mathcal{V})^{<v>};$ 
9   end
10  else if not utilize the randomized technique then
11     $[\mathcal{L}(\mathcal{U})^{<v>}, \mathcal{L}(\mathcal{S})^{<v>}, \mathcal{L}(\mathcal{V})^{<v>}] = \text{svd}(\mathcal{A}_{\mathcal{L}}^{<v>})$ 
12  end;
13   $S = \text{GST}\{\text{diag}(\mathcal{L}(\mathcal{S})^{<v>}), \tau \cdot \text{diag}(\mathcal{W}^{<v>}), p\};$ 
14   $\mathcal{L}(\mathcal{C})^{<v>} = \mathcal{L}(\mathcal{U})^{<v>} \cdot \text{diag}(\hat{S}) \cdot (\mathcal{L}(\mathcal{V})^{<v>})^T;$ 
15 end
Output:  $\mathcal{F}_{\mathcal{W}, p, \tau}(\mathcal{A}, \mathcal{L}) \leftarrow \mathcal{L}^{-1}(\mathcal{L}(\mathcal{C}))$ .

```

---

$\mathbb{R}^{k \times n_2 \times \dots \times n_d}$ . Then, we have

$$\mathcal{D}_{\mathcal{W}, p, \tau}(\mathcal{A}) = \mathcal{Q} *_{\mathcal{L}} \mathcal{D}_{\mathcal{W}, p, \tau}(\mathcal{B}).$$

*Proposition 2:* Let  $\mathcal{A} = \mathcal{Q}_1 *_{\mathcal{L}} \mathcal{B} *_{\mathcal{L}} \mathcal{Q}_2^T \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ , where  $\mathcal{Q}_1 \in \mathbb{R}^{n_1 \times k \times \dots \times n_d}$  and  $\mathcal{Q}_2 \in \mathbb{R}^{n_2 \times k \times \dots \times n_d}$  are partially orthogonal,  $\mathcal{B} \in \mathbb{R}^{k \times k \times \dots \times n_d}$ . Then, we have

$$\mathcal{D}_{\mathcal{W}, p, \tau}(\mathcal{A}) = \mathcal{Q}_1 *_{\mathcal{L}} \mathcal{D}_{\mathcal{W}, p, \tau}(\mathcal{B}) *_{\mathcal{L}} \mathcal{Q}_2^T.$$

From the Definition 2 and Theorem 3, we can find that the major bottleneck of solving the HWTSN minimization problem (9) is to compute the GTSVT operator involving time-consuming T-SVD multiple times. Based on the randomized R-TSVD and RB-TSVD algorithms, fast and efficient method is suggested to boost the computational efficiency of GTSVT operator. The proposed method is called as fast randomized GTSVT, whose calculation is divided into the following two steps: 1) compute the orthogonal subspace basis tensor via randomized techniques; 2) perform the deterministic GTSVT operator on a small-scale projected tensor. The computational procedure of fast randomized GTSVT is

shown in Algorithm 5. Note that the Algorithm 5 is highly parallelizable because the operations across frontal slices in the transformed domain can be readily distributed across different processors. Therefore, additional computational gains can be achieved via the parallel computing framework.

### C. Optimization Algorithm

In this subsection, the ADMM framework [98] is adopted to solve the proposed model (8). The nonconvex model (8) can be equivalently reformulated as follows:

$$\min_{\mathcal{L}, \mathcal{E}} \|\mathcal{L}\|_{\mathcal{W}_1, \mathcal{S}_p}^p + \lambda \|\mathbf{P}_\Omega(\mathcal{E})\|_{\mathcal{W}_2, \ell_q}^q, \text{ s.t. } \mathcal{L} + \mathcal{E} = \mathcal{M}. \quad (13)$$

The augmented Lagrangian function of (13) is

$$\begin{aligned} \mathcal{F}(\mathcal{L}, \mathcal{E}, \mathcal{Y}, \beta) = & \|\mathcal{L}\|_{\mathcal{W}_1, \mathcal{S}_p}^p + \lambda \|\mathbf{P}_\Omega(\mathcal{E})\|_{\mathcal{W}_2, \ell_q}^q \\ & + \langle \mathcal{Y}, \mathcal{L} + \mathcal{E} - \mathcal{M} \rangle + \beta/2 \|\mathcal{L} + \mathcal{E} - \mathcal{M}\|_F^2, \end{aligned} \quad (14)$$

where  $\mathcal{Y}$  is the dual variable and  $\beta$  is the regularization parameter. The ADMM framework alternately updates each optimization variable until convergence. The iteration template of the ADMM at the  $(\psi + 1)$ -th iteration is described as follows:

$$\mathcal{L}^{\psi+1} = \arg \min_{\mathcal{L}} \{\mathcal{F}(\mathcal{L}, \mathcal{E}^\psi, \mathcal{Y}^\psi, \beta^\psi)\}, \quad (15)$$

$$\mathcal{E}^{\psi+1} = \arg \min_{\mathcal{E}} \{\mathcal{F}(\mathcal{L}^{\psi+1}, \mathcal{E}, \mathcal{Y}^\psi, \beta^\psi)\}, \quad (16)$$

$$\mathcal{Y}^{\psi+1} = \mathcal{Y}^\psi + \beta^\psi (\mathcal{L}^{\psi+1} + \mathcal{E}^{\psi+1} - \mathcal{M}), \quad (17)$$

$$\beta^{\psi+1} = \min(\beta^{\max}, \vartheta \beta^\psi), \quad (18)$$

where  $\vartheta > 1$  is a control constant. Now we solve the subproblem (15) and (16) explicitly in the ADMM, respectively.

**Update  $\mathcal{L}^{\psi+1}$  (low-rank component)** The optimization subproblem (15) concerning  $\mathcal{L}^{\psi+1}$  can be written as

$$\min_{\mathcal{L}} \|\mathcal{L}\|_{\mathcal{W}_1, \mathcal{S}_p}^p + \beta^\psi/2 \|\mathcal{L} - \mathcal{M} + \mathcal{E}^\psi + \mathcal{Y}^\psi/\beta^\psi\|_F^2. \quad (19)$$

Let  $\mathcal{G}^\psi = \mathcal{M} - \mathcal{E}^\psi - \mathcal{Y}^\psi/\beta^\psi$ . Using the GTSVT algorithm that incorporates the randomized schemes, the subproblem (19) can be efficiently solved, i.e.,  $\mathcal{L}^{\psi+1} = \mathcal{F}_{\mathcal{W}_1, p, \frac{1}{\beta^\psi}}(\mathcal{G}^\psi, \mathcal{L})$ .

*Remark 7 (Update  $\mathcal{W}_1$  via reweighting strategy):* The weight tensor  $\mathcal{W}_1$  can be adaptively tuned at each iteration, and its formula in the  $\psi$ -th iteration is given by

$$(\mathcal{W}_1^\psi)^{<j>} (i, i) = \frac{c_1}{(\mathcal{K}^\psi)^{<j>} (i, i) + \epsilon_1} \cdot \frac{1}{\beta^\psi},$$

where  $j \in \{1, \dots, n_3 \dots n_d\}$ ,  $i \in \{1, \dots, \min(n_1, n_2)\}$ ,  $c_1 > 0$  is a constant,  $\epsilon_1$  is a small non-negative constant to avoid division by zero, and the entries on the diagonal of  $(\mathcal{K}^\psi)^{<j>}$  represent the singular values of  $\mathcal{L}(\mathcal{G}^\psi)^{<j>}$ . In such a reweighted technique, the sparsity performance is enhanced after each iteration and the updated  $\mathcal{W}_1^\psi$  satisfy:

$$(\mathcal{W}_1^\psi)^{<j>} (m, m) \geq (\mathcal{W}_1^\psi)^{<j>} (n, n) \geq 0, \forall m \geq n.$$

**Update  $\mathcal{E}^{\psi+1}$  (noise/outliers component)** The optimization subproblem (16) with respect to  $\mathcal{E}^{\psi+1}$  can be written as

$$\begin{aligned} \min_{\mathcal{E}} \lambda \|\mathbf{P}_\Omega(\mathcal{E})\|_{\mathcal{W}_2, \ell_q}^q + \beta^\psi/2 \|\mathcal{E} - \mathcal{M} + \mathcal{L}^{\psi+1} \\ + \mathcal{Y}^\psi/\beta^\psi\|_F^2. \end{aligned}$$

**Algorithm 6** Solve the Proposed Model (8) by ADMM.

**Input:**  $\mathbf{P}_\Omega(\mathcal{M}) \in \mathbb{R}^{n_1 \times \dots \times n_d}$ ,  $\mathcal{L}$ ,  $\lambda$ , target T-SVD Rank:  $k$ , block size:  $b$ , power iteration:  $t$ ,  $0 < p, q < 1$ ,  $\tau > 0$ ,  $c_1, c_2, \epsilon_1, \epsilon_2$ .

**1 Initialize:**  $\mathcal{L}^0 = \mathcal{E}^0 = \mathcal{Y}^0 = \mathbf{0}$ ,  $\vartheta, \beta^0, \beta^{\max}, \varpi, \psi = 0$ ;

**2 while** not converged **do**

**3**    Update  $\mathcal{L}^{\psi+1}$  by computing Algorithm 5;

**4**    Update  $\mathbf{P}_\Omega(\mathcal{E}^{\psi+1})$  by computing (20);

**5**    Update  $\mathbf{P}_{\Omega_\perp}(\mathcal{E}^{\psi+1})$  by computing (21);

**6**    Update  $\mathcal{Y}^{\psi+1}$  by computing (17);

**7**    Update  $\beta^{\psi+1}$  by computing (18);

**8**    Check the convergence conditions

$$\|\mathcal{L}^{\psi+1} - \mathcal{L}^\psi\|_\infty \leq \varpi, \|\mathcal{E}^{\psi+1} - \mathcal{E}^\psi\|_\infty \leq \varpi,$$

$$\|\mathcal{L}^{\psi+1} + \mathcal{E}^{\psi+1} - \mathcal{M}\|_\infty \leq \varpi.$$

**9 end**

**Output:**  $\mathcal{L} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ .

Let  $\mathcal{H}^\psi = \mathcal{M} - \mathcal{L}^{\psi+1} - \mathcal{Y}^\psi/\beta^\psi$ . The above problem can be solved by the following two subproblems with respect to  $\mathbf{P}_\Omega(\mathcal{E}^{\psi+1})$  and  $\mathbf{P}_{\Omega_\perp}(\mathcal{E}^{\psi+1})$ , respectively. Note that the weight tensor  $\mathcal{W}_2$  is updated at each iteration, and its form at the  $\psi$ -th iteration is set as follows:

$$(\mathcal{W}_2^\psi)(i_1, \dots, i_d) = \frac{c_2}{|\mathcal{H}^\psi(i_1, \dots, i_d)| + \epsilon_2} \cdot \frac{1}{\beta^\psi},$$

in which  $c_2 > 0$  is a constant,  $\epsilon_2 > 0$  is a small constant to avoid division by zero.

