

Robust Tensor SVD and Recovery With Rank Estimation

Qiquan Shi, Yiu-Ming Cheung¹, *Fellow, IEEE*, and Jian Lou²

Abstract—Tensor singular value decomposition (t-SVD) has recently become increasingly popular for tensor recovery under partial and/or corrupted observations. However, the existing t-SVD-based methods neither make use of a rank prior nor provide an accurate rank estimation (RE), which would limit their recovery performance. From the practical perspective, the tensor RE problem is nontrivial and difficult to solve. In this article, we, therefore, aim to *determine the correct rank of an intrinsic low-rank tensor from corrupted observations based on t-SVD and further improve recovery results with the estimated rank*. Specifically, we first induce the equivalence of the tensor nuclear norm (TNN) of a tensor and its *f*-diagonal tensor. We then simultaneously minimize the reconstruction error and TNN of the *f*-diagonal tensor, leading to RE. Subsequently, we relax our model by removing the TNN regularizer to improve the recovery performance. Furthermore, we consider more general cases in the presence of missing data and/or gross corruptions by proposing robust tensor principal component analysis and robust tensor completion with RE. The robust methods can achieve successful recovery by refining the models with correct estimated ranks. Experimental results show that the proposed methods outperform the state-of-the-art methods with significant improvements.

Index Terms—Rank estimation (RE), robust tensor PCA (RTPCA), robust tensor recovery, tensor completion (TC), tensor singular value decomposition (t-SVD).

I. INTRODUCTION

TENSORIAL data, such as color images and videos, are ubiquitous and have received considerable attention in many applications [1]. Tensor decomposition is a powerful computational tool for tensor analysis with missing entries, Gaussian noise, outliers, gross corruptions (non-Gaussian

noise), etc. It has been successfully applied to various fields, such as machine learning, data mining, and computer vision, for face recognition [2]; image inpainting [3], [4]; video background modeling [5], [6]; and hyperspectral image restoration [7]. In general, there are two common and fundamental decomposition models: 1) CANDECOMP/PARAFAC (CP) [8], [9] and 2) Tucker decomposition [10], which naturally produces two definitions of the tensor rank, that is: 1) CP-rank and 2) Tucker-rank, respectively.

In practice, some entries of tensors may often be missing in the acquisition process, such as costly experiments, etc. Missing data are common in real-world cases for many reasons. For example, in industrial applications, data, such as images, can be corrupted with irregular patterns due to the insufficient resolution of a device or malfunctioning equipment [11], [12]. Tensor completion (TC) techniques extended from matrix completion cases have been widely used for estimating missing data. Common TC approaches are based on CP and Tucker models. The CP-based TC methods can obtain good completion results under typical conditions if given a correct CP-rank. However, the CP-rank is generally NP-hard to compute [13], particularly with incomplete information. Although attempts have been made to determine the CP-rank using the Bayesian models [6], [14]–[16], these methods often underestimate or overestimate the truth, resulting in the deterioration of predictive performance [6]. While the convex relaxation of the CP rank is intractable, a convex surrogate for the Tucker rank, that is, a Tucker-based tensor nuclear norm (TNN) (sum of the nuclear norm of all matrices unfolded along each mode), has been proposed in [3] and has since appeared frequently in TC studies and has worked successfully [17]–[19]. For example, Liu *et al.* [3] proposed a high accuracy low-rank TC algorithm (HaLRTC) for estimating missing values in tensor visual data. However, these Tucker-based TC methods often require directly unfolding a tensor into matrices, which can destroy the intrinsic multidimensional structure of tensorial data, leading to vital information loss and degraded recovery performance [20].

In addition to missing data, outliers or non-Gaussian noise (e.g., sparse noise) can frequently occur in real-world data [6]. To address this problem, many robust CP-/Tucker-based approaches have been developed, such as robust tensor principal component analysis (PCA) methods [21], [22], which extend robust PCA (RPCA) to recover the low-rank and sparse components from corrupted observations. For tensors with both missing entries and gross corruptions, Goldfarb and Qin [23] combined the Tucker-based

Manuscript received 1 September 2020; revised 14 December 2020 and 27 February 2021; accepted 16 March 2021. Date of publication 19 April 2021; date of current version 16 September 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61672444; in part by the Hong Kong Baptist University (HKBU), Research Committee, Initiation Grant, Faculty Niche Research Areas (IGFNRA) 2018/19 under Grant RC-FNRA-IG/18-19/SCI/03; in part by the Innovation and Technology Fund of Innovation and Technology Commission of the Government of the Hong Kong SAR under Project ITS/339/18; in part by the project funded by HKBU Interdisciplinary Research Clusters Matching Scheme under Grant RC-IRCMs/18-19/SCI/01; and in part by the SZSTI under Grant JCYJ20160531194006833. This article was recommended by Associate Editor H. Qiao. (*Corresponding author: Yiu-Ming Cheung.*)

Qiquan Shi is with Huawei Noah's Ark Lab, Hong Kong, SAR, China (e-mail: shiqiquan@huawei.com).

Yiu-Ming Cheung and Jian Lou are with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, SAR, China (e-mail: ymc@comp.hkbu.edu.hk; jianlou@comp.hkbu.edu.hk).

This article has supplementary material provided by the authors and color versions of one or more figures available at <https://doi.org/10.1109/TCYB.2021.3067676>.

Digital Object Identifier 10.1109/TCYB.2021.3067676

nuclear-norm regularization with an ℓ_1 -norm loss function for robust TC. Later on, a Bayesian robust tensor factorization (BRTF) [6] employs a fully Bayesian generative model for recovering missing data and outliers or non-Gaussian noise with CP-rank estimation. Furthermore, Chen *et al.* [24] proposed a generalized weighted low-rank tensor factorization method (GWLRTF) integrated with noise modeling techniques for a mixture of Gaussians. However, the performance of these CP-based and Tucker-based robust tensor methods is also limited by the above-mentioned drawbacks of the CP and Tucker models.

A new tensor decomposition model, tensor singular value decomposition (t-SVD) [25]–[27], has recently become increasingly popular for solving TC problems [5], [28]–[30]. As t-SVD essentially treats third-order tensors as linear operators over matrices, the tensor multirank and tubal-rank defined based on t-SVD can well characterize the intrinsic low-rank structure of a tensor while avoiding the loss of information inherent in the unfolding of a tensor [20]. The TNN based on t-SVD has been proposed as a convex relaxation of tubal-rank [31]. Subsequently, Zhang *et al.* [28] used the TNN regularizer for incomplete and noisy videos, and achieved good inpainting and denoising results, and further theoretically analyzed the conditions for exact completion [29]. A twist TNN (t-TNN) [32] has also been developed for video completion. However, these TNN-based methods focus on rank minimization but do not utilize the low-rank prior, which would limit their recovery performance. A t-SVD-based factorization method (TCTF) [20] incorporates low-rank information with the factorization of a tensor as the product of two smaller-size tensors, which achieves better recovery results if given the true tubal rank. A heuristic rank-decreasing scheme is used to determine a rank for TCTF, but it often underestimates or overestimates the truth, resulting in degraded recovery accuracy.

TNN minimization has also been used in many methods for robust low-rank tensor recovery. For example, Lu *et al.* [30], [33] studied the tensor RPCA (TRPCA) problem by solving a convex objective of a weighted combination of the TNN of a tensor and the ℓ_1 -norm of the sparse error, and theoretically provided exact recovery for both the low rank and the sparse components under certain assumptions. They further studied the problem of robust low-rank tensor recovery from both partial and corrupted observations [30], [34]. Moreover, Zhou and Feng [35] proposed an outlier-robust tensor PCA (OR-TPCA) by combining the TNN minimization with the $\ell_{2,1}$ -norm, which achieves good results in outlier detection and unsupervised and semisupervised learning. However, these TNN-based robust tensor methods have the same drawbacks as TNN-based TC methods without utilizing the true tubal-rank information, which can cause performance degradation.

In this article, we solve the rank estimation (RE) problem and then make use of the correct estimated tubal-rank to improve the performance of recovering tensors with missing entries and/or gross corruptions. We propose a rank estimation method by simultaneously minimizing the reconstruction error and the TNN of the f -diagonal tensor of an incomplete tensor based on t-SVD, denoted as RE_{TNN} . In the RE_{TNN} model, we first induce the equivalence of TNN of a tensor

and its f -diagonal tensor. We then impose the TNN constraint on the f -diagonal tensor in the original domain and recast it as the ℓ_1 -norm of singular values in the Fourier domain, while estimating the missing entries. This results in RE with lower computational complexity than computing the TNN of the entire tensor. RE_{TNN} transforms the discrete RE to be tuning the continuous value of a regularization parameter, leading to good rank determination. However, RE_{TNN} focuses on RE only and the TNN regularization restricts RE_{TNN} to directly recover missing entries. To further improve the recovery performance, we propose a *relaxing strategy* for the RE_{TNN} model by only minimizing the reconstruction error *without* the TNN regularizer *after* RE. Therefore, a new t-SVD-based TC method is proposed and called tensor completion with RE (TC-RE). With the true tubal-rank R estimated by RE_{TNN} , TC-RE can achieve the optimal solution under mild conditions using rank- R t-SVD approximation, according to the multilinear generalization of the Eckart–Young–Mirsky theorem [25], [29]. To handle tensors with missing entries and/or gross corruptions (e.g., sparse noise), we integrate the proposed RE_{TNN} with ℓ_1 -norm regularization of the error component (corruptions) and further propose robust t-SVD models with RE, that is, robust tensor PCA with RE (RTPCA-RE) and robust TC with RE (RTC-RE) for low-rank tensor recovery under partial and/or grossly corrupted observations. RTPCA-RE and RTC-RE inherit the ability of RE_{TNN} to determine correct tubal-ranks and further improve their recovery performance via refinement schemes by explicitly making use of the estimated ranks to refine the models. Furthermore, we analyze and discuss the generalization of the proposed methods. We employ the alternating minimization method and the alternating direction method of multipliers (ADMM) [36] to solve our models. We evaluate the proposed methods on synthetic and real-world tensors with missing entries and/or sparse noise in different applications, for example, image/video inpainting and denoising, and video background modeling. The experimental results show our methods can achieve significant improvements over the state-of-the-art methods in terms of RE, TC, and robust tensor recovery.

In a nutshell, the main contributions of this article are four-fold.

- 1) We propose RE_{TNN} to determine the correct tubal rank of an incomplete tensor, where the RE_{TNN} model is formulated in the original domain and solved equivalently in the Fourier domain efficiently.
- 2) We propose TC-RE to further improve the recovery performance by developing a relax strategy. Given the correct estimated rank and sufficient observed entries, TC-RE can achieve an optimal completion solution.
- 3) We propose RTPCA-RE and RTC-RE to solve the robust tensor PCA and completion problems, respectively. These methods can not only estimate the correct tubal rank but also achieve successful recovery in the presence of missing data and/or gross corruption under the refinement schemes.
- 4) We analyze the generalization of the proposed methods by briefly discussing variants that can solve (robust) tensor learning problems.