**Regarding  $\mathbf{P}_\Omega(\mathcal{E}^{\psi+1})$ :** the optimization subproblem with respect to  $\mathbf{P}_\Omega(\mathcal{E}^{\psi+1})$  is formulated as following

$$\min_{\mathbf{P}_\Omega(\mathcal{E})} \lambda \|\mathbf{P}_\Omega(\mathcal{E})\|_{\mathcal{W}_2, \ell_q}^q + \beta^\psi/2 \|\mathbf{P}_\Omega(\mathcal{E} - \mathcal{H}^\psi)\|_F^2. \quad (20)$$

The closed-form solution for subproblem (20) can be computed by generalized element-wise shrinkage operator, i.e.,

$$\mathbf{P}_\Omega(\mathcal{E}^{\psi+1}) = \text{GST}(\mathbf{P}_\Omega(\mathcal{H}^\psi), \lambda \cdot (\beta^\psi)^{-1} \cdot \mathbf{P}_\Omega(\mathcal{W}_2^\psi), q).$$

**Regarding  $\mathbf{P}_{\Omega_\perp}(\mathcal{E}^{\psi+1})$ :** the optimization subproblem with respect to  $\mathbf{P}_{\Omega_\perp}(\mathcal{E}^{\psi+1})$  is formulated as following

$$\mathbf{P}_{\Omega_\perp}(\mathcal{E}^{\psi+1}) = \min_{\mathbf{P}_{\Omega_\perp}(\mathcal{E})} \beta^\psi/2 \|\mathbf{P}_{\Omega_\perp}(\mathcal{E} - \mathcal{H}^\psi)\|_F^2. \quad (21)$$

The closed-form solution for subproblem (21) can be obtained through the standard least square regression method.

### D. Convergence Analysis

In this subsection, we provide a theoretical guarantees for the convergence of the proposed Algorithm 6.

**Theorem 4:** The sequences  $\{\mathcal{Y}^\psi\}$ ,  $\{\mathcal{L}^\psi\}$  and  $\{\mathcal{E}^\psi\}$  generated by Algorithm 6 are bounded.

**Theorem 5:** Let  $\mathcal{L}$  be any invertible linear transform in (2) and it satisfies (3),  $m = \min(n_1, n_2)$ . If the diagonal elements of all matrix frontal slices on weighted tensor  $\mathcal{W}_1^\psi$  are sorted in a non-descending order, i.e.,

$$(\mathcal{W}_1^\psi)^{<j>} (1, 1) \dots \leq (\mathcal{W}_1^\psi)^{<j>} (m, m), \forall j \in [n_3 \dots n_d],$$

then the sequences  $\{\mathcal{L}^{\psi+1}\}$ ,  $\{\mathcal{E}^{\psi+1}\}$  and  $\{\mathcal{Y}^{\psi+1}\}$  generated by Algorithm 6 satisfy:

- 1)  $\lim_{\psi \rightarrow \infty} \|\mathcal{L}^{\psi+1} - \mathcal{L}^\psi\|_F = 0$ ; 2)  $\lim_{\psi \rightarrow \infty} \|\mathcal{E}^{\psi+1} - \mathcal{E}^\psi\|_F = 0$ ;
- 3)  $\lim_{\psi \rightarrow \infty} \|\mathcal{M} - \mathcal{L}^{\psi+1} - \mathcal{E}^{\psi+1}\|_F = 0$ .

### E. Complexity Analysis

Given an input tensor  $\mathbf{P}_\Omega(\mathcal{M}) \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , we analyze the per-iteration complexity of Algorithm 6 with/without randomized techniques. The per-iteration of Algorithm 6 needs to update  $\mathcal{L}$ ,  $\mathbf{P}_\Omega(\mathcal{E})$ ,  $\mathbf{P}_{\Omega^\perp}(\mathcal{E})$ ,  $\mathcal{Y}$ , respectively. Upadating  $\mathbf{P}_\Omega(\mathcal{E})$  requires to perform GST operation with a complexity of  $\mathcal{O}(|\Omega|)$ , where  $|\Omega|$  denotes the cardinality of  $\Omega$ .  $\mathbf{P}_{\Omega^\perp}(\mathcal{E})$  and  $\mathcal{Y}$  can be updated by a low consumed algebraic computation. The update of  $\mathcal{L}$  mainly involves matrix-matrix product, economic QR/SVD decomposition, linear transforms  $\mathcal{L}(\cdot)$  and its inverse operator  $\mathcal{L}^{-1}(\cdot)$ . Specifically, for any invertible linear transforms  $\mathcal{L}$ , the per-iteration complexity of  $\mathcal{L}$  is

- 1)  $\mathcal{O}(\prod_{i=1}^d n_i \cdot \sum_{j=3}^d n_j + \hat{l} \cdot \prod_{k=1}^d n_k)$ , with randomized technique;
- 2)  $\mathcal{O}(\prod_{i=1}^d n_i \cdot \sum_{j=3}^d n_j + \min\{n_1, n_2\} \cdot \prod_{k=1}^d n_k)$ , without randomized technique.

For some special invertible linear transforms, e.g., Fast Fourier Transform (FFT), the per-iteration complexity of  $\mathcal{L}$  is

- 1)  $\mathcal{O}(\prod_{i=1}^d n_i \cdot \sum_{j=3}^d \log(n_j) + \hat{l} \cdot \prod_{k=1}^d n_k)$ , with randomized technique;
- 2)  $\mathcal{O}(\prod_{i=1}^d n_i \cdot \sum_{j=3}^d \log(n_j) + \min\{n_1, n_2\} \cdot \prod_{k=1}^d n_k)$ , without randomized technique.

It is obvious that the versions using randomized technique can be advantageous when  $\hat{l} \ll \min\{n_1, n_2\}$ .

## VI. EXPERIMENTAL RESULTS

In this section, we perform extensive experiments on both synthetic and real-world tensor data to substantiate the superiority and effectiveness of the proposed approach. Due to the page limitations of this paper, partial synthetic and real experiments are provided in the supplementary material. All the experiments are run on the following two platforms: **1**) Windows 10 and Matlab (R2016a) with an Intel(R) Xeon(R) Gold-5122 3.60GHz CPU and 192GB memory; **2**) Windows 10 and Matlab (R2022b) with an Intel(R) Xeon(R) Gold-6230 2.10GHz CPU and 128GB memory.

### A. Real-World Applications

**Datasets:** In this section, we choose three types of real-world high-order tensor to verify the superiority and effectiveness of the proposed method over compared algorithms. Specifically, **Type I** mainly contains four fifth-order LFIIs named Bench, Bee-1, Framed and Mini, respectively. These LFIs with the size of  $434 \times 625 \times 3 \times 15 \times 15$  can be downloaded from the lytro illum light field dataset website.<sup>2</sup> **Type II** includes four four-order CVs called Rush-hour, Johnny, Stockholm, Intotree, respectively. We download these large-scale CVs from the derf website.<sup>3</sup> Only the first 100 frames of each video sequence are selected as the tested CVs owing to the computational limitation, in which each frame has the

<sup>2</sup><https://www.irisa.fr/temics/demos/IllumDatasetLF/index.html>

<sup>3</sup><https://media.xiph.org/video/derf/>

size  $720 \times 1280 \times 3$ . For each CV with 100 frames, it can be formulated as an  $720 \times 1280 \times 3 \times 100$  fourth-order tensor. **Type III** mainly involves three fourth-order MRSIs, which are named SPOT-5<sup>4</sup> ( $2000 \times 2000 \times 4 \times 13$ ), Landsat-7 ( $4500 \times 4500 \times 6 \times 11$ ), and T22LGN<sup>5</sup> ( $5001 \times 5001 \times 4 \times 7$ ), respectively. To speed up the calculation process, the spatial size of these MRSIs is downsampled (resized) to  $2000 \times 2000$ .

**Experiment Settings:** In this section, we mainly apply the proposed method (**HWTSN+w $\ell_q$** ) and its two accelerated versions to several real-world applications, and also compare them with other RLRTC approaches: **SNN+ $\ell_1$**  [11], **TRNN+ $\ell_1$**  [15], **TTNN+ $\ell_1$**  [24], **TSP- $k+\ell_1$**  [26], **LNOP** [28], **NRTRM** [29], **BCNRTC** [31], and **HWTNN+ $\ell_1$**  [64]. Two accelerated versions (i.e., they incorporate the unblocked and blocked randomized strategies, respectively) of “**HWTSN+w $\ell_q$** ” are called **HWTSN+w $\ell_q$ (UR)** and **HWTSN+w $\ell_q$ (BR)**, respectively. In our experiments, we normalize the gray-scale value of the tested tensors to the interval  $[0, 1]$ . For the RLRTC methods based on third-order T-SVD, we reshape the last two or three modes of tested tensors into one mode. The observed tensor is constructed as follows: the random-valued impulse noise with ratio  $\tau$  is uniformly and randomly added to each frontal slice of the ground-truth tensor, and then we sample  $(sr \cdot \prod_{i=1}^d n_i)$  pixels from the noisy tensor to form the observed tensor  $\mathbf{P}_\Omega(\mathcal{M})$  at random. Unless otherwise stated, all parameters involved in the competing methods were optimally assigned or selected as suggested in the reference papers. The Peak Signal-to-Noise Ratio (PSNR), the structural similarity (SSIM), and the CPU time are employed to evaluate the recovery performance. Generally, better recovery performances are reflected by higher PSNR and SSIM values.

**Parameters Setting:** The transform in **TTNN+ $\ell_1$** , **NRTRM**, **HWTNN+ $\ell_1$** , and **HWTSN+w $\ell_q$**  is set as the FFT for consistency. The parameter  $\lambda$  of **SNN+ $\ell_1$**  is set as  $\lambda \in \{[10, 10, 1, 1] * \alpha, [10, 10, 1, 1] * \alpha\}$ ,  $\alpha \in \{1, 3, 5, 8, 10\}$ . For **TRNN+ $\ell_1$** , we set  $\lambda \in \{0.018, 0.015, 0.012, 0.01, 0.009, 0.007\}$ . For **TTNN+ $\ell_1$** , we set  $\lambda = \varsigma / (\max(n_1, n_2) \cdot \prod_{i=3}^d n_i)^{1/2}$ ,  $\varsigma \in \{1, 1.2, 1.5, 1.8, 2\}$ . For **TSP- $k+\ell_1$** , we set  $k \in \{3, 4, 5\}$ ,  $\lambda \in \{100, 200, 300, 400\}$ . For **LNOP**, the parameter  $p$  of  $\ell_p$ -ball projection is set to be 0.7,  $\epsilon = 500$ ,  $\lambda = 10^7$ . For **NRTRM**, the minimax concave penalty (MCP) function is utilized in both regularizers  $G_1, G_2$ , the parameter  $\eta$  of MCP is chosen as  $\max(n_1, n_2)/\alpha^k$  for  $G_1$  and  $1/\beta^k$  for  $G_2$ , respectively;  $c \in \{0.7, 0.9, 1.4\}$ , and  $\lambda = \kappa / (\max(n_1, n_2) \cdot \prod_{i=3}^d n_i)^{1/2}$ ,  $\kappa \in \{1.2, 1.5, 1.8, 2, 2.2, 2.5\}$ . For **HWTNN+ $\ell_1$** , we set  $c = \max(n_1, n_2)$ ,  $\epsilon = 10^{-16}$  and  $\lambda = \theta / (\max(n_1, n_2) \cdot \prod_{i=3}^d n_i)^{1/2}$ ,  $\theta \in \{25, 30, 35, 40, 45, 50, 55, 60, 65, 70\}$ . For **our algorithms**, we set  $t = 1$ ,  $\vartheta = 1.15$ ,  $\beta^0 = 10^{-3}$ ,  $\beta^{\max} = 10^8$ ,  $\varpi = 10^{-4}$ ,  $\epsilon_1 = \epsilon_2 = 10^{-16}$ ,  $c_1 = \omega \cdot \max(n_1, n_2)$ ,  $\omega \in \{0.5, 1, 2, 5, 10, 15, 20, 25, 30\}$ ,  $c_2 = 1$ ,  $\lambda = \xi / (\max(n_1, n_2) \cdot \prod_{i=3}^d n_i)^{1/2}$ ,  $\xi \in \{1, 3, 5, 6, 8, 10, 12, 15\}$ ,  $k = 100$ ,  $b = 20$  for MRSIs ( $k = 50$ ,  $b = 10$  for CVs and LFIs). The adjustable parameters  $p$  and  $q$  are set to be inversely proportional to the constant  $c_1$ , respectively. In other words, as  $p$  and  $q$  go up,  $\omega$  goes down.  $p$  and  $q$  are selected as 0.9 in VI-A.1–VI-A.3.

<sup>4</sup><https://take5.theia.cnes.fr/atdistrib/take5/client/#/home>

<sup>5</sup><https://theia.cnes.fr/atdistrib/rocket/#/home>

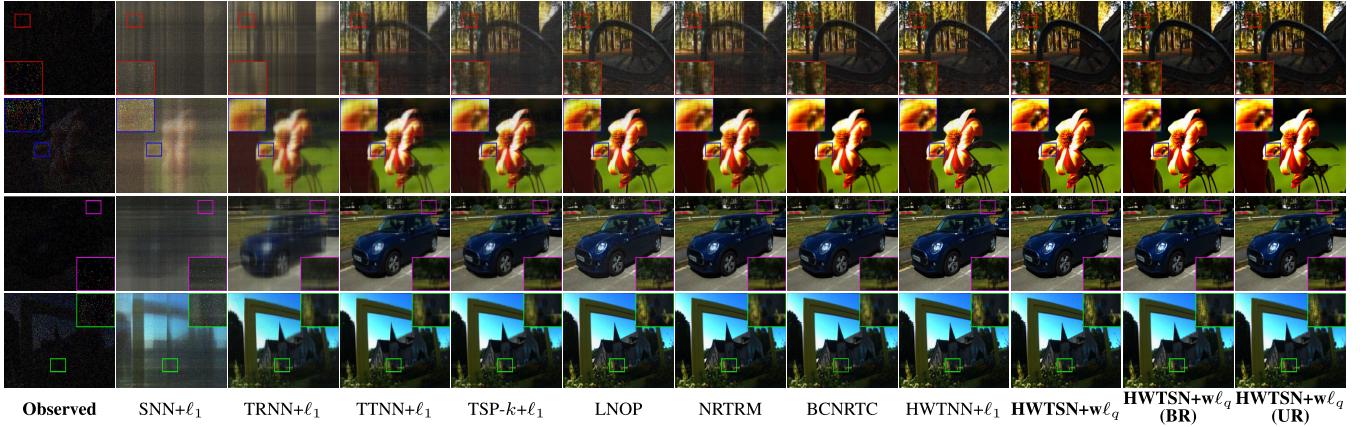


Fig. 1. Visual comparison of various methods for LFIs recovery. From top to bottom, the parameter pair  $(sr, \tau)$  are  $(0.05, 0.5)$ ,  $(0.1, 0.5)$ ,  $(0.05, 0.3)$  and  $(0.1, 0.3)$ , respectively. Top row: the  $(6, 6)$ -th frame of Bench. The second row: the  $(9, 9)$ -th frame of Bee-1. The third row: the  $(12, 12)$ -th frame of Mini. Bottom row: the  $(15, 15)$ -th frame of Framed.

TABLE II

THE PSNR, SSIM VALUES AND CPU TIME OBTAINED BY VARIOUS RLRTC METHODS FOR DIFFERENT FIFTH-ORDER LFIS. THE BEST AND THE SECOND-BEST RESULTS ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY

| LFI-Name             | Bench                        |                              |                              |                              | Bee-1                        |                              |                              |                              | Framed                       |                              |                              |                              | Mini                         |                              |                              |                              | Average Time (s) |
|----------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------|
|                      | $sr$                         |                              | 5%                           |                              | 10%                          |                              | 5%                           |                              | 10%                          |                              | 5%                           |                              | 10%                          |                              | 5%                           |                              |                  |
| $\tau$               | 30%                          | 50%                          | 30%                          | 50%                          | 30%                          | 50%                          | 30%                          | 50%                          | 30%                          | 50%                          | 30%                          | 50%                          | 30%                          | 50%                          | 30%                          | 50%                          |                  |
| SNN+ $\ell_1$ [11]   | 14.29<br>0.176               | 11.94<br>0.091               | 14.81<br>0.267               | 13.69<br>0.168               | 12.72<br>0.137               | 11.07<br>0.081               | 13.39<br>0.179               | 11.74<br>0.128               | 12.76<br>0.203               | 11.47<br>0.112               | 13.99<br>0.269               | 11.67<br>0.179               | 14.29<br>0.195               | 12.92<br>0.132               | 15.99<br>0.245               | 15.37<br>0.197               | 10871            |
|                      | 21.65<br>0.704               | 18.16<br>0.434               | 21.27<br>0.746               | 17.29<br>0.608               | 18.04<br>0.776               | 13.71<br>0.353               | 22.74<br>0.831               | 16.66<br>0.464               | 19.07<br>0.797               | 14.02<br>0.511               | 23.07<br>0.355               | 17.32<br>0.643               | 20.83<br>0.469               | 16.16<br>0.483               | 25.25<br>0.394               | 19.72<br>0.658               |                  |
| TRNN+ $\ell_1$ [15]  | 18.16<br>0.446               | 14.89<br>0.312               | 21.27<br>0.574               | 17.29<br>0.337               | 18.04<br>0.356               | 13.71<br>0.224               | 22.74<br>0.474               | 16.66<br>0.316               | 19.07<br>0.511               | 14.02<br>0.355               | 23.07<br>0.643               | 17.32<br>0.469               | 20.83<br>0.483               | 16.16<br>0.394               | 25.25<br>0.658               | 19.72<br>0.408               | 9111             |
|                      | 22.33<br>0.687               | 17.43<br>0.415               | 24.02<br>0.785               | 21.21<br>0.611               | 22.88<br>0.734               | 17.08<br>0.333               | 25.62<br>0.843               | 21.32<br>0.456               | 23.43<br>0.772               | 17.81<br>0.491               | 22.24<br>0.863               | 20.75<br>0.661               | 26.36<br>0.757               | 18.04<br>0.486               | 27.45<br>0.818               | 22.92<br>0.638               |                  |
| TTNN+ $\ell_1$ [24]  | 21.65<br>0.704               | 16.48<br>0.434               | 23.04<br>0.746               | 20.72<br>0.608               | 22.44<br>0.776               | 16.24<br>0.353               | 24.74<br>0.831               | 20.75<br>0.464               | 23.33<br>0.797               | 16.95<br>0.516               | 25.37<br>0.687               | 22.13<br>0.667               | 25.67<br>0.775               | 17.51<br>0.486               | 27.45<br>0.818               | 22.92<br>0.638               | 4708             |
|                      | 22.33<br>0.687               | 17.43<br>0.415               | 24.02<br>0.785               | 21.21<br>0.611               | 22.88<br>0.734               | 17.08<br>0.333               | 25.62<br>0.843               | 21.32<br>0.456               | 23.43<br>0.772               | 17.81<br>0.491               | 22.24<br>0.863               | 20.75<br>0.661               | 26.36<br>0.757               | 18.04<br>0.486               | 27.45<br>0.818               | 22.92<br>0.638               |                  |
| TSP- $k+\ell_1$ [26] | 21.87<br>0.628               | 18.35<br>0.436               | 27.02<br>0.813               | 21.84<br>0.649               | 22.46<br>0.658               | 18.08<br>0.348               | 28.37<br>0.879               | 21.87<br>0.512               | 22.88<br>0.674               | 19.05<br>0.493               | 22.79<br>0.839               | 20.97<br>0.637               | 24.58<br>0.658               | 20.97<br>0.529               | 30.89<br>0.853               | 25.73<br>0.725               | 6798             |
|                      | 23.94<br>0.789               | 18.66<br>0.445               | 25.79<br>0.817               | 21.91<br>0.646               | 25.72<br>0.826               | 17.97<br>0.354               | 28.19<br>0.846               | 21.32<br>0.485               | 23.43<br>0.871               | 17.81<br>0.512               | 22.24<br>0.909               | 20.75<br>0.716               | 26.36<br>0.842               | 18.04<br>0.506               | 29.02<br>0.891               | 23.93<br>0.712               |                  |
| LNOP [28]            | 21.87<br>0.628               | 18.35<br>0.436               | 27.02<br>0.813               | 21.84<br>0.649               | 22.46<br>0.658               | 18.08<br>0.348               | 28.37<br>0.879               | 21.87<br>0.512               | 22.88<br>0.674               | 19.05<br>0.493               | 22.79<br>0.839               | 20.97<br>0.637               | 24.58<br>0.658               | 20.97<br>0.529               | 30.89<br>0.853               | 25.73<br>0.725               | 6465             |
|                      | 23.67<br>0.764               | 19.41<br>0.458               | 26.41<br>0.838               | 22.63<br>0.703               | 25.46<br>0.834               | 19.01<br>0.362               | 28.89<br>0.891               | 23.73<br>0.599               | 26.05<br>0.864               | 20.53<br>0.541               | 29.11<br>0.927               | 24.84<br>0.789               | 28.30<br>0.829               | 20.31<br>0.518               | 31.92<br>0.913               | 26.59<br>0.773               |                  |
| NRTRM [29]           | 23.94<br>0.789               | 18.66<br>0.445               | 25.79<br>0.817               | 21.91<br>0.646               | 25.72<br>0.826               | 17.97<br>0.354               | 28.19<br>0.846               | 21.32<br>0.485               | 23.43<br>0.871               | 17.81<br>0.512               | 22.24<br>0.909               | 20.75<br>0.716               | 26.36<br>0.842               | 18.04<br>0.506               | 29.02<br>0.891               | 23.93<br>0.712               | 5895             |
|                      | 23.94<br>0.789               | 18.66<br>0.445               | 25.79<br>0.817               | 21.91<br>0.646               | 25.72<br>0.826               | 17.97<br>0.354               | 28.19<br>0.846               | 21.32<br>0.485               | 23.43<br>0.871               | 17.81<br>0.512               | 22.24<br>0.909               | 20.75<br>0.716               | 26.36<br>0.842               | 18.04<br>0.506               | 29.02<br>0.891               | 23.93<br>0.712               |                  |
| HWTNN+ $\ell_1$ [64] | <b>25.63</b><br><b>0.808</b> | 21.94<br>0.543               | <b>27.77</b><br><b>0.854</b> | 23.95<br>0.705               | 28.24<br>0.839               | 22.29<br>0.407               | <b>30.53</b><br><b>0.896</b> | 25.25<br>0.564               | <b>28.45</b><br><b>0.893</b> | 23.03<br>0.583               | <b>31.08</b><br><b>0.931</b> | 26.39<br>0.767               | <b>30.89</b><br><b>0.872</b> | 25.41<br>0.629               | <b>33.29</b><br><b>0.919</b> | 28.44<br>0.776               | 5754             |
|                      | <b>26.85</b><br><b>0.834</b> | <b>23.44</b><br><b>0.671</b> | <b>30.13</b><br><b>0.892</b> | <b>26.29</b><br><b>0.808</b> | <b>29.56</b><br><b>0.889</b> | <b>25.28</b><br><b>0.651</b> | <b>33.03</b><br><b>0.938</b> | 27.69<br>0.723               | <b>29.95</b><br><b>0.916</b> | <b>25.55</b><br><b>0.743</b> | <b>33.78</b><br><b>0.961</b> | <b>29.89</b><br><b>0.915</b> | <b>32.25</b><br><b>0.902</b> | <b>28.99</b><br><b>0.766</b> | <b>35.82</b><br><b>0.952</b> | <b>32.78</b><br><b>0.898</b> |                  |
| HWTSN+w $\ell_q$     | 24.83<br>0.743               | 22.86<br>0.645               | 25.36<br>0.746               | 24.62<br>0.721               | 28.16<br>0.846               | 24.85<br>0.661               | 28.69<br>0.874               | <b>27.79</b><br><b>0.847</b> | 28.15<br>0.863               | 25.25<br>0.735               | 29.01<br>0.884               | 27.98<br>0.868               | 29.39<br>0.808               | 27.29<br>0.723               | 29.55<br>0.817               | 29.19<br>0.811               | 2869             |
|                      | 24.92<br>0.741               | <b>22.96</b><br><b>0.649</b> | 25.46<br>0.749               | <b>24.78</b><br><b>0.728</b> | <b>28.74</b><br><b>0.846</b> | <b>24.94</b><br><b>0.663</b> | 28.76<br>0.874               | <b>27.88</b><br><b>0.848</b> | 28.23<br>0.864               | <b>25.28</b><br><b>0.736</b> | 29.19<br>0.886               | <b>28.29</b><br><b>0.869</b> | 29.46<br>0.809               | <b>27.38</b><br><b>0.725</b> | 29.67<br>0.816               | <b>29.25</b><br><b>0.812</b> |                  |

In each RLRTC method, the top represents the PSNR values while the bottom denotes the SSIM values.