The remainder of this article is presented as follows. We review preliminaries and related works in Section II. The proposed TC and robust tensor recovery with RE methods are derived in detail in Sections III and VI, respectively. The time complexity analysis is provided and the generalization of the proposed methods is discussed in Section V. Experimental results are reported in Section VI. Section VII concludes this article.

II. PRELIMINARIES AND RELATED WORKS

A. Notation

The number of dimensions of a tensor is the *order* and each dimension is a *mode* of it. A vector (first-order tensor) is denoted by a bold lowercase letter $\mathbf{x} \in \mathbb{R}^I$. A matrix (second-order tensor) is denoted by a bold capital letter $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$. A higher-order ($N \geq 3$) tensor is denoted by a calligraphic letter $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$. The i th entry of a vector $\mathbf{x} \in \mathbb{R}^I$ is denoted by x_i , and the (i, j) th entry of a matrix $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ is denoted by $X_{i,j}$. The (i_1, \dots, i_N) th entry of an N th-order tensor \mathcal{X} is denoted by $\mathcal{X}_{i_1, \dots, i_N}$. The Frobenius norm of a tensor \mathcal{X} is defined by $\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$, where $\langle \mathcal{X}, \mathcal{X} \rangle = \sum_{i_1} \sum_{i_2} \dots \sum_{i_N} \mathcal{X}(i_1, \dots, i_N)^2$ denotes inner product [1]. $\Omega \in \mathbb{R}^{I_1 \times \dots \times I_N}$ is a binary index set: $\Omega(i_1, \dots, i_N) = 1$ if $\mathcal{X}(i_1, \dots, i_N)$ is observed, and $\Omega(i_1, \dots, i_N) = 0$ otherwise. \mathcal{P}_Ω is the associated sampling operator that acquires only the entries indexed by Ω

$$\begin{aligned} (\mathcal{P}_\Omega(\mathcal{X}))(i_1, \dots, i_N) \\ = \begin{cases} \mathcal{X}(i_1, \dots, i_N), & \text{if } (i_1, \dots, i_N) \in \Omega \\ 0, & \text{if } (i_1, \dots, i_N) \in \Omega^c \end{cases} \end{aligned} \quad (1)$$

where Ω^c is the complement of Ω . Furthermore, for a third-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, its MATLAB notation $\mathcal{X}(:, :, i)$ or $\mathcal{X}^{(i)}$ denotes the i th *frontal slice* of \mathcal{X} . Besides, $\widehat{\mathcal{X}} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ refers to the Discrete Fourier transform (DFT) of \mathcal{X} along the 3-D, that is, $\widehat{\mathcal{X}} = \text{fft}(\mathcal{X}, [], 3)$ in MATLAB, and the inverse DFT (ifft) is computed via $\mathcal{X} = \text{ifft}(\widehat{\mathcal{X}}, [], 3)$.

B. Definitions Related to T-SVD

There are operations related to t-SVD defined as follows.

Definition 1 (Block-Diagonal Matrix) [25]: Let $\widehat{\mathbf{X}}$ denote the block-diagonal matrix of $\widehat{\mathcal{X}}$ in the Fourier domain, that is, $\widehat{\mathbf{X}} = \text{blockdiag}(\widehat{\mathcal{X}}) = \text{diag}(\widehat{\mathbf{X}}^{(1)}, \dots, \widehat{\mathbf{X}}^{(I_3)}) \in \mathbb{C}^{I_1 I_3 \times I_2 I_3}$, where $\text{diag}()$ is the diagonal operator.

Definition 2 (t-Product) [25]: The t -product $\mathcal{Z} = \mathcal{X} * \mathcal{Y}$ of $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and $\mathcal{Y} \in \mathbb{R}^{I_2 \times I_4 \times I_3}$ is a tensor of size $\mathcal{Z} \in \mathbb{R}^{I_1 \times I_4 \times I_3}$, where the (i, j) th tube denoted by $\mathcal{Z}(i, j, :)$ for $i = 1, 2, \dots, I_1$ and $j = 1, 2, \dots, I_4$ of the tensor \mathcal{Z} is given by $\sum_{k=1}^{I_2} \mathcal{X}(i, k, :) * \mathcal{Y}(k, j, :)$. $*$ denotes the t -product symbol.

Definition 3 (Conjugate Transpose) [25]: Let \mathcal{X} be a tensor of size $I_1 \times I_2 \times I_3$, then \mathcal{X}^\top is the $I_2 \times I_1 \times I_3$ tensor obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through I_3 . The symbol $^\top$ refers to the conjugate transpose.

Definition 4 (Identity Tensor and Orthogonal Tensor) [25]: A tensor $\mathcal{I} \in \mathbb{R}^{I_1 \times I_1 \times I_3}$ is identity if its first frontal slice is

Algorithm 1 t-SVD of a Third-Order Tensor [28]

Input: $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$
 $\widehat{\mathcal{X}} = \text{fft}(\mathcal{X}, [], 3)$;
for $i = 1$ to I_3 **do**
 $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \text{SVD}(\widehat{\mathcal{X}}^{(i)}); \widehat{\mathbf{U}}^{(i)} = \mathbf{U}; \widehat{\mathbf{S}}^{(i)} = \mathbf{S}; \widehat{\mathbf{V}}^{(i)} = \mathbf{V}$;
end for
 $\mathcal{U} = \text{ifft}(\widehat{\mathbf{U}}, [], 3); \mathcal{S} = \text{ifft}(\widehat{\mathbf{S}}, [], 3); \mathcal{V} = \text{ifft}(\widehat{\mathbf{V}}, [], 3)$;
Output: $\mathcal{U} \in \mathbb{R}^{I_1 \times I_1 \times I_3}, \mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times I_3}, \mathcal{V} \in \mathbb{R}^{I_2 \times I_2 \times I_3}$.

the $I_1 \times I_1$ identity matrix and all other frontal slices are 0. A tensor \mathcal{Q} is orthogonal if $\mathcal{Q}^\top * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^\top = \mathcal{I}$.

Definition 5 (f-Diagonal Tensor) [25]: A tensor \mathcal{X} is called f -diagonal if each frontal slice of \mathcal{X} is a diagonal matrix.

Definition 6 (Tensor Multirank and Tubal-rank) [20], [25]: The *multirank* of $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is a vector $\mathbf{r} \in \mathbb{R}^{I_3}$, whose i th entry is the rank of the i th frontal slice of $\widehat{\mathcal{X}}$, that is, $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_{I_3}) = (\text{rank}(\widehat{\mathbf{X}}^{(1)}), \dots, \text{rank}(\widehat{\mathbf{X}}^{(I_3)}))$, where $\text{rank}()$ denotes the rank of a matrix. The *tubal-rank* R of \mathcal{X} is the largest rank of all the frontal slices of $\widehat{\mathcal{X}}$ in the Fourier domain, that is, $R = \max(\mathbf{r}_1, \dots, \mathbf{r}_{I_3})$.

Definition 7 (t-SVD) [25]: The t-SVD of $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is given by

$$\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top \quad (2)$$

where $\mathcal{U} \in \mathbb{R}^{I_1 \times I_1 \times I_3}$ and $\mathcal{V} \in \mathbb{R}^{I_2 \times I_2 \times I_3}$ are orthogonal tensors. $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is an f -diagonal tensor.

The t-SVD of a tensor is obtained by computing the matrix SVDs in the Fourier domain (see Algorithm 1). Besides, it is usually sufficient to compute the truncated t-SVD with a tubal-rank R . For $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ with the tubal-rank R ($R < \min(I_1, I_2)$), the truncated t-SVD is given by $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$, where $\mathcal{U} \in \mathbb{R}^{I_1 \times R \times I_3}$, $\mathcal{V} \in \mathbb{R}^{I_2 \times R \times I_3}$, and $\mathcal{S} \in \mathbb{R}^{R \times R \times I_3}$. This truncated version of the t-SVD will be used throughout the article.

Definition 8 (TNN) [33]: The TNN of a tensor \mathcal{X} is denoted by $\|\mathcal{X}\|_{\text{TNN}}$ and defined as the average of the nuclear norm of all the frontal slices of $\widehat{\mathcal{X}}$, that is, $\|\mathcal{X}\|_{\text{TNN}} = (1/I_3) \sum_{i=1}^{I_3} \|\widehat{\mathbf{X}}^{(i)}\|_*$.

Theorem 1 [25], [29]: Assume the t-SVD of $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is given by $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$. For $p \leq R < \min(I_1, I_2)$, we define $\mathcal{X}_p = \sum_{i=1}^p \mathcal{U}(:, i, :) * \mathcal{S}(:, i, :) * \mathcal{V}(:, i, :)$, then

$$\mathcal{X}_p = \arg \min_{\mathcal{X} \in \mathbb{X}} \|\mathcal{X} - \widehat{\mathcal{X}}\|_F^2 \quad (3)$$

where $\mathbb{X} = \{\mathcal{C} = \mathcal{A} * \mathcal{B} | \mathcal{A} \in \mathbb{R}^{I_1 \times R \times I_3}, \mathcal{B} \in \mathbb{R}^{I_2 \times R \times I_3}\}$ is the set of tensors with tubal-rank p ($p \leq R$). Thus, the unique optimal solution of low tubal-rank p approximation of \mathcal{X} is given by the truncated rank- p approximation (truncated t-SVD) of \mathcal{X} , that is, \mathcal{X}_p . This theorem can be viewed as the multilinear generalization of the Eckart–Young–Mirsky theorem [37], [38].

C. Related Works

1) *T-SVD for Tensor Completion*: Based on t-SVD, Zhang *et al.* [28] and Zhang and Aeron [29] solved the TC problems by minimizing the TNN of the incomplete tensor, that is

$$\min_{\mathcal{X}} \|\mathcal{X}\|_{\text{TNN}} \quad \text{s.t.} \quad \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}) \quad (4)$$

where \mathcal{X} is the target tensor to be recovered given partially observed entries from \mathcal{T} . Zhang and Aeron further theoretically analyze the conditions for exact completion in [29]. The TNN-based model (4) aims to minimize the tubal-rank using the TNN relaxation without utilizing the information of the rank, which probably limits its completion performance. Most recently, another t -SVD-based TC method (TCTF) [20] has incorporated the low tubal-rank information with factorization of \mathcal{X} (as the product of two tensors $\mathcal{Y} \in \mathbb{R}^{I_1 \times R \times I_3}$ and $\mathcal{Z} \in \mathbb{R}^{I_2 \times R \times I_3}$ with smaller sizes)

$$\min_{\mathcal{X}, \mathcal{Y}, \mathcal{Z}} \|\mathcal{X} - \mathcal{Y} * \mathcal{Z}\|_F^2 \quad \text{s.t. } \mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{T}) \quad (5)$$

which can achieve better recovery efficiently without computing the TNN of entire \mathcal{X} in (4). TCTF requires a tensor rank to be determined *a priori*. To the best of our knowledge, TCTF is the only existing t -SVD-based method that considers RE, but its heuristic rank-decreasing scheme often under- or over-estimates the truth, leading to degraded recovery performance.