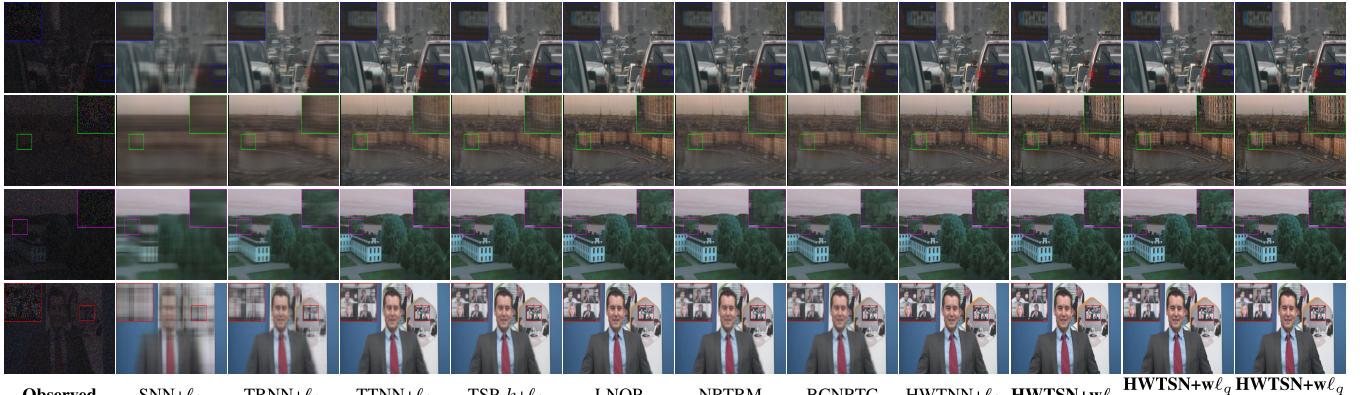


Fig. 2. Visual comparison of various methods for CVs restoration. From top to bottom, the parameter pair  $(sr, \tau)$  are  $(0.1, 0.3)$ ,  $(0.1, 0.5)$ ,  $(0.2, 0.3)$  and  $(0.2, 0.5)$ , respectively. Top row: the 60-th frame of Rush-hour. The second row: the 50-th frame of Stockholm. The third row: the 34-th frame of Intotree. Bottom row: the 10-th frame of Johnny.

**1) Application in LFIs Recovery:** In this experiment, we choose four fifth-order LFIs to showcase the superiority and effectiveness of the proposed algorithms. Figure 1 shows the recovered LFIs and corresponding zoomed regions acquired by different RLRTC methods at two extremely low sampling rates (i.e.,  $sr = 5\%, 10\%$ ) and two noise levels

(i.e.,  $\tau = 30\%, 50\%$ ). From the enlarged areas, we can observe that the LFIs restored by our method preserves more details than those achieved by other competitive algorithms. The PSNR, SSIM values and average CPU time of various RLRTC methods for four LFIs with different noise levels and observation ratios are displayed in Table II. The following

TABLE III

THE PSNR, SSIM VALUES AND CPU TIMEOBTAINED BY VARIOUS RLRTC METHODS FOR DIFFERENT FOURTH-ORDER CVs. THE BEST AND THE SECOND-BEST RESULTS ARE HIGHLIGHTED IN REDAND BLUE, RESPECTIVELY

| CV-Name               | Rush-hour    |              |              |              | Johnny       |              |              |              | Stockholm    |              |              |              | Intotree     |              |              |              | Average Time (s) |
|-----------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|------------------|
|                       | $sr$         |              | 10%          | 20%          | $\tau$       |              | 10%          | 20%          | $sr$         |              | 10%          | 20%          | $\tau$       |              | 10%          | 20%          |                  |
|                       | 30%          | 50%          | 30%          | 50%          | 30%          | 50%          | 30%          | 50%          | 30%          | 50%          | 30%          | 50%          | 30%          | 50%          | 30%          | 50%          |                  |
| SNN+ $\ell_1$ [11]    | 18.37        | 17.26        | 21.78        | 19.57        | 18.03        | 16.61        | 21.80        | 18.61        | 19.96        | 18.96        | 21.42        | 20.53        | 20.81        | 18.88        | 23.08        | 21.85        | 12968            |
|                       | 0.711        | 0.559        | 0.771        | 0.689        | 0.758        | 0.734        | 0.803        | 0.745        | 0.501        | 0.484        | 0.537        | 0.502        | 0.591        | 0.512        | 0.622        | 0.539        |                  |
| TRNN+ $\ell_1$ [15]   | 23.79        | 20.47        | 26.28        | 23.02        | 24.59        | 19.84        | 27.18        | 23.06        | 22.19        | 20.77        | 23.48        | 21.64        | 24.54        | 22.28        | 25.97        | 23.64        | 6968             |
|                       | 0.733        | 0.701        | 0.811        | 0.718        | 0.765        | 0.739        | 0.847        | 0.755        | 0.499        | 0.488        | 0.594        | 0.512        | 0.599        | 0.631        | 0.572        | 0.631        |                  |
| TTNN+ $\ell_1$ [24]   | 24.77        | 22.39        | 28.13        | 24.94        | 27.29        | 24.19        | 30.57        | 26.93        | 23.02        | 21.32        | 25.82        | 22.82        | 25.51        | 23.64        | 27.54        | 24.99        | 5843             |
|                       | 0.758        | 0.734        | 0.833        | 0.753        | 0.868        | 0.782        | 0.911        | 0.791        | 0.609        | 0.494        | 0.649        | 0.567        | 0.605        | 0.577        | 0.632        | 0.602        |                  |
| TSP- $k+\ell_1$ [26]  | 26.59        | 23.46        | 28.68        | 25.46        | 29.42        | 24.98        | 31.55        | 27.48        | 24.39        | 21.81        | 26.08        | 23.24        | 26.58        | 23.64        | 27.85        | 24.85        | 9924             |
|                       | 0.819        | 0.741        | 0.845        | 0.808        | 0.886        | 0.813        | 0.912        | 0.835        | 0.643        | 0.526        | 0.686        | 0.593        | 0.633        | 0.578        | 0.637        | 0.617        |                  |
| LNOP [28]             | 26.67        | 23.31        | 29.13        | 25.91        | 28.03        | 23.69        | 29.55        | 28.39        | 24.66        | 22.39        | 26.85        | 24.43        | 26.24        | 23.78        | 27.70        | 25.95        | 9138             |
|                       | 0.830        | 0.748        | 0.857        | 0.829        | 0.768        | 0.742        | 0.861        | 0.832        | 0.638        | 0.533        | 0.694        | 0.603        | 0.609        | 0.603        | 0.634        | 0.623        |                  |
| NRTRM [29]            | 26.52        | 23.94        | 29.06        | 25.16        | 28.91        | 25.96        | 31.61        | 27.27        | 24.33        | 22.19        | 26.21        | 22.89        | 26.49        | 24.46        | 27.86        | 25.19        | 7834             |
|                       | 0.831        | 0.772        | 0.855        | 0.776        | 0.891        | 0.838        | 0.919        | 0.868        | 0.642        | 0.538        | 0.706        | 0.605        | 0.639        | 0.611        | 0.661        | 0.632        |                  |
| BCNRTC [31]           | 26.39        | 23.76        | 28.77        | 25.01        | 28.80        | 25.61        | 31.04        | 26.93        | 24.54        | 22.30        | 26.47        | 22.99        | 26.39        | 24.41        | 27.67        | 24.99        | 7180             |
|                       | 0.835        | 0.771        | 0.868        | 0.828        | 0.883        | 0.819        | 0.908        | 0.868        | 0.639        | 0.544        | 0.709        | 0.598        | 0.645        | 0.597        | 0.678        | 0.632        |                  |
| HWTNN+ $\ell_1$ [64]  | 29.23        | 26.09        | <b>32.22</b> | 28.22        | <b>31.25</b> | 28.16        | <b>34.24</b> | <b>30.43</b> | 26.16        | 23.34        | 28.38        | 24.66        | 27.49        | 25.56        | <b>28.72</b> | 26.27        | 6737             |
|                       | 0.839        | 0.777        | <b>0.876</b> | 0.838        | <b>0.893</b> | 0.843        | <b>0.923</b> | <b>0.876</b> | 0.663        | 0.551        | 0.718        | 0.627        | 0.647        | 0.619        | <b>0.687</b> | 0.643        |                  |
| HWTSN+w $\ell_q$      | <b>30.27</b> | <b>27.38</b> | <b>33.68</b> | <b>30.02</b> | <b>32.36</b> | <b>29.45</b> | <b>35.33</b> | <b>32.07</b> | <b>26.86</b> | 24.01        | 28.62        | <b>26.89</b> | <b>27.92</b> | <b>26.53</b> | <b>28.98</b> | <b>27.94</b> | 8760             |
|                       | <b>0.851</b> | <b>0.794</b> | <b>0.885</b> | <b>0.867</b> | <b>0.905</b> | <b>0.868</b> | <b>0.931</b> | <b>0.907</b> | 0.668        | 0.554        | 0.741        | <b>0.707</b> | <b>0.659</b> | <b>0.638</b> | <b>0.695</b> | <b>0.667</b> |                  |
| HWTSN+w $\ell_q$ (BR) | 29.15        | 26.66        | 31.27        | 28.89        | 29.85        | 28.06        | 30.66        | 29.46        | 26.50        | <b>24.19</b> | <b>29.21</b> | 26.19        | 27.60        | 26.16        | 28.43        | 27.38        | 4243             |
|                       | 0.841        | 0.784        | 0.868        | 0.850        | 0.868        | 0.844        | 0.879        | 0.869        | <b>0.686</b> | <b>0.579</b> | <b>0.755</b> | 0.688        | 0.649        | 0.623        | 0.679        | 0.658        |                  |
| HWTSN+w $\ell_q$ (UR) | <b>29.28</b> | <b>26.77</b> | 31.41        | <b>29.03</b> | 29.93        | <b>28.19</b> | 30.76        | 29.53        | <b>26.60</b> | <b>24.27</b> | <b>29.27</b> | <b>26.30</b> | <b>27.68</b> | <b>26.21</b> | 28.45        | <b>27.48</b> | 4374             |
|                       | <b>0.842</b> | <b>0.789</b> | 0.865        | <b>0.852</b> | 0.869        | <b>0.844</b> | 0.878        | 0.870        | <b>0.688</b> | <b>0.582</b> | <b>0.758</b> | <b>0.692</b> | <b>0.651</b> | <b>0.625</b> | 0.684        | <b>0.659</b> |                  |

In each RLRTC method, the top represents the PSNR values while the bottom denotes the SSIM values.