2) *T-SVD for Robust Tensor PCA*: Zhou and Feng [35] proposed OR-TPCA by combining TNN minimization with $\ell_{2,1}$ norm regularization

$$\min_{\mathcal{X}, \mathcal{E}} \|\mathcal{X}\|_{\text{TNN}} + \beta \|\mathcal{E}\|_{2,1} \quad \text{s.t. } \mathcal{T} = \mathcal{X} + \mathcal{E} \quad (6)$$

where $\|\mathcal{E}\|_{2,1}$ is the $\ell_{2,1}$ norm to characterize the sparsity of outliers \mathcal{E} . OR-TPCA achieves good results on outlier detection and unsupervised/semisupervised learning. However, it is not capable of general tasks in removing non-Gaussian noise from a single tensor (e.g., an image), based on our preliminary studies. Furthermore, Lu *et al.* [30] considered the robust tensor PCA problem by solving a convex TRPCA objective, that is, a weighted combination of the TNN of a low-rank tensor and the ℓ_1 -norm of the error

$$\min_{\mathcal{X}, \mathcal{E}} \|\mathcal{X}\|_{\text{TNN}} + \beta \|\mathcal{E}\|_1 \quad \text{s.t. } \mathcal{T} = \mathcal{X} + \mathcal{E} \quad (7)$$

where $\beta = 1/\sqrt{(\max(I_1, I_2)I_3)}$. TRPCA can achieve exact recovery for both the low-rank and the sparse components under suitable assumptions [30].

3) *T-SVD for Robust Tensor Completion*: Few t -SVD-based methods are available for recovering tensors with both missing entries and sparse noise. Based on TNN regularization, Lu *et al.* [34] solved the noisy low-rank TC by reformulating the TRPCA model (7) as follows:

$$\min_{\mathcal{X}, \mathcal{E}} \|\mathcal{X}\|_{\text{TNN}} + \beta \|\mathcal{E}\|_1 \quad \text{s.t. } \mathcal{P}_{\Omega}(\mathcal{T}) = \mathcal{P}_{\Omega}(\mathcal{X} + \mathcal{E}). \quad (8)$$

The code of this algorithm (call LRTCR-TNN) has been published in the LibADMM Toolbox¹ [34]. To the best of our knowledge, this is the only t -SVD-based robust TC method.

Remark 1: Rank information is very important for matrix/tensor recovery. Many studies in the literature have formulated the low-rank matrix/TC task as the problem of estimating the rank of a certain matrix/tensor [3], [17], [19], [28], [39]–[44]. However, like other similar methods (e.g., TNN,

TCTF, TRPCA, and OR-TPCA), LRTCR-TNN aims to minimize the rank of observations and does not explicitly make use of the true rank prior, which would limit their performance. On the other hand, some factorization methods, such as [45]–[47] require a large enough initial rank to ensure good recovery performance, usually leading to much higher computational cost than ours when dealing with large-scale data. To alleviate this situation, we introduce an efficient RE technique, which equivalently transforms the estimating of discrete rank to be tuning the wide range of continuous values of a hyperparameter. Based on the flexible RE, we further fully utilize the rank information to further improve the recovery performance as presented in the following section.

III. PROPOSED METHODS

In this section, we propose two t -SVD-based methods: 1) one aims to solve the RE problem for incomplete tensors and 2) the other aims to improve TC performance.

A. Rank Estimation via RE_{TNN}

To improve the recovery performance by utilizing a rank prior, we must first correctly determine the rank. The existing TNN-based methods aim to minimize the rank only and become very slow for large-scale tensors, due to the heavy costs of computing TNN of the entire tensor. Motivated by these, we first induce a lemma.

Lemma 1: Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ with the tubal-rank R , and $\mathcal{S} \in \mathbb{R}^{R \times R \times I_3}$ is the f -diagonal tensor of \mathcal{X} with orthogonal tensor factors \mathcal{U} and \mathcal{V} , that is, $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\top}$, then

$$\|\mathcal{X}\|_{\text{TNN}} = \|\mathcal{S}\|_{\text{TNN}} \quad (9)$$

where $\|\mathcal{X}\|_{\text{TNN}}$ is the TNN of \mathcal{X} .

See the proof of this lemma in Appendix A of the Supplementary Material.² With Lemma 1 and based on t -SVD, we impose the TNN regularizer on the f -diagonal tensor while minimizing the reconstruction error, to determine the rank of an incomplete tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ (with a low tubal rank)

$$\min_{\mathcal{X}, \mathcal{U}, \mathcal{S}, \mathcal{V}} \frac{1}{2} \|\mathcal{X} - \mathcal{U} * \mathcal{S} * \mathcal{V}^{\top}\|_F^2 + \lambda \|\mathcal{S}\|_{\text{TNN}} \quad \text{s.t. } \mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{T}), \mathcal{U}^{\top} * \mathcal{U} = \mathcal{I}, \mathcal{V}^{\top} * \mathcal{V} = \mathcal{I} \quad (10)$$

where λ is the penalty parameter and \mathcal{S} is the f -diagonal tensor of the target (recovery) tensor \mathcal{X} . To solve the model (10) simply, we further develop the following theorem.

Theorem 2: The objective function (10) is equivalent to

$$\min_{\widehat{\mathbf{X}}^{(i)}, \widehat{\mathbf{U}}^{(i)}, \widehat{\mathbf{S}}^{(i)}, \widehat{\mathbf{V}}^{(i)}} \frac{1}{I_3} \sum_{i=1}^{I_3} \left(\frac{1}{2} \left\| \widehat{\mathbf{X}}^{(i)} - \widehat{\mathbf{U}}^{(i)} \widehat{\mathbf{S}}^{(i)} \widehat{\mathbf{V}}^{(i)\top} \right\|_F^2 + \lambda \left\| \text{diag}(\widehat{\mathbf{S}}^{(i)}) \right\|_1 \right) \quad \text{s.t. } \mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{T}), \widehat{\mathbf{U}}^{(i)\top} \widehat{\mathbf{U}}^{(i)} = \mathbf{I} \quad \widehat{\mathbf{V}}^{(i)\top} \widehat{\mathbf{V}}^{(i)} = \mathbf{I}, i = 1 \cdots I_3 \quad (11)$$

²Available at: <https://www.dropbox.com/sh/c8gbzgfvhq10k42/AADPolrrrKVUInJp8U5ytBSoa?dl=0>.

¹LibADMM: <https://github.com/canyilu/LibADMM>.

where $\text{diag}(\widehat{\mathbf{S}}^{(i)}) = \widehat{\mathbf{s}}^{(i)} = \{\widehat{s}_r^{(i)}\}_{r=1}^{\mathbf{r}_i}$ are the singular values of $\widehat{\mathbf{X}}^{(i)}$ (the i th frontal slice of $\widehat{\mathcal{X}}$). $\text{diag}()$ is the diagonal operator. $\{\mathbf{r}_i\}_{i=1}^{I_3}$ are the tensor multiranks. \mathbf{I} is an identity matrix.

Proof: According to the properties of the t -product [25], we know that $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ and $\widehat{\mathbf{X}} = \widehat{\mathbf{U}}\widehat{\mathbf{S}}\widehat{\mathbf{V}}^\top$ are equivalent and

$$\begin{aligned} \|\mathcal{X} - \mathcal{U} * \mathcal{S} * \mathcal{V}^\top\|_F^2 &= \frac{1}{I_3} \|\widehat{\mathbf{X}} - \widehat{\mathbf{U}}\widehat{\mathbf{S}}\widehat{\mathbf{V}}^\top\|_F^2 \\ &= \frac{1}{I_3} \sum_{i=1}^{I_3} \|\widehat{\mathbf{X}}^{(i)} - \widehat{\mathbf{U}}^{(i)}\widehat{\mathbf{S}}^{(i)}\widehat{\mathbf{V}}^{(i)\top}\|_F^2 \end{aligned} \quad (12)$$

in which $\widehat{\mathbf{X}}^{(i)} = \widehat{\mathbf{U}}^{(i)}\widehat{\mathbf{S}}^{(i)}\widehat{\mathbf{V}}^{(i)\top} = \sum_{r=1}^{\mathbf{r}_i} \widehat{s}_r^{(i)} \widehat{\mathbf{u}}_r^{(i)} \widehat{\mathbf{v}}_r^{(i)\top}$ is the SVD of $\widehat{\mathbf{X}}^{(i)}$, and $\{\widehat{\mathbf{U}}^{(i)}, \widehat{\mathbf{S}}^{(i)}, \widehat{\mathbf{V}}^{(i)}\}$ are the i th frontal slice of $\{\widehat{\mathbf{U}}, \widehat{\mathbf{S}}, \widehat{\mathbf{V}}\}$, respectively. As $\widehat{\mathbf{S}}$ is an f -diagonal tensor, $\|\widehat{\mathbf{S}}^{(i)}\|_* = \|\widehat{\mathbf{s}}^{(i)}\|_1$, where $\widehat{\mathbf{s}}^{(i)} = \text{diag}(\widehat{\mathbf{S}}^{(i)})$. With the TNN of \mathcal{S} is $\|\mathcal{S}\|_{\text{TNN}} = (1/I_3) \sum_{i=1}^{I_3} \|\widehat{\mathbf{S}}^{(i)}\|_*$, we have

$$\|\mathcal{S}\|_{\text{TNN}} = \frac{1}{I_3} \sum_{i=1}^{I_3} \|\text{diag}(\widehat{\mathbf{S}}^{(i)})\|_1. \quad (13)$$

Hence, (10) is equivalent to (11). \blacksquare

With Theorem 2, we recast model (10) in the original space to the equivalent model (11) in the Fourier space, which is easier to solve. By reformulating $\|\mathcal{S}\|_{\text{TNN}}$ as the ℓ_1 -norm of $\{\widehat{\mathbf{s}}^{(i)}\}_{i=1}^{I_3}$ (singular values in the Fourier domain), we reduce the computational complexity further than model (10), and the TNN model (4). We name this rank estimation via t -SVD with TNN regularization, as RE_{TNN} .

Remark 2: The ℓ_1 -norm regularization on $\text{diag}(\widehat{\mathbf{S}}^{(i)})_{i=1}^{I_3}$ in (11) [equivalent to the TNN of f -diagonal tensor \mathcal{S} in (10)] makes the vectors of singular values sparse while the zero entries of $\text{diag}(\widehat{\mathbf{S}}^{(i)})_{i=1}^{I_3}$ are finally removed, leading to low tensor multirank and tubal-RE. Such a group sparsity technique [48]–[50] transforms the discrete RE to be tuning the continuous value of a regularization parameter, leading to good rank determination. Specifically, the discrete nature of the rank makes it is difficult to tune, which often leads to significant performance degradation even when the rank is just slightly changed to its neighboring values. Instead, our equivalent form (11) proposes to tune the continuous value of λ , which gives a good empirical performance with a wide range of values, as supported by our extensive experimental results. For example, the results presented in Appendix D of the supplementary material demonstrate that: a wide range of values of parameter λ and initial rank \widehat{R} lead to good tensor RE and recovery performance in most cases, excepting for the cases of too many missing entries. In other words, the RE and completion performance of proposed methods are stable and not sensitive to the values of parameters in general.

1) *Optimization of RE_{TNN} via Alternating Minimization³:* We use a simple alternating minimization method for the optimization of RE_{TNN} . As we need initial values for $\{\mathbf{r}_i\}_{i=1}^{I_3}$ to optimize model (11), we denote \widehat{R} as the initialization of tubal-rank for each entry of multirank; thus, $\widehat{R} = \mathbf{r}_1 = \dots = \mathbf{r}_{I_3}$. We divide the target variables into $I_3 \times 4$ groups:

Algorithm 2 RE via t -SVD With TNN Regularization (RE_{TNN})

- 1: **Input:** $\mathcal{P}_\Omega(\mathcal{T})$, Ω , λ , initial rank \widehat{R} , K , and tol .
- 2: **Initialization:** Set $\mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T})$, $\mathcal{P}_{\Omega^c}(\mathcal{X}) = \mathbf{0}$.
- 3: **for** $k = 1, \dots, K$ **do**
- 4: $\widehat{\mathcal{X}} = \text{fft}(\mathcal{X}, [], 3)$;
- 5: **for** $i = 1, \dots, I_3$ **do**
- 6: Update $\widehat{\mathbf{U}}^{(i)}$ and $\widehat{\mathbf{V}}^{(i)}$ by (16), and $\widehat{\mathbf{S}}^{(i)}$ by (19)
- 7: Tensor multi-rank estimation: **Only keep** the elements in $\text{diag}(\widehat{\mathbf{S}}^{(i)})$ where $\text{diag}(\widehat{\mathbf{S}}^{(i)}) > 0$, and then compute i th entry of multi-rank $\mathbf{r}_i = \text{length}(\text{diag}(\widehat{\mathbf{S}}^{(i)}))$ and update $\widehat{R} = \mathbf{r}_i$.
- 8: Update $\widehat{\mathbf{X}}^{(i)} = \widehat{\mathbf{U}}^{(i)}\widehat{\mathbf{S}}^{(i)}\widehat{\mathbf{V}}^{(i)\top}$.
- 9: **end for**
- 10: **Update** \mathcal{X} : $\mathcal{X} = \text{ifft}(\widehat{\mathcal{X}}, [], 3)$ and $\mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T})$.
- 11: If $\frac{\|\mathcal{X}^{(k+1)} - \mathcal{X}^k\|_F}{\|\mathcal{X}^{(k+1)}\|_F} < \text{tol}$, break; otherwise, continue.
- 12: **end for**
- 13: **Tensor tubal-rank estimation:** $R = \max\{\mathbf{r}_1, \dots, \mathbf{r}_{I_3}\}$
- 14: **Output:** Tensor tubal-rank R , estimated tensor \mathcal{X} .

$\{\widehat{\mathbf{U}}^{(i)}, \widehat{\mathbf{S}}^{(i)}, \widehat{\mathbf{V}}^{(i)}, \widehat{\mathbf{X}}^{(i)}\}_{i=1}^{I_3}$. These groups could be optimized in parallel.

The objective function (11) with respect to the i th block $\{\widehat{\mathbf{S}}^{(i)}, \widehat{\mathbf{U}}^{(i)}, \widehat{\mathbf{V}}^{(i)}, \widehat{\mathbf{X}}^{(i)}\}$ is

$$\begin{aligned} \min_{\widehat{\mathbf{X}}^{(i)}, \widehat{\mathbf{U}}^{(i)}, \widehat{\mathbf{S}}^{(i)}, \widehat{\mathbf{V}}^{(i)}} \quad & \frac{1}{2} \|\widehat{\mathbf{X}}^{(i)} - \widehat{\mathbf{U}}^{(i)}\widehat{\mathbf{S}}^{(i)}\widehat{\mathbf{V}}^{(i)\top}\|_F^2 + \lambda \|\text{diag}(\widehat{\mathbf{S}}^{(i)})\|_1 \\ \text{s.t.} \quad & \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}), \widehat{\mathbf{U}}^{(i)\top}\widehat{\mathbf{U}}^{(i)} = \mathbf{I} \\ & \widehat{\mathbf{V}}^{(i)\top}\widehat{\mathbf{V}}^{(i)} = \mathbf{I}, i = 1 \dots I_3. \end{aligned} \quad (14)$$

In this way, we can update each block while fixing other blocks alternatively.

Update $\widehat{\mathbf{U}}^{(i)}$ and $\widehat{\mathbf{V}}^{(i)}$: Equation (14) with respect to $\widehat{\mathbf{U}}^{(i)}$ and $\widehat{\mathbf{V}}^{(i)}$ is equivalent to

$$\begin{aligned} \min_{\widehat{\mathbf{U}}^{(i)}, \widehat{\mathbf{V}}^{(i)}} \quad & \frac{1}{2} \|\widehat{\mathbf{X}}^{(i)} - \widehat{\mathbf{U}}^{(i)}\widehat{\mathbf{S}}^{(i)}\widehat{\mathbf{V}}^{(i)\top}\|_F^2 \\ \text{s.t.} \quad & \widehat{\mathbf{U}}^{(i)\top}\widehat{\mathbf{U}}^{(i)} = \mathbf{I}, \widehat{\mathbf{V}}^{(i)\top}\widehat{\mathbf{V}}^{(i)} = \mathbf{I}, i = 1 \dots I_3 \end{aligned} \quad (15)$$

where $\widehat{\mathbf{U}}^{(i)} = [\widehat{\mathbf{u}}_r^{(1)}, \widehat{\mathbf{u}}_r^{(2)}, \dots, \widehat{\mathbf{u}}_r^{(\widehat{R})}]$, $\widehat{\mathbf{S}}^{(i)} = [\widehat{s}_r^{(1)}, \widehat{s}_r^{(2)}, \dots, \widehat{s}_r^{(\widehat{R})}]$, and $\widehat{\mathbf{V}}^{(i)} = [\widehat{\mathbf{v}}_r^{(1)}, \widehat{\mathbf{v}}_r^{(2)}, \dots, \widehat{\mathbf{v}}_r^{(\widehat{R})}]$. According to the Eckart–Young–Mirsky theorem, we have the closed-form solutions for $\widehat{\mathbf{U}}^{(i)}$ and $\widehat{\mathbf{V}}^{(i)}$ simultaneously using rank- \widehat{R} truncated SVD, that is

$$\begin{aligned} [\widehat{\mathbf{U}}^{(i)0}, \widehat{\mathbf{S}}^{(i)0}, \widehat{\mathbf{V}}^{(i)0}] &= \text{SVD}(\widehat{\mathbf{X}}^{(i)}) \Rightarrow \\ \widehat{\mathbf{U}}^{(i)} &= \widehat{\mathbf{U}}_{\mathbf{r}_i}^{(i)} = \widehat{\mathbf{U}}^{(i)0}(:, 1 : \widehat{R}) \\ \widehat{\mathbf{V}}^{(i)} &= \widehat{\mathbf{V}}_{\mathbf{r}_i}^{(i)} = \widehat{\mathbf{V}}^{(i)0}(:, 1 : \widehat{R}) \end{aligned} \quad (16)$$

where $\mathbf{r}_i = \widehat{R}$ is the initial multirank of the i th front slice of $\widehat{\mathcal{X}}$ (i.e., $\widehat{\mathbf{X}}^{(i)}$). In this way, we obtain the solutions for $\widehat{\mathbf{U}}^{(i)}$ and $\widehat{\mathbf{V}}^{(i)}$ simultaneously in one iteration.⁴

⁴We also provide another alternative way to solve $\{\widehat{\mathbf{u}}_r^{(i)}\}_{i=1}^{\widehat{R}}$ and $\{\widehat{\mathbf{v}}_r^{(i)}\}_{i=1}^{\widehat{R}}$. See Appendix B of the Supplementary Material.

³For simplicity, we omit the iteration variable k in this article.

Algorithm 3 TC With RE

1: **Input:** $\mathcal{P}_\Omega(\mathcal{T})$, Ω , λ , \hat{R} , K , and tol .
2: Obtain the estimated \mathcal{X} and tubal-rank R by **Algorithm 2**.
3: **for** $k = 1, \dots, K$ **do**
4: $\hat{\mathcal{X}} = \text{fft}(\mathcal{X}, [], 3)$
5: **for** $i = 1$ to I_3 **do**
6: $[\hat{\mathbf{U}}, \hat{\mathbf{S}}, \hat{\mathbf{V}}] = \text{SVD}(\hat{\mathcal{X}}^{(i)})$
7: $\hat{\mathcal{X}}_R^{(i)} = \hat{\mathbf{U}}(:, 1 : R) \hat{\mathbf{S}}(1 : R, 1 : R) \hat{\mathbf{V}}^T(:, 1 : R)$
8: **end for**
9: $\mathcal{X}_R = \text{ifft}(\hat{\mathcal{X}}_R, [], 3)$
10: $\mathcal{P}_\Omega(\mathcal{X}_R) = \mathcal{P}_\Omega(\mathcal{T})$
11: If $\frac{\|\mathcal{X}_R^{(k+1)} - \mathcal{X}_R^k\|_F}{\|\mathcal{X}_R^{(k+1)}\|_F} < tol$, break; otherwise, continue.
12: **end for**
13: **Output:** Recovered tensor \mathcal{X}_R , tensor tubal-rank R .