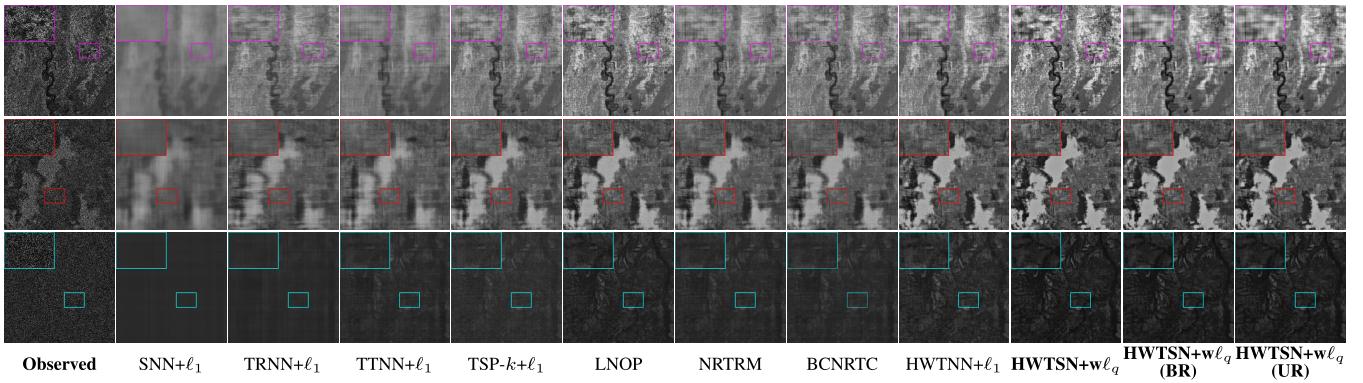


Fig. 3. Visual comparison of various methods for MRSIs inpainting. From top to bottom, the parameter pair ( $sr, \tau$ ) are (0.4, 0.1), (0.4, 0.3), and (0.4, 0.5), respectively. Top row: the (5, 1)-th frame of Landsat-7. Middle row: the (2, 6)-th frame of SPOT-5. Bottom row: the (3, 5)-th frame of T22LGN.

TABLE IV

THE PSNR, SSIM VALUES AND CPU TIMEOBTAINED BY VARIOUS RLRTC METHODS FOR DIFFERENT FOURTH-ORDER MRSIs. THE BEST AND THE SECOND-BEST RESULTS ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY

| MRSI-Name            | Landsat-7    |              |              |              |                   |              | T22LGN       |              |              |              |              |              | SPOT-5       |              |              |              |              |              | Average Time (s) |
|----------------------|--------------|--------------|--------------|--------------|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|------------------|
|                      | $sr$         |              | 20%          |              | 40%               |              | $sr$         |              | 20%          |              | 40%          |              | $sr$         |              | 20%          |              | 40%          |              |                  |
|                      | 10%          | 30%          | 50%          | 10%          | 30%               | 50%          | 10%          | 30%          | 50%          | 10%          | 30%          | 50%          | 10%          | 30%          | 50%          | 10%          | 30%          | 50%          |                  |
| SNN+ $\ell_1$ [11]   | 21.45        | 20.86        | 20.13        | 23.12        | 22.49             | 21.44        | 25.94        | 25.91        | 25.69        | 27.01        | 26.68        | 25.76        | 22.38        | 21.58        | 20.91        | 25.63        | 24.19        | 22.55        | 23817            |
|                      | 0.537        | 0.453        | 0.386        | 0.576        | 0.498             | 0.477        | 0.637        | 0.605        | 0.592        | 0.718        | 0.666        | 0.596        | 0.573        | 0.509        | 0.395        | 0.631        | 0.563        | 0.492        |                  |
| TRNN+ $\ell_1$ [15]  | 23.96        | 22.32        | 20.70        | 24.61        | 22.67             | 21.30        | 28.57        | 27.22        | 25.02        | 29.15        | 27.52        | 25.89        | 27.06        | 24.55        | 22.16        | 28.19        | 25.07        | 23.11        | 22414            |
|                      | 0.671        | 0.589        | 0.565        | 0.699        | 0.691             | 0.614        | 0.716        | 0.696        | 0.586        | 0.797        | 0.723        | 0.666        | 0.681        | 0.568        | 0.467        | 0.716        | 0.584        | 0.523        |                  |
| TTNN+ $\ell_1$ [24]  | 23.17        | 21.89        | 20.19        | 24.05        | 22.61             | 21.15        | 29.88        | 28.34        | 24.58        | 31.26        | 29.37        | 26.29        | 26.49        | 24.38        | 22.22        | 28.17        | 25.65        | 23.49        | 10107            |
|                      | 0.652        | 0.543        | 0.458        | 0.739        | 0.691             | 0.575        | 0.793        | 0.778        | 0.497        | 0.817        | 0.797        | 0.692        | 0.684        | 0.537        | 0.431        | 0.701        | 0.596        | 0.519        |                  |
| TSP- $k+\ell_1$ [26] | 24.07        | 22.41        | 20.36        | 25.82        | 23.62             | 21.14        | 29.12        | 25.22        | 22.54        | 31.14        | 26.03        | 22.59        | 27.07        | 24.57        | 22.29        | 29.74        | 26.36        | 23.34        | 27412            |
|                      | 0.673        | 0.588        | 0.461        | 0.742        | 0.628             | 0.472        | 0.791        | 0.511        | 0.409        | 0.836        | 0.545        | 0.419        | 0.657        | 0.571        | 0.413        | 0.775        | 0.601        | 0.506        |                  |
| LNOP [28]            | 24.61        | 22.72        | 20.83        | 26.27        | 23.35             | 21.42        | 30.98        | 28.61        | 26.43        | 33.63        | 29.12        | 26.89        | 27.66        | 25.55        | 23.45        | 29.88        | 26.63        | 24.66        | 18418            |
|                      | 0.679        | 0.585        | 0.589        | 0.763        | 0.688             | 0.592        | 0.841        | 0.736        | 0.703        | 0.919        | 0.825        | 0.706        | 0.643        | 0.593        | 0.484        | 0.781        | 0.632        | 0.572        |                  |
| NRTRM [29]           | 23.76        | 22.40        | 20.45        | 25.72        | 23.84             | 21.33        | 30.34        | 28.67        | 25.68        | 33.12        | 30.41        | 25.87        | 27.08        | 24.84        | 22.55        | 29.95        | 26.84        | 23.73        | 18560            |
|                      | 0.669        | 0.626        | 0.546        | 0.757        | 0.639             | 0.577        | 0.833        | 0.782        | 0.607        | 0.893        | 0.844        | 0.621        | 0.682        | 0.589        | 0.432        | 0.799        | 0.645        | 0.569        |                  |
| BCNRTC [31]          | 24.46        | 22.19        | 20.22        | 26.51        | 23.55             | 21.29        | 30.98        | 26.95        | 25.88        | 33.76        | 30.05        | 26.21        | 27.82        | 24.32        | 22.65        | 30.41        | 27.04        | 23.97        | 11380            |
|                      | 0.677        | 0.612        | 0.394        | 0.757        | 0.693             | 0.598        | 0.846        | 0.677        | 0.548        | 0.914        | 0.834        | 0.643        | 0.685        | 0.551        | 0.418        | 0.798        | 0.695        | 0.568        |                  |
| HWTNN+ $\ell_1$ [64] | <b>25.18</b> | 23.48        | 21.59        | <b>26.33</b> | 24.98             | 22.89        | <b>32.09</b> | 29.19        | 25.53        | <b>34.24</b> | <b>31.46</b> | 26.46        | <b>28.45</b> | 26.24        | 23.61        | <b>30.47</b> | <b>28.36</b> | 25.49        | 14835            |
|                      | <b>0.692</b> | 0.659        | 0.630        | <b>0.768</b> | 0.693             | 0.665        | <b>0.854</b> | 0.758        | 0.639        | <b>0.923</b> | <b>0.852</b> | 0.706        | <b>0.687</b> | 0.596        | 0.490        | <b>0.808</b> | <b>0.699</b> | 0.581        |                  |
| HWTSN+w $\ell_q$     | <b>25.99</b> | <b>24.71</b> | <b>23.21</b> | <b>28.11</b> | <b>25.99</b>      | <b>23.61</b> | <b>32.46</b> | <b>30.16</b> | <b>27.98</b> | <b>34.98</b> | <b>31.92</b> | <b>28.64</b> | <b>29.58</b> | <b>27.79</b> | <b>25.82</b> | <b>32.47</b> | <b>30.07</b> | <b>26.77</b> | 18974            |
|                      | <b>0.722</b> | <b>0.683</b> | <b>0.669</b> | <b>0.777</b> | <b>0.724</b> </td |              |              |              |              |              |              |              |              |              |              |              |              |              |                  |

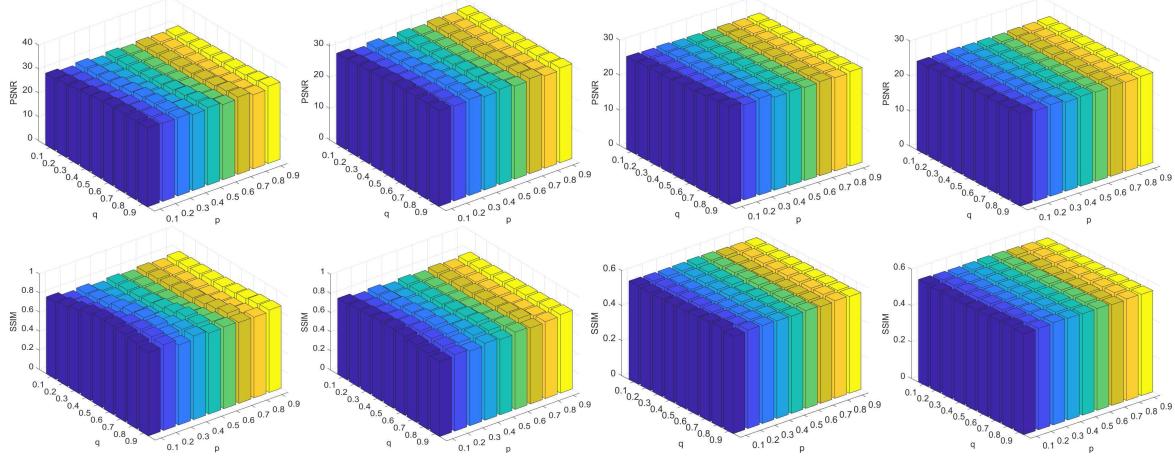


Fig. 4. The influence of parameters  $p$  and  $q$  upon MRSIs inpainting. From left to right, corresponding algorithm and parameters pair  $(sr, \tau)$  are HWTNSN+w $\ell_q$ : (0.4, 0.3), randomized version ( $k = 70$ ): (0.4, 0.3), HWTNSN+w $\ell_q$ : (0.2, 0.5), and randomized version ( $k = 70$ ): (0.2, 0.5), respectively.

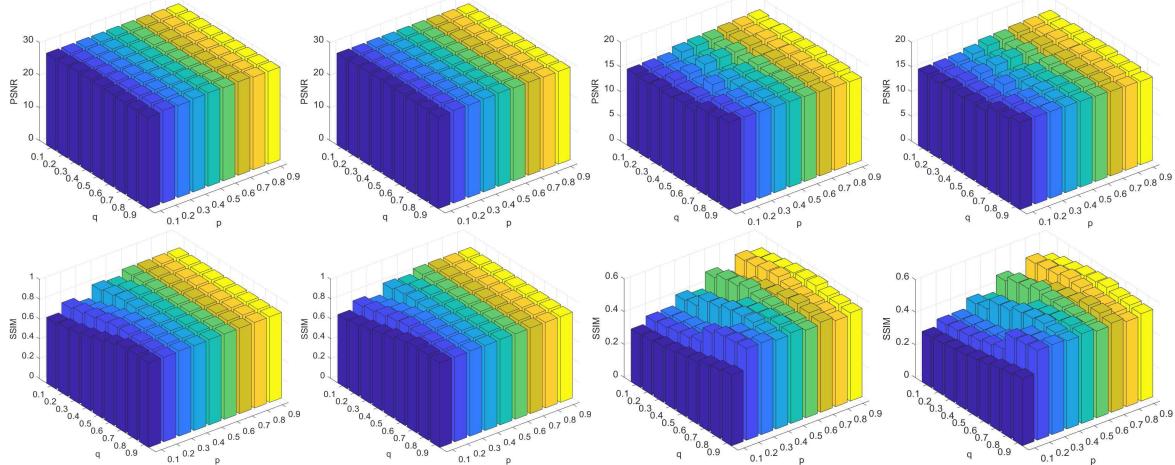


Fig. 5. The influence of parameters  $p$  and  $q$  upon LFI recovery. From left to right, corresponding algorithm and parameter pair  $(sr, \tau)$  are HWTNSN+w $\ell_q$ : (0.05, 0.3), randomized version ( $k = 50$ ): (0.05, 0.3), HWTNSN+w $\ell_q$ : (0.01, 0.5), and randomized version ( $k = 50$ ): (0.01, 0.5), respectively.

third-order T-SVD, although the ones employing nonconvex schemes (i.e., LNOP, BCNRTC and NRTRM) require more running time over the ones utilizing convex methods (i.e., TTNN+ $\ell_1$  and TSP- $k+\ell_1$ ), they achieve higher PSNR and SSIM values in most cases. This phenomenon also exists in the methods based on high-order T-SVD. **3)** Compared with other methods, the HWTNSN+w $\ell_q$  achieves an approximately 2~5 dB gain in the mean PSNR values, and its accelerated versions obtains an about 40%~60% percent drop in the average CPU time. **4)** In our algorithms, the versions fused randomization ideas shorten the running time by about 55% percent over the deterministic version with a slight reduction of psnr and ssim values. **5)** There is no significant difference between the two proposed randomized RHTC algorithms in PSNR, SSIM, and CPU time. It is very likely that only for very large tensors will the computational advantage of the HWTNSN+w $\ell_q$ (BR) method using blocked randomized schemes become apparent.

**2) Application in CVs Restoration:** In this experiment, four fourth-order CVs are used to evaluate the performance of the proposed algorithm. Figure 2 displays the visual comparison of the proposed and competitive RLRTC algorithms for various CVs restoration at sampling rates  $sr = 10\%$ ,  $20\%$  and noise levels  $\tau = 30\%$ ,  $50\%$ . From the zoomed regions, we can see that the HWTNSN+w $\ell_q$  exhibits tangibly better restoration quality over other comparative methods according to the color,

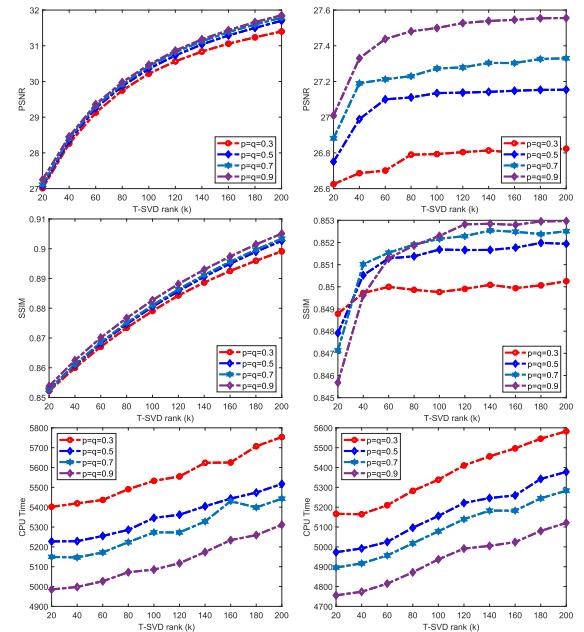


Fig. 6. The influence of various rank parameters  $k$  upon MRSIs inpainting. From left to right, the parameter pair  $(sr, \tau)$  are (0.4, 0.1) and (0.2, 0.5), respectively.

brightness, and outline. In Table III, we report the PSNR, SSIM values and CPU time of ten RLRTC methods for four

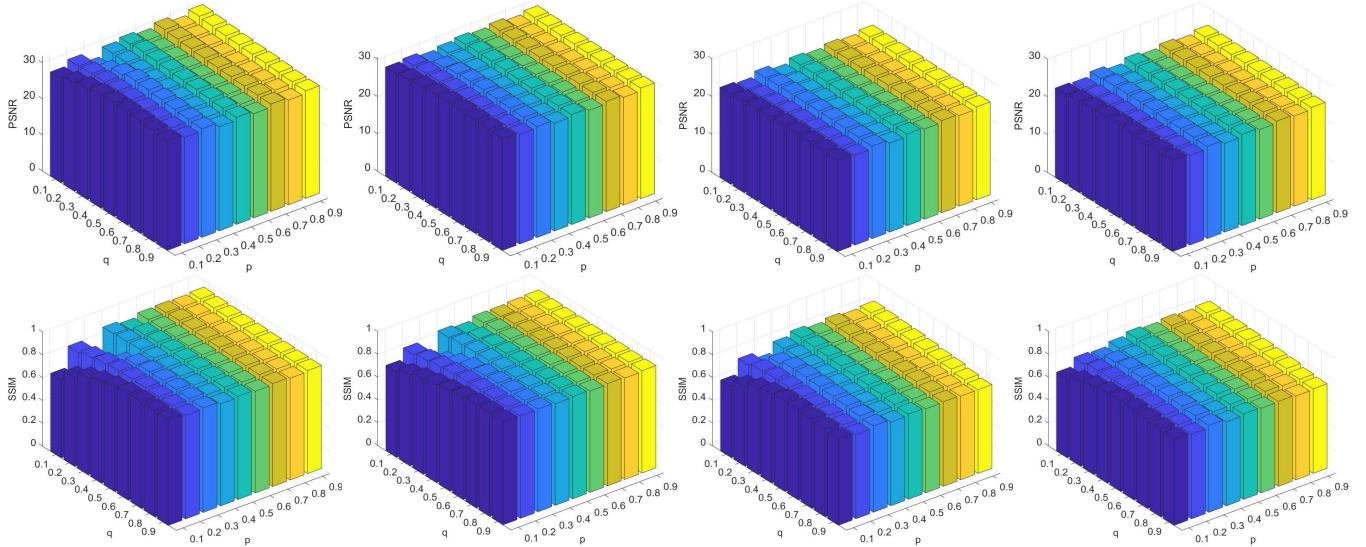


Fig. 7. The influence of parameters  $p$  and  $q$  upon CVs restoration. From left to right, corresponding algorithm and parameter pair ( $sr, \tau$ ) are HWTNSN+ $w\ell_q$ : (0.1, 0.3), randomized version ( $k = 50$ ): (0.1, 0.3), HWTNSN+ $w\ell_q$ : (0.05, 0.5), and randomized version ( $k = 50$ ): (0.05, 0.5), respectively.

CVs, where  $sr = 0.1, 0.2$  and  $\tau = 0.3, 0.5$ . These results show that the PSNR and SSIM metrics acquired by HWTNSN+ $w\ell_q$  are higher than those obtained by the baseline method, i.e., HWTNN+ $\ell_1$ . In contrast to the competitive non-convex methods (i.e., LNOP, NRTRM and BCNRTC), the improvements of proposed non-convex algorithm (i.e., HWTNSN+ $w\ell_q$ ) are around 3 dB in term of PSNR index while the reductions of its randomized version are about 52% according to the CPU time. Furthermore, under the comprehensive balance of PSNR, SSIM, and CPU time, the proposed randomized RHTC method is always superior to other popular algorithms. Other findings are similar to the case of LFI recovery.

**3) Application in MRSIs Inpainting:** This experiment mainly utilizes three fourth-order MRSIs to evaluate the performance of proposed and compared algorithms. Table IV presents the PSNR, SSIM values and CPU time provided by ten RLRTC methods on three large-scale MRSIs with observation rates:  $sr \in \{0.4, 0.2\}$ , noise ratios:  $\tau \in \{0.1, 0.3, 0.5\}$ . Accordingly, some visual examples are illustrated in Figure 3, which indicates that the proposed method retains more details and textures over the other state-of-the-art approaches. Strikingly, in comparison with the deterministic RLRTC algorithms, our RHTC algorithms incorporating with randomized technology can decrease the computational time by about 70% with little or no loss of PSNR and SSIM. This demonstrates the effectiveness of randomized approach for processing large-scale tensor data. Other conclusions achieved from the quantitative results are similar to those obtained from the tasks of CVs restoration and LFI recovery. Overall, the proposed randomized RHTC method can dramatically shorten the CPU running time while still achieving reasonable recovery precision over other popular approaches, especially for large-scale inpainting tasks.

**4) Parameters Analysis:** First, we analyze how different values of  $p$  and  $q$  influence the recovery performance of our proposed RHTC algorithms (i.e., HWTNSN+ $w\ell_q$  and its two randomized versions) under three types of real-world tensors with different sampling rates and impulse noise levels. To save the computational time, the spatial size (i.e., the dimensions of the first two modes) of LFI, CVs, and MRSIs are down-sampled to  $217 \times 312$ ,  $360 \times 640$  and  $500 \times 500$ , respectively.

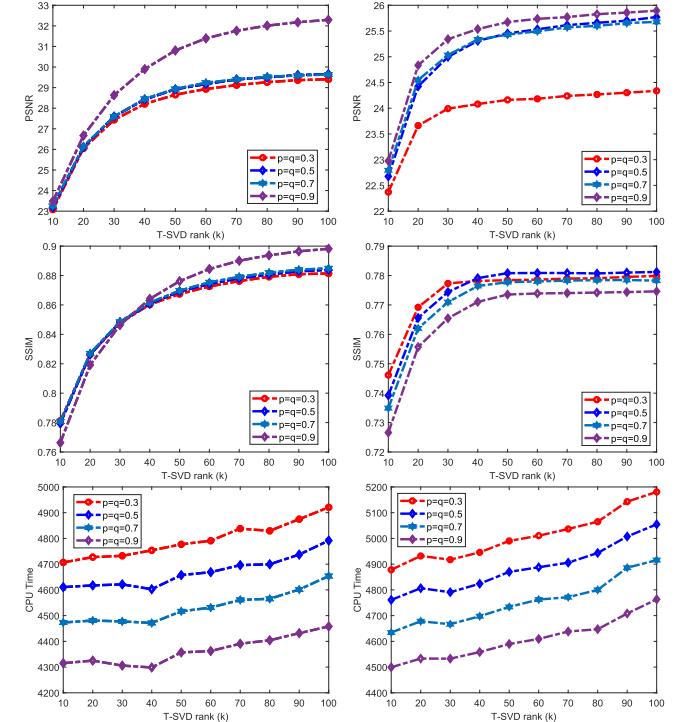


Fig. 8. The influence of various rank parameters  $k$  upon CVs restoration. From left to right, the parameter pair ( $sr, \tau$ ) are (0.2, 0.3) and (0.1, 0.5), respectively.