Update $\hat{\mathbf{S}}^{(i)}$: The function (14) with respect to $\hat{\mathbf{S}}^{(i)}$ is

$$\mathcal{L}_{\hat{\mathbf{S}}^{(i)}} = \frac{1}{2} \left\| \hat{\mathbf{X}}^{(i)} - \hat{\mathbf{U}}_{\mathbf{r}_i}^{(i)} \hat{\mathbf{S}}^{(i)} \hat{\mathbf{V}}_{\mathbf{r}_i}^{(i)\top} \right\|_F^2 + \lambda \left\| \text{diag}(\hat{\mathbf{S}}^{(i)}) \right\|_1. \quad (17)$$

Since $\{\hat{\mathbf{U}}_{\mathbf{r}_i}^{(i)}, \hat{\mathbf{V}}_{\mathbf{r}_i}^{(i)}\}$ are orthogonal, (17) is equivalent to

$$\mathcal{L}_{\hat{\mathbf{S}}^{(i)}} = \frac{1}{2} \left\| \hat{\mathbf{S}}^{(i)} - \hat{\mathbf{U}}_{\mathbf{r}_i}^{(i)\top} \hat{\mathbf{X}}^{(i)} \hat{\mathbf{V}}_{\mathbf{r}_i}^{(i)} \right\|_F^2 + \lambda \left\| \text{diag}(\hat{\mathbf{S}}^{(i)}) \right\|_1. \quad (18)$$

We here denote $t = \text{diag}(\hat{\mathbf{U}}_{\mathbf{r}_i}^{(i)\top} \hat{\mathbf{X}}^{(i)} \hat{\mathbf{V}}_{\mathbf{r}_i}^{(i)})$, which is essentially equivalent to $\text{diag}(\hat{\mathbf{S}}_{\mathbf{r}_i}^{(i)}) = \text{diag}(\hat{\mathbf{S}}^{(i)0}(:, 1 : \hat{R}))$. Based on the soft-thresholding algorithm [51] for ℓ_1 -norm regularization, $\hat{\mathbf{S}}^{(i)}$ is updated by

$$\text{diag}(\hat{\mathbf{S}}^{(i)}) = \text{prox}_\lambda(t) = \begin{cases} t - \lambda, & (t > \lambda) \\ 0, & (|t| \leq \lambda) \\ t + \lambda, & (t < -\lambda) \end{cases} \quad (19)$$

where $\hat{\mathbf{S}}^{(i)}$ is diagonal and prox is the soft-thresholding operator [51].

Tensor Multirank Estimation: After updating $\hat{\mathbf{S}}^{(i)}$, we check its values and only keep those nonzero ones. Thus, the number of the remaining singular values in $\text{diag}(\hat{\mathbf{S}}^{(i)})$ is the i th entry of tensor multirank \mathbf{r}_i , that is, multirank $\mathbf{r}_i = \text{length}(\text{diag}(\hat{\mathbf{S}}^{(i)}))$. Meanwhile, we update the initial rank $\hat{R} = \mathbf{r}_i$ for the update of the $(i + 1)$ th front slice.

Update $\hat{\mathbf{X}}^{(i)}$: Equation (14) with respect to $\hat{\mathbf{X}}^{(i)}$ is

$$\min_{\hat{\mathbf{X}}^{(i)}} \frac{1}{2} \left\| \hat{\mathbf{X}}^{(i)} - \hat{\mathbf{U}}^{(i)} \hat{\mathbf{S}}^{(i)} \hat{\mathbf{V}}^{(i)\top} \right\|_F^2. \quad (20)$$

By deriving the Karush–Kuhn–Tucker (KKT) conditions for (20), each $\hat{\mathbf{X}}^{(i)} = \hat{\mathbf{U}}^{(i)} \hat{\mathbf{S}}^{(i)} \hat{\mathbf{V}}^{(i)\top}$. After updating all the frontal slices of $\hat{\mathcal{X}}$, we can compute \mathcal{X} by: $\mathcal{X} = \text{ifft}(\hat{\mathcal{X}}, [], 3)$ and $\mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T})$.

Tensor Tubal-Rank Estimation: With updating all of the above variables for each $\hat{\mathbf{X}}^{(i)}$, we have estimated all $\{\mathbf{r}_i\}_{i=1}^{I_3}$ for all frontal slices and obtain the multirank and, finally, determine the tubal-rank $R = \max\{\mathbf{r}_1, \dots, \mathbf{r}_3\}$.

Finally, we summarize RE_{TNN} in Algorithm 2.

B. Tensor Completion via TC-RE

RE_{TNN} can accurately determine the rank and simultaneously estimate the missing entries. However, the TNN

regularization of model (10) restricts RE_{TNN} from directly minimizing the reconstruction error, leading to limited recovery results. To further improve recovery performance, we propose a *relaxing strategy*: we relax RE_{TNN} model (10) by *only* minimizing the reconstruction error *without* the TNN regularization *after* obtaining the tubal-rank information. In other words, we first estimate the tubal-rank R by RE_{TNN} and then we relax the model by removing the TNN regularizer and further improve the recovery accuracy. Thus, after RE by Algorithm 2, (10) is relaxed as

$$\begin{aligned} \min_{\mathcal{X}, \mathbf{U}, \mathbf{S}, \mathbf{V}} \quad & \left\| \mathcal{X} - \mathbf{U} * \mathbf{S} * \mathbf{V}^\top \right\|_F^2, \\ \text{s.t.} \quad & \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}), \mathbf{U}^\top * \mathbf{U} = \mathcal{I}, \mathbf{V}^\top * \mathbf{V} = \mathcal{I}. \end{aligned} \quad (21)$$

This relaxed model (21) can be solved iteratively in closed-form for each frontal slice of \mathcal{X} using the rank- R truncated t-SVD approximation.⁵ We name this new t-SVD-based TC method TC-RE and summarize it in Algorithm 3.

Remark 3: If \mathcal{T} (with the tubal-rank R) obeys the tensor incoherence conditions and observes enough randomly sampled entries, \mathcal{T} can be exactly recovered with theoretical guarantees [29]. Thus, for an incomplete tensor with enough observed entries, the missing entries can be exactly predicted under appropriate conditions. In the TC-RE model (21), with the true rank R estimated by RE_{TNN}, the missing entries are updated iteratively by computing the rank- R approximation of \mathcal{X} using the truncated t-SVD of \mathcal{X} given the observed entries from \mathcal{T} ($\mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T})$), and finally can be recovered exactly under the appropriate conditions. On the other hand, the unique optimal rank- R approximation of \mathcal{X} is given by the truncated t-SVD of \mathcal{X} according to the multilinear generalization of the Eckart–Young–Mirsky theorem [25], [29]. Hence, making use of the information of true rank R and with the relaxing strategy, TC-RE can achieve the optimal completion solution for low-rank datasets under the appropriate conditions.

IV. PROPOSED ROBUST TENSOR RECOVERY WITH RANK ESTIMATION

Considering tensors can also be corrupted by noise in addition to missing values in practice, we further propose two robust t-SVD methods to address the robust tensor PCA and robust TC problems, which can recover low-rank tensor based on incomplete and/or grossly corrupted observations and, meanwhile, provide RE.

A. Robust Tensor PCA With RE

In practice, real-world data are commonly corrupted by non-Gaussian noise (gross corruptions) due to sensor failures, malicious tampering, or other system errors [6]. To solve this problem, we further propose the RTPCA-RE. RTPCA-RE aims to recover a low-rank tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ from corrupted observations $\mathcal{T} = \mathcal{X} + \mathcal{E}$, where $\mathcal{E} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ represents

⁵The rank- R truncated t-SVD algorithm refers to Algorithm 1 in Appendix C of the Supplementary Material.

Algorithm 4 Robust Tensor PCA With RE (RTPCA-RE)

1: **Input:** $\mathcal{P}_{\Omega}(\mathcal{T})$, Ω , λ , initial rank \widehat{R} , K , and tol .
2: **Initialization:** Set $\mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{T})$, $\mathcal{P}_{\Omega^c}(\mathcal{X}) = \mathbf{0}$;
Set $\beta = 1/\sqrt{\max(I_1, I_2)I_3}$ and $\gamma = 1/\lambda$; Initialize
 $\{\{\widehat{\mathbf{u}}_r^{(i)}, \widehat{\mathbf{v}}_r^{(i)}, \mathbf{s}_r^{(i)}\}_{r=1}^{\widehat{R}}\}_{i=1}^{I_3}$ of $\widehat{\mathcal{X}}$ randomly.
3: **Step 1 – Rank Estimation (RE):**
4: **for** $k = 1, \dots, K$ **do**
5: $\widehat{\mathcal{X}} = \text{fft}(\mathcal{X}, [\], 3)$;
6: **for** $i = 1, \dots, I_3$ **do**
7: Update $\widehat{\mathbf{U}}^{(i)}$ and $\widehat{\mathbf{V}}^{(i)}$ by (16), and $\widehat{\mathbf{S}}^{(i)}$ by (19)
8: Tensor multi-rank estimation: **Only keep** the elements in
 $\text{diag}(\widehat{\mathbf{S}}^{(i)})$ where $\text{diag}(\widehat{\mathbf{S}}^{(i)}) > 0$, and then compute i_{th} entry
of multi-rank $\mathbf{r}_i = \text{length}(\text{diag}(\widehat{\mathbf{S}}^{(i)}))$ and update $\widehat{R} = \mathbf{r}_i$.
9: Update $\widehat{\mathbf{X}}^{(i)} = \widehat{\mathbf{U}}^{(i)}\widehat{\mathbf{S}}^{(i)}\widehat{\mathbf{V}}^{(i)\top}$.
10: **end for**
11: Update \mathcal{E} , \mathcal{Z} , \mathcal{X} by (26), (28) and (30), respectively.
12: Update $\mathcal{Y}_1 = \mathcal{Y}_1 + \gamma(\mathcal{X} - \mathcal{Z})$, $\mathcal{Y}_2 = \mathcal{Y}_2 + \gamma(\mathcal{E} - \mathcal{T} + \mathcal{Z})$.
13: If $\frac{\|\mathcal{X}^{(k+1)} - \mathcal{X}^k\|_F}{\|\mathcal{X}^{(k+1)}\|_F} < tol$, break; otherwise, continue.
14: **end for**
15: **Tensor tubal-rank estimation:** $R = \max\{\mathbf{r}_1, \dots, \mathbf{r}_{I_3}\}$
16: **Step 2–Refinement Scheme:** Repeat the Line 4-7,9-15 using the
estimated tubal-rank R (without rank estimation step in Line 8)
to further refine the model.
17: **Output:** Tensor tubal-rank R , estimated tensor \mathcal{X} .

errors with arbitrary magnitude and distribution. Considering \mathcal{E} has a sparsity property [30], [52], [53], we reformulate the RE_{TNN} model (10) as

$$\begin{aligned} \min_{\mathcal{X}, \mathcal{U}, \mathcal{S}, \mathcal{V}, \mathcal{E}} \quad & \|\mathcal{X} - \mathcal{U} * \mathcal{S} * \mathcal{V}^\top\|_F^2 + \lambda \|\mathcal{S}\|_{\text{TNN}} + \beta \|\mathcal{E}\|_1 \\ \text{s.t.} \quad & \mathcal{T} = \mathcal{X} + \mathcal{E}, \quad \mathcal{U}^\top * \mathcal{U} = \mathcal{I}, \quad \mathcal{V}^\top * \mathcal{V} = \mathcal{I} \end{aligned} \quad (22)$$

where λ and β are the penalty parameters. To solve model (22) via ADMM, we introduce an auxiliary variable \mathcal{Z} for \mathcal{X} to obtain

$$\begin{aligned} \min_{\mathcal{X}, \mathcal{U}, \mathcal{S}, \mathcal{V}, \mathcal{E}, \mathcal{Z}} \quad & \|\mathcal{X} - \mathcal{U} * \mathcal{S} * \mathcal{V}^\top\|_F^2 + \lambda \|\mathcal{S}\|_{\text{TNN}} + \beta \|\mathcal{E}\|_1 \\ \text{s.t.} \quad & \mathcal{T} = \mathcal{Z} + \mathcal{E}, \quad \mathcal{X} = \mathcal{Z}, \quad \mathcal{U}^\top * \mathcal{U} = \mathcal{I} \\ & \mathcal{V}^\top * \mathcal{V} = \mathcal{I}. \end{aligned} \quad (23)$$