Relevant experimental results in the supplementary material have shown that FFT has performed very well in various recovery tasks compared to other linear transforms. Therefore, we only choose FFT as the transform  $\mathcal{L}$  in our experiments. Besides, the values  $p$  and  $q$  are set from 0.1 to 0.9 with an interval of 0.1. The other parameters are set to be the same as the ones mentioned earlier. The corresponding experimental results are presented in Figure 4, 5, 7. Since the estimated precision obtained by the two randomized methods are very similar, we only show the results of HWTNSN+ $w\ell_q$ (UR). From these results, we can find that the difference between most of the PSNR or SSIM values obtained by our randomized or deterministic RHTC method is not very large under various

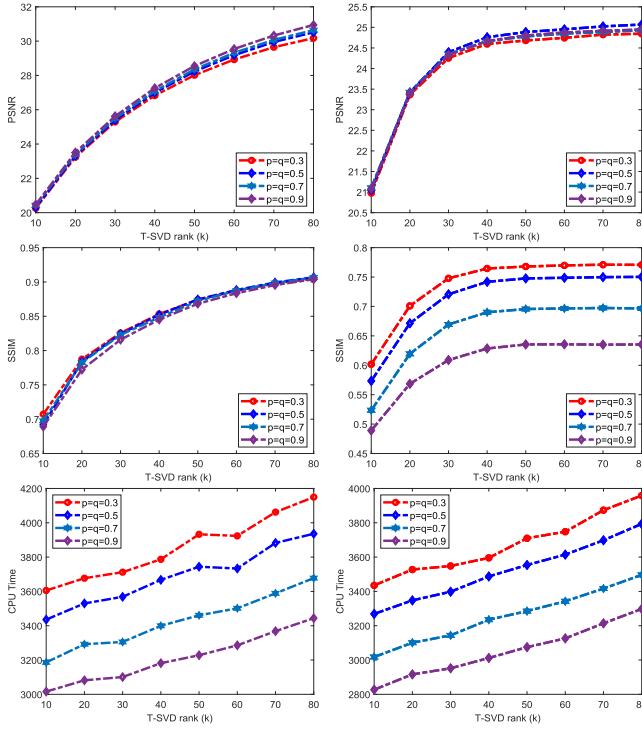


Fig. 9. The influence of various rank parameters  $k$  upon LFIs recovery. From left to right, the parameter pair  $(sr, \tau)$  are  $(0.1, 0.3)$  and  $(0.05, 0.5)$ , respectively.

parameters  $(p, q)$  corresponding to each group of fixed  $(sr, \tau)$ . When the missing rate and noise level are quite high, selecting relatively large  $p$  and  $q$  may result in relatively better performance in most cases for different inpainting tasks.

Following the above experiments, we then investigate the influence of different rank parameter  $k$  upon restoration results of our randomized RHTC algorithms for three categories of real-world tensors with different noise levels and observed ratios. The sizes corresponding to all tensor data utilized in this experiment are the same as those used in subsection VI-A.1–VI-A.3. In our experiments, we set  $p = q = 0.3, 0.5, 0.7, 0.9$ ,  $\mathcal{L} = \text{FFT}$  for simplicity. The other parameters are set to be the same as the ones mentioned earlier. Because two randomized versions: HWTSN+ $w\ell_q(\text{BR})$  and HWTSN+ $w\ell_q(\text{UR})$  have little difference in accuracy and efficiency, we only show the results of the latter. From the experimental results shown in Figure 6, 8, 9, the following conclusions can be drawn: 1) relatively better recovery performance can be achieved by selecting relatively small T-SVD rank  $k$ , i.e.,  $k \ll \min(n_1, n_2)$ ; 2) with the increase of adjustable parameters  $p$  and  $q$ , the PSNR and SSIM values obtained by our randomized RHTC method slightly increase whereas the CPU time slightly decrease in most cases for various recovery tasks.

## VII. CONCLUSION AND FUTURE WORK

In this article, we first suggest two efficient LRTA methods, which are developed by fusing unblocked and blocked random projection schemes, respectively. On this basis, we continue to investigate fast and effective algorithms for solving the RHTC problem modeled by nonconvex low-rank and noise/outliers regularizers. Theoretical results on error bounds for the LRTA algorithms and convergence analysis for the RHTC algorithms are provided. The model construction, algorithm design and

theoretical analysis are all based on the high-order T-SVD framework. Extensive experiments on both synthetic and real-world tensor data have verified the effectiveness and superiority of the proposed LRTA and RHTC approaches. This work will lay the foundation for many tensor-based data analysis tasks such as high-order tensor clustering, regression, classification, etc.

In the future, by devising novel mode-wise projection/sampling methods that incorporate advanced sketching techniques [99], we first intend to explore the effective randomized algorithms for the fixed-precision LRTA under the T-SVD framework. On this basis, we further investigate the problems of large-scale high-order tensor recovery, clustering, regression and classification from the perspective of model, algorithm and theory. Secondly, we plan to extend the above batch-based randomized methods to the online versions, which can deal with large-scale streaming tensors incrementally in online mode, and even with dynamically changing tensors. Thirdly, we will explore high-performance data-driven algorithms for LRTA in the streaming, sliding window, or distributed model by incorporating advanced sketching techniques into deep neural network.

## REFERENCES

- [1] C. F. Beckmann and S. M. Smith, “Tensorial extensions of independent component analysis for multisubject FMRI analysis,” *NeuroImage*, vol. 25, no. 1, pp. 294–311, Mar. 2005.
- [2] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, “Quantum-chemical insights from deep tensor neural networks,” *Nature Commun.*, vol. 8, no. 1, p. 13890, Jan. 2017.
- [3] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, “Tensors for data mining and data fusion: Models, applications, and scalable algorithms,” *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 2, pp. 1–44, Mar. 2017.
- [4] 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, Jul. 2017.
- [5] M. Marquez, H. Rueda-Chacon, and H. Arguello, “Compressive spectral light field image reconstruction via online tensor representation,” *IEEE Trans. Image Process.*, vol. 29, pp. 3558–3568, 2020.
- [6] J. Lin, T.-Z. Huang, X.-L. Zhao, Y. Chen, Q. Zhang, and Q. Yuan, “Robust thick cloud removal for multitemporal remote sensing images using coupled tensor factorization,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5406916.
- [7] Z. Long, C. Zhu, J. Liu, and Y. Liu, “Bayesian low rank tensor ring for image recovery,” *IEEE Trans. Image Process.*, vol. 30, pp. 3568–3580, 2021.
- [8] A. Bibi and B. Ghanem, “High order tensor formulation for convolutional sparse coding,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1790–1798.
- [9] X. Zhang, D. Wang, Z. Zhou, and Y. Ma, “Robust low-rank tensor recovery with rectification and alignment,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 238–255, Jan. 2021.
- [10] J. Hou, F. Zhang, H. Qiu, J. Wang, Y. Wang, and D. Meng, “Robust low-tubal-rank tensor recovery from binary measurements,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4355–4373, Aug. 2022.
- [11] D. Goldfarb and Z. Qin, “Robust low-rank tensor recovery: Models and algorithms,” *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 1, pp. 225–253, Jan. 2014.
- [12] B. Huang, C. Mu, D. Goldfarb, and J. Wright, “Provable models for robust low-rank tensor completion,” *Pacific J. Optim.*, vol. 11, no. 2, pp. 339–364, 2015.
- [13] 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, Apr. 2016.
- [14] W. Chen, X. Gong, and N. Song, “Nonconvex robust low-rank tensor reconstruction via an empirical Bayes method,” *IEEE Trans. Signal Process.*, vol. 67, no. 22, pp. 5785–5797, Nov. 2019.

- [15] H. Huang, Y. Liu, Z. Long, and C. Zhu, "Robust low-rank tensor ring completion," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 1117–1126, 2020.
- [16] C. Chen, Z.-B. Wu, Z.-T. Chen, Z.-B. Zheng, and X.-J. Zhang, "Auto-weighted robust low-rank tensor completion via tensor-train," *Inf. Sci.*, vol. 567, pp. 100–115, Aug. 2021.
- [17] Q. Liu, X. Li, H. Cao, and Y. Wu, "From simulated to visual data: A robust low-rank tensor completion approach using  $\ell_p$ -regression for outlier resistance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3462–3474, Jun. 2022.
- [18] X. P. Li and H. C. So, "Robust low-rank tensor completion based on tensor ring rank via  $\ell_{p,\epsilon}$ -norm," *IEEE Trans. Signal Process.*, vol. 69, pp. 3685–3698, 2021.
- [19] X. P. Li, Z.-Y. Wang, Z.-L. Shi, H. C. So, and N. D. Sidiropoulos, "Robust tensor completion via capped Frobenius norm," *IEEE Trans. Neural Netw. Learn. Syst.*, 2023, doi: [10.1109/TNNLS.2023.3236415](https://doi.org/10.1109/TNNLS.2023.3236415).
- [20] Y. He and G. K. Atia, "Coarse to fine two-stage approach to robust tensor completion of visual data," *IEEE Trans. Cybern.*, vol. 54, no. 1, pp. 136–149, Jan. 2024, doi: [10.1109/TCYB.2022.3198932](https://doi.org/10.1109/TCYB.2022.3198932).
- [21] Q. Jiang and M. Ng, "Robust low-tubal-rank tensor completion via convex optimization," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2649–2655.
- [22] A. Wang, Z. Jin, and G. Tang, "Robust tensor decomposition via t-SVD: Near-optimal statistical guarantee and scalable algorithms," *Signal Process.*, vol. 167, Feb. 2020, Art. no. 107319.
- [23] A. Wang, X. Song, X. Wu, Z. Lai, and Z. Jin, "Robust low-tubal-rank tensor completion," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3432–3436.
- [24] G. Song, M. K. Ng, and X. Zhang, "Robust tensor completion using transformed tensor singular value decomposition," *Numer. Linear Algebra Appl.*, vol. 27, no. 3, p. e2299, May 2020.
- [25] M. K. Ng, X. Zhang, and X.-L. Zhao, "Patched-tube unitary transform for robust tensor completion," *Pattern Recognit.*, vol. 100, Apr. 2020, Art. no. 107181.
- [26] J. Lou and Y.-M. Cheung, "Robust low-rank tensor minimization via a new tensor spectral  $k$ -support norm," *IEEE Trans. Image Process.*, vol. 29, pp. 2314–2327, 2020.
- [27] Y. He and G. K. Atia, "Robust low-tubal-rank tensor completion based on tensor factorization and maximum correntropy criterion," *IEEE Trans. Neural Netw. Learn. Syst.*, 2023, doi: [10.1109/TNNLS.2023.3280086](https://doi.org/10.1109/TNNLS.2023.3280086).
- [28] L. Chen, X. Jiang, X. Liu, and Z. Zhou, "Robust low-rank tensor recovery via nonconvex singular value minimization," *IEEE Trans. Image Process.*, vol. 29, pp. 9044–9059, 2020.
- [29] D. Qiu, M. Bai, M. K. Ng, and X. Zhang, "Nonlocal robust tensor recovery with nonconvex regularization," *Inverse Problems*, vol. 37, no. 3, Mar. 2021, Art. no. 035001.
- [30] X. Zhao, M. Bai, and M. K. Ng, "Nonconvex optimization for robust tensor completion from grossly sparse observations," *J. Sci. Comput.*, vol. 85, no. 2, p. 46, Nov. 2020.
- [31] X. Zhao, M. Bai, D. Sun, and L. Zheng, "Robust tensor completion: Equivalent surrogates, error bounds, and algorithms," *SIAM J. Imag. Sci.*, vol. 15, no. 2, pp. 625–669, Jun. 2022.
- [32] 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, Apr. 2020.
- [33] Q. Gao, P. Zhang, W. Xia, D. Xie, X. Gao, and D. Tao, "Enhanced tensor RPCA and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 2133–2140, Jun. 2021.
- [34] Q. Shi, Y.-M. Cheung, and J. Lou, "Robust tensor SVD and recovery with rank estimation," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10667–10682, Oct. 2022.