Then, the Lagrange function of (23) is

$$\begin{aligned} \mathcal{L} = \quad & \|\mathcal{X} - \mathcal{U} * \mathcal{S} * \mathcal{V}^\top\|_F^2 + \lambda \|\mathcal{S}\|_{\text{TNN}} + \beta \|\mathcal{E}\|_1 \\ & + \frac{\gamma}{2} \|\mathcal{X} - \mathcal{Z} + \mathcal{Y}_1/\gamma\|_F^2 + \frac{\gamma}{2} \|(\mathcal{Z} + \mathcal{E} - \mathcal{T} + \mathcal{Y}_2/\gamma)\|_F^2 \\ \text{s.t.} \quad & \mathcal{U}^\top * \mathcal{U} = \mathcal{I}, \quad \mathcal{V}^\top * \mathcal{V} = \mathcal{I} \end{aligned} \quad (24)$$

where γ is the penalty multiplier and \mathcal{Y}_1 and \mathcal{Y}_2 are the Lagrange dual variables. In the ADMM optimization for (24), $\{\mathcal{U}, \mathcal{S}, \mathcal{V}\}$ are updated by fixing \mathcal{X} and \mathcal{E} . Thus, \mathcal{U} , \mathcal{V} , and \mathcal{S} essentially can be solved by (16) and (19), respectively. We then derive the solutions for \mathcal{E} , \mathcal{Z} , and \mathcal{X} .

Update \mathcal{E} : The Lagrange function (24) with respect to \mathcal{E} is

$$\mathcal{L}_{\mathcal{E}} = \frac{\gamma}{2} \|(\mathcal{E} - (\mathcal{T} - \mathcal{Z} - \mathcal{Y}_2/\gamma))\|_F^2 + \beta \|\mathcal{E}\|_1. \quad (25)$$

Equation (25) is the proximal mapping with respect to the ℓ_1 -norm, which can be obtained by the soft-thresholding

Algorithm 5 Robust Tensor Completion With RE (RTC-RE)

1: **Input:** $\mathcal{P}_{\Omega}(\mathcal{T})$, Ω , λ , initial rank \widehat{R} , K , and tol .
2: **Initialization:** Set $\mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{T})$, $\mathcal{P}_{\Omega^c}(\mathcal{X}) = \mathbf{0}$;
Set $\beta = 1/\sqrt{\max(I_1, I_2)I_3}$ and $\gamma = 1/\lambda$; Initialize
 $\{\{\widehat{\mathbf{u}}_r^{(i)}, \widehat{\mathbf{v}}_r^{(i)}, \mathbf{s}_r^{(i)}\}_{r=1}^{\widehat{R}}\}_{i=1}^{I_3}$ of $\widehat{\mathcal{X}}$ randomly.
3: **Step 1–Rank Estimation (RE):**
4: **for** $k = 1, \dots, K$ **do**
5: $\widehat{\mathcal{X}} = \text{fft}(\mathcal{X}, [\], 3)$;
6: **for** $i = 1, \dots, I_3$ **do**
7: Update $\widehat{\mathbf{U}}^{(i)}$ and $\widehat{\mathbf{V}}^{(i)}$ by (16), and $\widehat{\mathbf{S}}^{(i)}$ by (19)
8: Tensor multi-rank estimation: **Only keep** the elements in
 $\text{diag}(\widehat{\mathbf{S}}^{(i)})$ where $\text{diag}(\widehat{\mathbf{S}}^{(i)}) > 0$, and then compute i_{th} entry
of multi-rank $\mathbf{r}_i = \text{length}(\text{diag}(\widehat{\mathbf{S}}^{(i)}))$ and update $\widehat{R} = \mathbf{r}_i$.
9: Update $\widehat{\mathbf{X}}^{(i)} = \widehat{\mathbf{U}}^{(i)}\widehat{\mathbf{S}}^{(i)}\widehat{\mathbf{V}}^{(i)\top}$.
10: **end for**
11: Update \mathcal{E} , \mathcal{Z} , \mathcal{X} by (35), (37) and (30), respectively.
12: Update $\mathcal{Y}_1 = \mathcal{Y}_1 + \gamma(\mathcal{X} - \mathcal{Z})$, $\mathcal{Y}_2 = \mathcal{Y}_2 + \gamma(\mathcal{E}_{\Omega} - \mathcal{T}_{\Omega} + \mathcal{Z}_{\Omega})$.
13: If $\frac{\|\mathcal{X}^{(k+1)} - \mathcal{X}^k\|_F}{\|\mathcal{X}^{(k+1)}\|_F} < tol$, break; otherwise, continue.
14: **end for**
15: **Tensor tubal-rank estimation:** $R = \max\{\mathbf{r}_1, \dots, \mathbf{r}_{I_3}\}$.
16: **Step 2–Refinement Scheme:** Repeat the Line 4-7,9-15 using the
estimated tubal-rank R (without rank estimation step in Line 8)
to further refine the model.
17: **Output:** Tensor tubal-rank R , estimated tensor \mathcal{X} .

operation [51], that is

$$\mathcal{E} = \text{prox}_{\beta/\gamma}(\mathcal{T} - \mathcal{Y}_1 - \mathcal{Z}). \quad (26)$$

Update \mathcal{Z} : The Lagrange function (24) with respect to \mathcal{Z} is

$$\mathcal{L}_{\mathcal{Z}} = \frac{\gamma}{2} \|\mathcal{X} - \mathcal{Z} + \mathcal{Y}_1/\gamma\|_F^2 + \frac{\gamma}{2} \|(\mathcal{Z} + \mathcal{E} - \mathcal{T} + \mathcal{Y}_2/\gamma)\|_F^2. \quad (27)$$

Setting the partial derivation of (36) with respect to \mathcal{Z} to 0, we can update \mathcal{Z} by

$$\mathcal{Z} = \frac{1}{2}(\mathcal{T} - \mathcal{E} + \mathcal{X} - (\mathcal{Y}_2 - \mathcal{Y}_1)/\gamma). \quad (28)$$

Update \mathcal{X} : The Lagrange function (24) with respect to \mathcal{X} is

$$\mathcal{L}_{\mathcal{X}} = \frac{1}{2} \|\mathcal{X} - \mathcal{U} * \mathcal{S} * \mathcal{V}^\top\|_F^2 + \frac{\gamma}{2} \|\mathcal{X} - \mathcal{Z} + \mathcal{Y}_1/\gamma\|_F^2. \quad (29)$$

Setting the partial derivation of (29) with respect to \mathcal{X} to 0, we can obtain the solution for \mathcal{X}

$$\mathcal{X} = (\mathcal{U} * \mathcal{S} * \mathcal{V}^\top + \gamma \mathcal{Z} - \mathcal{Y}_1)/(1 + \gamma). \quad (30)$$

Finally, we summarize the new robust tensor PCA method, RTPCA-RE, in Algorithm 4.

B. Robust Tensor Completion With Rank Estimation

We further consider a more general and challenging problem: recovering tensors corrupted by simultaneous missing values and sparse noise, that is, robust tensor completion (RTC, also called robust tensor PCA plus tensor completion) problem. Based on the proposed RE_{TNN} model (10), we further propose robust TC with RE

$$\min_{\mathcal{X}, \mathcal{U}, \mathcal{S}, \mathcal{V}, \mathcal{E}} \quad \|\mathcal{X} - \mathcal{U} * \mathcal{S} * \mathcal{V}^\top\|_F^2 + \lambda \|\mathcal{S}\|_{\text{TNN}} + \beta \|\mathcal{E}\|_1$$

$$\begin{aligned} \text{s.t. } \mathcal{P}_{\Omega}(\mathcal{T}) &= \mathcal{P}_{\Omega}(\mathcal{X} + \mathcal{E}), \mathcal{U}^{\top} * \mathcal{U} \\ &= \mathcal{I}, \mathcal{V}^{\top} * \mathcal{V} = \mathcal{I} \end{aligned} \quad (31)$$

where λ and β are the penalty parameters. Similarly, we solve the model (31) using ADMM by introducing an auxiliary variable \mathcal{Z} for \mathcal{X} and, thus, have

$$\begin{aligned} \min_{\mathcal{X}, \mathcal{U}, \mathcal{S}, \mathcal{V}, \mathcal{E}, \mathcal{Z}} \quad & \left\| \mathcal{X} - \mathcal{U} * \mathcal{S} * \mathcal{V}^{\top} \right\|_F^2 + \lambda \|\mathcal{S}\|_{\text{TNN}} + \beta \|\mathcal{E}\|_1 \\ \text{s.t. } \quad & \mathcal{P}_{\Omega}(\mathcal{T}) = \mathcal{P}_{\Omega}(\mathcal{Z} + \mathcal{E}), \mathcal{X} = \mathcal{Z} \\ & \mathcal{U}^{\top} * \mathcal{U} = \mathcal{I}, \mathcal{V}^{\top} * \mathcal{V} = \mathcal{I}. \end{aligned} \quad (32)$$

Then, the Lagrange function of (32) is

$$\begin{aligned} \mathcal{L} &= \left\| \mathcal{X} - \mathcal{U} * \mathcal{S} * \mathcal{V}^{\top} \right\|_F^2 + \lambda \|\mathcal{S}\|_{\text{TNN}} + \beta \|\mathcal{E}\|_1 \\ &+ \frac{\gamma}{2} \|\mathcal{X} - \mathcal{Z} + \mathcal{Y}_1/\gamma\|_F^2 + \frac{\gamma}{2} \\ &\times \|\mathcal{P}_{\Omega}(\mathcal{Z} + \mathcal{E} - \mathcal{T} + \mathcal{Y}_2/\gamma)\|_F^2 \\ \text{s.t. } \quad & \mathcal{U}^{\top} * \mathcal{U} = \mathcal{I}, \mathcal{V}^{\top} * \mathcal{V} = \mathcal{I} \end{aligned} \quad (33)$$

where γ is the Lagrange multiplier and \mathcal{Y}_1 and \mathcal{Y}_2 are the Lagrange dual variables. Similar to the RTPCA-RE optimization, \mathcal{U} , \mathcal{V} , and \mathcal{S} of model (33) essentially can be solved by (16) and (19), respectively. In the following, we derive the solutions for \mathcal{E} , \mathcal{Z} , and \mathcal{X} .

Update \mathcal{E} : The Lagrange function (33) with respect to \mathcal{E} is

$$\mathcal{L}_{\mathcal{E}} = \frac{\gamma}{2} \|\mathcal{P}_{\Omega}(\mathcal{E} - (\mathcal{T} - \mathcal{Z} - \mathcal{Y}_2/\gamma))\|_F^2 + \beta \|\mathcal{E}\|_1. \quad (34)$$

With the soft-thresholding technique [51], the solution of \mathcal{E} is

$$\begin{cases} \mathcal{E}_{\Omega^c} = \mathbf{0} \\ \mathcal{E}_{\Omega} = \text{prox}_{\beta/\gamma}(\mathcal{T}_{\Omega} - \mathcal{Y}_1_{\Omega} - \mathcal{Z}_{\Omega}). \end{cases} \quad (35)$$

Update \mathcal{Z} : The Lagrange function (33) with respect to \mathcal{Z} is

$$\begin{aligned} \mathcal{L}_{\mathcal{Z}} &= \frac{\gamma}{2} \|\mathcal{X} - \mathcal{Z} + \mathcal{Y}_1/\gamma\|_F^2 \\ &+ \frac{\gamma}{2} \|\mathcal{P}_{\Omega}(\mathcal{Z} + \mathcal{E} - \mathcal{T} + \mathcal{Y}_2/\gamma)\|_F^2. \end{aligned} \quad (36)$$

Setting the partial derivation of (36) with respect to \mathcal{Z} to 0, we can obtain the solution for \mathcal{Z}

$$\begin{cases} \mathcal{Z}_{\Omega^c} = \mathcal{X}_{\Omega^c} + \mathcal{Y}_1_{\Omega^c}/\gamma \\ \mathcal{Z}_{\Omega} = \frac{1}{2}(\mathcal{T}_{\Omega} - \mathcal{E}_{\Omega} + \mathcal{X}_{\Omega} - (\mathcal{Y}_2_{\Omega} - \mathcal{Y}_1_{\Omega})/\gamma). \end{cases} \quad (37)$$

Update \mathcal{X} : We can obtain the solution of \mathcal{X} using (30).

Finally, we summarize this new robust tensor completion with rank estimation RTC-RE in Algorithm 5.

Refinement Scheme: After the RE steps, both RTPCA-RE and RTC-RE refine their own models by explicitly utilizing the estimated rank to further optimize the variables.

Remark 4: RTPCA-RE is a special case of RTC-RE to handle data corrupted by sparse noise only without missing values. The two robust tensor methods RTPCA-RE and RTC-RE inherit the ability of RE_{TNN} and can accurately estimate the tensor rank in the presence of missing data and/or gross corruptions under mild conditions (e.g., enough observed entries or a small proportion of sparse noise). By explicitly making use of the correctly estimated rank under the refinement schemes, these two methods can achieve successful robust tensor recovery (Refer to Section VI about experimental evaluation).

V. ALGORITHM ANALYSIS AND GENERALIZATION

A. Computational Complexity Analysis

- 1) For RE_{TNN} (Algorithm 2), at each iteration, the computational complexity includes the following.
 - a) The cost of conducting the DFT of $\{\widehat{\mathbf{X}}^{(i)}\}_{i=1}^{I_3}$ is $O(I_1 I_2 I_3 \log I_3)$ in line 4.
 - b) The cost of updating $\{\widehat{\mathbf{U}}^{(i)}, \widehat{\mathbf{S}}^{(i)}, \widehat{\mathbf{V}}^{(i)}\}_{i=1}^{I_3}$ in line 6 is $O(\widehat{R} I_1 I_2 I_3)$. Here, $\widehat{R} = \mathbf{r}_1 = \dots = \mathbf{r}_{I_3}$ is the initialization of tubal rank of \mathcal{X} and we set $\widehat{R} < \min(I_1, I_2)$ in general, for example, $\widehat{R} = 1/2$ or $1/4 \times \min(I_1, I_2)$.
 - c) The cost of the soft-thresholding operation of updating $\{\widehat{\mathbf{s}}_r^{(i)}\}_{r=1}^{\widehat{R}}\}_{i=1}^{I_3}$ is $O(\widehat{R} I_3)$.
 - d) The cost of updating \mathcal{X} by inverse DFT is $O(I_1 I_2 I_3 \log I_3)$ in line 10.

Hence, the total cost of RE_{TNN} at each iteration is $O(I_1 I_2 I_3 \log I_3 + \widehat{R} I_1 I_2 I_3)$.

- 2) For TC-RE (Algorithm 3), it consists of the cost of RE_{TNN} and the computation of tubal-rank R approximation of \mathcal{X} (truncated t-SVD of \mathcal{X}). The time complexity of rank- R of t-SVD is $O(R I_1 I_2 I_3)$. Thus, the computational complexity of TC-RE is the same as RE_{TNN} in each iteration while TC-RE involves more iterations.
- 3) For the two robust methods RTPCA-RE (Algorithm 4) and RTC-RE (Algorithm 5), they have similar computational complexity as TC-RE but need cost more time to compute the soft-thresholding of \mathcal{E} and \mathcal{Z} [i.e., (26) and (28), and (35) and (37), respectively]. Both the time cost of computing \mathcal{E} and \mathcal{Z} are $O(I_1 I_2 I_3)$ per iteration. In short, the time complexity of RTPCA-RE/RTC-RE is $O(I_1 I_2 I_3 \log I_3 + \widehat{R} I_1 I_2 I_3)$ per iteration.

1) Comparison With the TNN-Based Methods: Compared with the existing TNN-based methods (e.g., TNN [29], t-TNN [32], and TRPCA [30]), whose costs at each iteration are $O(I_1 I_2 I_3 \log I_3 + \min(I_1, I_2) I_1 I_2 I_3)$, the proposed RE_{TNN} ($O(I_1 I_2 I_3 \log I_3 + \widehat{R} I_1 I_2 I_3)$) only involves the computation of \widehat{R} ($< \min(I_1, I_2)$) pairs of singular vectors, which effectively avoids the expensive full SVD when computing the tensor singular value thresholding. After obtaining the low true tubal-rank R ($< \widehat{R}$), TC-RE/RTPCA-RE/RTC-RE also has a lower computational cost than TNN-based methods per iteration.

B. Generalization of the Proposed Methods

In the preceding sections, we propose four methods: 1) RE_{TNN}; 2) TC-RE; 3) RTPCA-RE; and 4) RTC-RE, to solve the problems of tensor RE, TC, robust tensor PCA, and robust TC, respectively. To generalize the proposed methods, we propose a general unified framework, that is, Robust t-SVD with RE (RtSVD-RE)

$$\begin{aligned} \min_{\mathcal{X}, \mathcal{E}, \mathcal{U}, \mathcal{S}, \mathcal{V}} \quad & \left\| \mathcal{X} - \mathcal{U} * \mathcal{S} * \mathcal{V}^{\top} \right\|_F^2 + \lambda \|\mathcal{S}\|_{\text{TNN}} + \beta f(\mathcal{E}) \\ \text{s.t. } \quad & \mathcal{P}_{\Omega}(\mathcal{T}) = \mathcal{P}_{\Omega}(\mathcal{X} + \mathcal{E}), \mathcal{U}^{\top} * \mathcal{U} \\ & = \mathcal{I}, \mathcal{V}^{\top} * \mathcal{V} = \mathcal{I} \end{aligned} \quad (38)$$

where the first and second terms contribute to RE, and the third term $\beta f(\mathcal{E})$ is a cost function that can be variant for different problems.

1) Connection to the Proposed Methods:

- 1) If $\beta = 0$, (38) is specified to RE_{TNN} model (10).
- 2) If $\lambda = \beta = 0$, (38) is specified to the TC-RE model (21).
- 3) If $\beta > 0$, $f(\mathcal{E}) = \|\mathcal{E}\|_1$, and $\mathbf{\Omega} = \mathbf{1} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ (i.e., no missing), (38) is specified to RTPCA-RE model (22).
- 4) If $\beta > 0$ and $f(\mathcal{E}) = \|\mathcal{E}\|_1$, the general model (38) is specified to the RTC-RE model (31).

2) *Possible Specific Cases of the RtSVD-RE Model:* In addition to the four proposed methods as specific cases of the general model, there are some promising variants, for example:

- 1) if $\beta > 0$ and $f(\mathcal{E}) = \|\mathcal{E}\|_F$, model (38) is specified to a robust tensor model to handle both missing data and Gaussian noise. Furthermore, if $\mathbf{\Omega} = \mathbf{1}$, the specified model is variant to recover data with the Gaussian noises;
- 2) if $\beta > 0$ and $f(\mathcal{E}) = \|\mathcal{E}\|_{2,1}$, the general model is specified to a robust model used for outliers detection inspired by [35], which combines t-SVD factorization with $\ell_{2,1}$ -norm regularization, and provides good results in outlier detection;
- 3) if $\beta < 0$ and $f(\mathcal{E}) = f(\mathcal{S}) = \|\mathcal{S}^{(i)} - \widehat{\mathcal{S}}\|_F$, model (38) can be specified to a feature extraction model inspired by [54]

$$\begin{aligned}
 & \min_{\mathcal{X}, \mathcal{U}, \mathcal{S}, \mathcal{V}} \left\| \mathcal{X} - \mathcal{U} * \mathcal{S} * \mathcal{V}^\top \right\|_F^2 \\
 & \quad + \lambda \left\| \mathcal{S} \right\|_{\text{TNN}} - \beta' f \left\| \mathcal{S}^{(i)} - \widehat{\mathcal{S}} \right\|_F \\
 & \text{s.t. } \mathcal{P}_{\mathbf{\Omega}}(\mathcal{T}) = \mathcal{P}_{\mathbf{\Omega}}(\mathcal{X}), \widehat{\mathcal{S}} = \frac{1}{M} \sum_{i=1}^M \mathcal{S}^{(i)} \\
 & \quad \mathcal{U}^\top * \mathcal{U} = \mathcal{I}, \mathcal{V}^\top * \mathcal{V} = \mathcal{I}
 \end{aligned} \quad (39)$$

where $-\beta' = \beta < 0$ and $\widehat{\mathcal{S}} = (1/M) \sum_{i=1}^M \mathcal{S}^{(i)}$, $i = 1, \dots, M$ is the mean of extracted features, M is the total number of data samples \mathcal{S} is the f -diagonal tensor, which consists of the extracted features of incomplete tensors \mathcal{T} with observed entries in $\mathbf{\Omega}$. Model (39) explores the relationship among data samples via maximizing feature variance $[f(\mathcal{S})]$ and, meanwhile, estimates the missing entries via low-rank t-SVD approximation, leading to informative features extracted directly from observed entries. For example, given a face dataset that has M face images with missing pixels, we consider each face sample (size $I_1 \times I_2$) as each frontal slice of the entire dataset $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times M}$. To extract informative features from these incomplete face images for further application (e.g., face recognition), we can use model (39) to obtain the solution by viewing \mathcal{S} as the extracted features ($\mathcal{S}^{(i)}$ consists features from each incomplete face image). Here, the RE function can be used to determine the number of features (low dimensional). Thus, this specific model (39) is a tensor feature extraction method that aims to extract low-dimensional features from incomplete data directly.