- [35] M. Yang, Q. Luo, W. Li, and M. Xiao, "Nonconvex 3D array image data recovery and pattern recognition under tensor framework," *Pattern Recognit.*, vol. 122, Feb. 2022, Art. no. 108311.
- [36] M. Che, X. Wang, Y. Wei, and X. Zhao, "Fast randomized tensor singular value thresholding for low-rank tensor optimization," *Numer. Linear Algebra Appl.*, vol. 29, no. 6, p. e2444, Dec. 2022.
- [37] 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, Oct. 2021.
- [38] F. Zhang, L. Yang, J. Wang, and X. Luo, "Randomized sampling techniques based low-tubal-rank plus sparse tensor recovery," *Knowl.-Based Syst.*, vol. 261, Feb. 2023, Art. no. 110198.
- [39] M. Li, W. Li, Y. Chen, and M. Xiao, "The nonconvex tensor robust principal component analysis approximation model via the weighted  $\ell_p$ -norm regularization," *J. Sci. Comput.*, vol. 89, no. 3, p. 67, Dec. 2021.
- [40] H. Qiu, Y. Wang, S. Tang, D. Meng, and Q. Yao, "Fast and provable nonconvex tensor RPCA," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 18211–18249.
- [41] J. Wang, J. Hou, and Y. C. Eldar, "Tensor robust principal component analysis from multilevel quantized observations," *IEEE Trans. Inf. Theory*, vol. 69, no. 1, pp. 383–406, Jan. 2023.
- [42] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-Y. Ji, and T.-H. Ma, "Tensor  $N$ -tubal rank and its convex relaxation for low-rank tensor recovery," *Inf. Sci.*, vol. 532, pp. 170–189, Sep. 2020.
- [43] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2017.
- [44] H. Kong, X. Xie, and Z. Lin, "t-Schatten- $p$  norm for low-rank tensor recovery," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1405–1419, Dec. 2018.
- [45] 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, Aug. 2022.
- [46] H. Xu, J. Zheng, X. Yao, Y. Feng, and S. Chen, "Fast tensor nuclear norm for structured low-rank visual inpainting," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 538–552, Feb. 2022.
- [47] H. Wang, F. Zhang, J. Wang, T. Huang, J. Huang, and X. Liu, "Generalized nonconvex approach for low-tubal-rank tensor recovery," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3305–3319, Aug. 2021.
- [48] L. Feng, C. Zhu, Z. Long, J. Liu, and Y. Liu, "Multiplex transformed tensor decomposition for multidimensional image recovery," *IEEE Trans. Image Process.*, vol. 32, pp. 3397–3412, 2023, doi: [10.1109/TIP.2023.3284673](https://doi.org/10.1109/TIP.2023.3284673).
- [49] 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, Jan. 2013.
- [50] 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. Image Process.*, vol. 26, no. 5, pp. 2466–2479, May 2017.
- [51] X.-L. Zhao, J.-H. Yang, T.-H. Ma, T.-X. Jiang, M. K. Ng, and T.-Z. Huang, "Tensor completion via complementary global, local, and non-local priors," *IEEE Trans. Image Process.*, vol. 31, pp. 984–999, 2022.
- [52] J. Xue, Y. Zhao, Y. Bu, J. C. Chan, and S. G. Kong, "When Laplacian scale mixture meets three-layer transform: A parametric tensor sparsity for tensor completion," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13887–13901, Dec. 2022.
- [53] Y. Qiu, G. Zhou, Q. Zhao, and S. Xie, "Noisy tensor completion via low-rank tensor ring," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 1127–1141, Jan. 2024, doi: [10.1109/TNNLS.2022.3181378](https://doi.org/10.1109/TNNLS.2022.3181378).
- [54] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev. Soc. Ind. Appl. Math.*, vol. 51, no. 3, pp. 455–500, Aug. 2009.
- [55] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, Sep. 1966.
- [56] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, Jan. 2011.
- [57] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, "Tensor ring decomposition," 2016, *arXiv:1606.05535*.
- [58] E. Kilmer, M. Kilmer, and S. Aeron, "Tensor–tensor products with invertible linear transforms," *Linear Algebra Appl.*, vol. 485, pp. 545–570, Nov. 2015.
- [59] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, Aug. 2011.
- [60] 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.
- [61] M. E. Kilmer, L. Horesh, H. Avron, and E. Newman, "Tensor–tensor algebra for optimal representation and compression of multiway data," *Proc. Nat. Acad. Sci. USA*, vol. 118, no. 28, Jul. 2021, Art. no. e2015851118.
- [62] W. Qin, H. Wang, F. Zhang, J. Wang, X. Luo, and T. Huang, "Low-rank high-order tensor completion with applications in visual data," *IEEE Trans. Image Process.*, vol. 31, pp. 2433–2448, 2022.
- [63] W. Qin, H. Wang, W. Ma, and J. Wang, "Robust high-order tensor recovery via nonconvex low-rank approximation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 3633–3637.

- [64] W. Qin, H. Wang, F. Zhang, M. Dai, and J. Wang, "Robust low-rank tensor reconstruction using high-order t-SVD," *J. Electron. Imag.*, vol. 30, no. 6, Nov. 2021, Art. no. 063016.
- [65] H. Wang, J. Peng, W. Qin, J. Wang, and D. Meng, "Guaranteed tensor recovery fused low-rankness and smoothness," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10990–11007, Sep. 2023, doi: [10.1109/TPAMI.2023.3259640](https://doi.org/10.1109/TPAMI.2023.3259640).
- [66] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, Jan. 2011.
- [67] M. W. Mahoney et al., "Randomized algorithms for matrices and data," *Found. Trends Mach. Learn.*, vol. 3, no. 2, pp. 123–224, 2011.
- [68] D. P. Woodruff, "Sketching as a tool for numerical linear algebra," *Found. Trends Theor. Comput. Sci.*, vol. 10, nos. 1–2, pp. 1–157, 2014.
- [69] P.-G. Martinsson and J. A. Tropp, "Randomized numerical linear algebra: Foundations and algorithms," *Acta Numerica*, vol. 29, pp. 403–572, May 2020.
- [70] A. Buluc et al., "Randomized algorithms for scientific computing (RASC)," 2021, *arXiv:2104.11079*.
- [71] Y. Wang, H.-Y. Tung, A. J. Smola, and A. Anandkumar, "Fast and guaranteed tensor decomposition via sketching," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 28, 2015, pp. 991–999.
- [72] C. Battaglino, G. Ballard, and T. G. Kolda, "A practical randomized CP tensor decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 39, no. 2, pp. 876–901, 2018.
- [73] B. W. Larsen and T. G. Kolda, "Practical leverage-based sampling for low-rank tensor decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 43, no. 3, pp. 1488–1517, Sep. 2022.
- [74] O. A. Malik and S. Becker, "Low-rank tucker decomposition of large tensors using tensorsketch," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 31, 2018, pp. 10096–10106.
- [75] L. Ma and E. Solomonik, "Fast and accurate randomized algorithms for low-rank tensor decompositions," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 34, 2021, pp. 24299–24312.
- [76] R. Minster, A. K. Saibaba, and M. E. Kilmer, "Randomized algorithms for low-rank tensor decompositions in the tucker format," *SIAM J. Math. Data Sci.*, vol. 2, no. 1, pp. 189–215, Jan. 2020.
- [77] M. Che, Y. Wei, and H. Yan, "Randomized algorithms for the low multilinear rank approximations of tensors," *J. Comput. Appl. Math.*, vol. 390, Jul. 2021, Art. no. 113380.
- [78] M. Che, Y. Wei, and Y. Xu, "Randomized algorithms for the computation of multilinear rank- $(\mu_1, \mu_2, \mu_3)$  approximations," *J. Global Optim.*, vol. 87, nos. 2–4, pp. 373–403, Nov. 2023.
- [79] L. Yuan, C. Li, J. Cao, and Q. Zhao, "Randomized tensor ring decomposition and its application to large-scale data reconstruction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2127–2131.
- [80] S. Ahmadi-Asl et al., "Randomized algorithms for fast computation of low rank tensor ring model," *Mach. Learn., Sci. Technol.*, vol. 2, no. 1, Dec. 2020, Art. no. 011001.
- [81] O. A. Malik and S. Becker, "A sampling-based method for tensor ring decomposition," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 7400–7411.
- [82] Y. Yu and H. Li, "Practical sketching-based randomized tensor ring decomposition," *Numer. Linear Algebra Appl.*, vol. 2024, p. e2548, Jan. 2024, doi: [10.1002/nla.2548](https://doi.org/10.1002/nla.2548).
- [83] M. Che and Y. Wei, "Randomized algorithms for the approximations of tucker and the tensor train decompositions," *Adv. Comput. Math.*, vol. 45, no. 1, pp. 395–428, Feb. 2019.
- [84] Z. Chen, H. Jiang, G. Yu, and L. Qi, "Low-rank tensor train decomposition using TensorSketch," 2023, *arXiv:2309.08093*.
- [85] T. Shi, M. Ruth, and A. Townsend, "Parallel algorithms for computing the tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 45, no. 3, pp. C101–C130, Jun. 2023.
- [86] H. A. Daas, G. Ballard, and P. Benner, "Parallel algorithms for tensor train arithmetic," *SIAM J. Sci. Comput.*, vol. 44, no. 1, pp. C25–C53, Feb. 2022.
- [87] H. Al Daas et al., "Randomized algorithms for rounding in the tensor-train format," *SIAM J. Sci. Comput.*, vol. 45, no. 1, pp. A74–A95, Feb. 2023.
- [88] J. Zhang, A. K. Saibaba, M. E. Kilmer, and S. Aeron, "A randomized tensor singular value decomposition based on the t-product," *Numer. Linear Algebra Appl.*, vol. 25, no. 5, p. e2179, Oct. 2018.
- [89] D. A. Tarzanagh and G. Michailidis, "Fast randomized algorithms for t-product based tensor operations and decompositions with applications to imaging data," *SIAM J. Imag. Sci.*, vol. 11, no. 4, pp. 2629–2664, Jan. 2018.
- [90] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, "A new truncation strategy for the higher-order singular value decomposition," *SIAM J. Sci. Comput.*, vol. 34, no. 2, pp. A1027–A1052, Jan. 2012.
- [91] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its Oracle properties," *J. Amer. Stat. Assoc.*, vol. 96, no. 456, pp. 1348–1360, 2001.
- [92] P.-G. Martinsson and S. Voronin, "A randomized blocked algorithm for efficiently computing rank-revealing factorizations of matrices," *SIAM J. Sci. Comput.*, vol. 38, no. 5, pp. S485–S507, Jan. 2016.
- [93] W. Yu, Y. Gu, J. Li, S. Liu, and Y. Li, "Single-pass PCA of large high-dimensional data," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3350–3356.
- [94] E. J. Candes and M. B. Wakin, "Enhancing sparsity by reweighted  $\ell_1$  minimization," *J. Fourier Anal. Appl.*, vol. 14, pp. 877–905, Apr. 2008.
- [95] Y.-B. Zhao and D. Li, "Reweighted  $\ell_1$ -minimization for sparse solutions to underdetermined linear systems," *SIAM J. Optim.*, vol. 22, no. 3, pp. 1065–1088, Jan. 2012.
- [96] M. S. Asif and J. Romberg, "Fast and accurate algorithms for reweighted  $\ell_1$ -norm minimization," *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5905–5916, Dec. 2013.
- [97] W. Zuo, D. Meng, L. Zhang, X. Feng, and D. Zhang, "A generalized iterated shrinkage algorithm for non-convex sparse coding," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 217–224.
- [98] S. Boyd et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [99] R. Murray et al., "Randomized numerical linear algebra: A perspective on the field with an eye to software," 2023, *arXiv:2302.11474*.