VI. EXPERIMENTS

A. Experimental Setup

1) *Compared Methods:* We compare the proposed methods with the competing 17 methods in three categories.⁶

- 1) *Seven TC Methods:* Two Tucker-/CP-based methods: a) HaLRTC [3] and b) TNCP [56] and three t-SVD-based completion methods: a) TNN [29]; b) t-TNN [32]; and c) TCTF [20], smooth tensor tree completion (STTC) [57], and TC via tensor ring with balanced unfolding (TRBU) [58].
- 2) *Seven Robust PCA Methods:* Two matrix methods: a) RPCA [59] and b) PSSV [60]; two Tucker-based: a) SNN [41] and b) RKCA [21]; three t-SVD-based: a) OR-TRPCA [35]; b) IRTPCA [61]; and c) TRPCA [30].
- 3) *Three Robust Tensor Completion Methods:* a) BRTF [6] (CP-based); b) GWLRTF-Tucker [24] (Tucker-based); and c) LRTCR-TNN [34] (t-SVD-based).

In addition, to further analyze the effectiveness of the proposed RE_{TNN}, we replace the heuristic RE of TCTF as our RE_{TNN}, that is, RE_{TNN} + TCTF, and we denote this new combined method as TCTF-RE. We evaluate how RE_{TNN} improves the performance of TCTF in Section VI-D1 and VI-E. Thus, we actually compare 18 competing methods in total.

2) *Data Setting:* We generate the *synthetic* tensors with low tubal-ranks following [20] and test the real-world tensor (images and videos) commonly used in the literature. For missing data setting, we uniformly sample 10%–90% entries (observed entries) of each tensor at random for training and denote “SR” as sample ratio, and the ratio of missing entries = 1–SR. For the corrupted data setting, we randomly select 5%–50% entries of each tensor corrupted by gross corruptions (e.g., sparse noise) drawn from a uniform distribution $U(-|H|, |H|)$, $H = \max(\mathcal{X}(\cdot))$ following [6]. We denote “CR” as gross corruptions ratio to refer to the ratio of entries of a tensor corrupted by sparse noise.

3) *Parameter Setting:* We set the maximum iterations $K = 500$ and the stopping tolerance $\text{tol} = 1e - 15$ for all methods, and $\lambda = 10 \times ([\max(\mathcal{P}_{\mathbf{\Omega}}(\mathcal{T}))]/[\sqrt{\text{mean}(\text{size}(\mathcal{T}))} \times (1 - \text{SR})])$, the initial rank $\widehat{R} = \text{round}(1/2 \times \min(I_1, I_2))$ by default for the proposed methods (see the analysis in Section VI-B). For the parameter β regarding the sparse error regularization in the robust methods, we use the default parameters recommended by [30] and [52] for both their approaches and ours, that is, $\beta = 1/\sqrt{(\max(I_1, I_2)I_3)}$. Other parameters of the compared methods are tuned guided by their original papers to give the best performance. We measure tensor recovery performance using the widely used relative square error (RSE): $\|\mathcal{X} - \mathcal{T}\|_F / \|\mathcal{T}\|_F$, where \mathcal{X} is viewed as recovered successfully if $\text{RSE} < 10^{-3}$ [3]. The average results of ten runs are reported.

⁶We have tested other CP-/Tucker-based methods, such as [23] and [55], and also compare other t-SVD based methods, for example, [19], while they obtain similar or worse results than these 17 methods. So, their results are not reported here for simplicity.

B. Tensor Rank Estimation and Completion Sensitivity on λ and \hat{R}

We study the parameter sensitivity of RE_{TNN} and TC-RE regarding two key parameters λ and \hat{R} on the synthetic [100 × 100 × 100 ($R = 5$)] and real-world tensors [House image ($R = 26$) and Windmill video ($R = 17$)]. The results are shown in Appendix D of the supplementary material: a wide range of values of the regularization parameter λ and initial rank \hat{R} lead to good tensor RE and recovery performance in most cases, excepting for the cases of too many missing entries. In other words, the RE and completion performance of RE_{TNN} and TC-RE are stable and not sensitive to the values of λ and \hat{R} in general. We thus can fix λ using a fixed formula, say, $\lambda = \rho$ (e.g., $\rho = 5, \dots, 10$) × ((max($\mathcal{P}_\Omega(\mathcal{T})$))/[$\sqrt{\text{mean}(\text{size}(\mathcal{T}))} \times (1 - \text{SR})$]) by default. and set $\hat{R} = \text{round}(1/2 \times \min(I_1, I_2))$ by default, based on our preliminary studies. Note that RTPCA-RE and RTC-RE inherit the ability of RE_{TNN} and TC-RE and also show stale performance with a wide range of parameters, such as RE_{TNN} and TC-RE. For simplicity, we thus do not show the parameter sensitivity analysis of these two methods. In addition, for the parameter β regarding the sparse error regularization in the robust methods, we use the default parameters recommended by [30] and [52] for both their approach and ours, that is, $\beta = 1/\sqrt{\max(I_1, I_2)I_3}$.

In summary, the proposed methods are not sensitive to the parameters and stably yield good tensor RE and completion results, so we do not need to tune λ and \hat{R} in general and can compute them using default formulas.

C. Convergence Study

We analyze the convergence of our methods on the synthetic tensor 100 × 100 × 100 ($R = 5$) in terms of training error: $\|\mathcal{X}^{(k+1)} - \mathcal{X}^k\|_F/\|\mathcal{X}^{(k+1)}\|_F$ and estimated ranks (see the convergence curves in Appendix D of the supplementary material). RE_{TNN} converges to the true rank solution within 30 iterations and TC-RE converges within 200 iterations for the easy problems (e.g., SR > 30%), while they need more iterations to achieve convergence if there are more missing entries. For the robust methods RTPCA-RE and RTC-RE, they achieve the convergence of RE within 100 iterations and converge to very small training error ($1e-15$) within about 400 iterations, which are more than that of RE_{TNN} and TC-RE due to the corruption of noise (e.g., CR < 40%).

In general, we can stop the running of RE step within 30 (for TC-RE)/100 (for RTPCA-RE/RTC-RE) iterations to obtain the true ranks while letting TC-RE/RTPCA-RE/RTC-RE achieve better recovery performance with more iterations (setting the maximum iterations $K = 500$ by default).

D. Evaluation of Synthetic Data

To evaluate the proposed methods, we first conduct tests on two third-order synthetic tensors with size $I_1 = I_2 = I_3 = I = \{100, 500\}$ with true tubal-rank $R = \{5, 50\}$, respectively. Setting the tensors with various problems, we report the results in Tables I–III, where we highlight the *best* recovery results and the *correct* estimated ranks in **bold** fonts and second-best results in underline.

TABLE I
COMPARISON OF RECOVERING TENSORS WITH MISSING ENTRIES

Problem	TNN [29]		TCTF [20]		TCTF-RE		RE _{TNN}		TC-RE				
	Data	SR (%)	RSE	Time	RSE	Time	Est.R	RSE	Time	RSE	Time	Est.R	
$I = 100$	40	2.15E-04	202.4	2.52E-02	230.6	9	<u>2.20E-14</u>	175.5	4.70E-03	16.0	5.85E-15	127.1	5
$R = 5$	60	1.26E-04	86.6	8.07E-03	129.6	2	<u>4.01E-15</u>	183.9	2.97E-03	8.6	1.92E-15	54.7	5
$R = 50$	80	5.39E-05	46.3	1.99E-03	218.0	6	5.43E-16	190.3	2.14E-03	5.4	<u>6.56E-16</u>	28.5	5
$I = 500$	40	<u>2.95E-04</u>	29098.6	2.25E-02	21711.4	57	1.10E-02	19023.5	1.21E-03	1711.4	8.73E-15	30128.9	50
$R = 5$	60	1.39E-04	14608.2	1.01E-03	14568.3	18	<u>6.77E-12</u>	16947.9	8.29E-04	841.2	3.84E-15	9946.3	50
$R = 50$	80	6.95E-05	6784.9	2.69E-04	15813.0	43	<u>9.87E-14</u>	15280.2	5.98E-04	545.7	1.10E-15	4447.9	50

TABLE II
COMPARISON OF RECOVERING TENSORS WITH GROSS CORRUPTIONS

Problem	TRPCA [30]			RTPCA-RE (RE)			
	Data	CR (%)	RSE	Time	RSE	Time	Est.R
$I = 100$	10	8.89E-08	543.5	385.6	9.89E-16 (1.53E-05)	84.7 (21.3)	5
$R = 5$	30	<u>2.75E-07</u>	398.7	398.7	2.73E-14 (2.19E-05)	137.5 (39.5)	5
$R = 50$	50	<u>2.09E-02</u>	44848.5	60232.7	1.20E-02 (4.33E-02)	154.4 (46.2)	25
$I = 500$	10	<u>1.41E-08</u>	44848.5	60232.7	8.23E-16 (1.23E-05)	47836.9 (8820.5)	50
$R = 50$	30	1.04E-04	85886.9	85886.9	3.38E-13 (2.11E-05)	46839.8 (13604.0)	50
$R = 50$	50	<u>3.68E-03</u>	85886.9	85886.9	1.92E-03 (7.49E-03)	69576.7 (19389.5)	80

TABLE III
COMPARISON OF RECOVERING TENSORS WITH BOTH MISSING ENTRIES AND GROSS CORRUPTIONS (CR = 10%)

Problem	LRTC-RE [34]			TC-RE (RE)			
	Data	SR (%)	RSE	Time	RSE	Time	Est.R
$I = 100$	40	<u>2.79E-04</u>	597.3	616.1	1.10E-10 (1.07E-03)	423.3 (114.4)	5
$R = 5$	60	5.82E-07	611.6	611.6	1.53E-15 (5.72E-05)	379.3 (52.4)	5
$R = 50$	80	<u>3.84E-07</u>	119799.7	79847.1	8.47E-16 (6.77E-05)	317.8 (38.9)	5
$I = 500$	40	1.43E-03	119799.7	79847.1	9.99E-04 (1.42E-03)	86200.2 (28171.8)	50
$R = 5$	60	<u>4.74E-05</u>	51187.0	51187.0	6.07E-08 (8.96E-04)	70170.6 (12843.4)	50
$R = 50$	80	<u>6.76E-08</u>	51187.0	51187.0	1.45E-15 (8.07E-04)	63807.1 (5605.3)	50

1) *Tensor Completion*: We report the comparison again of two t -SVD methods and the five TC methods in Table I: one is TNN [29], which is the best performing method on synthetic tensors, and the other is TCTF [20], which is the most recent t -SVD-based TC method with RE. Table I shows that TC-RE consistently outperforms TNN and TCTF by several orders of magnitude in all cases. TCTF fails to correctly estimate the tensor rank in these cases, resulting in bad results. With the correct ranks determined by RE_{TNN}, TCTF-RE achieves much better recovery results than TCTF in all cases and even outperforms our RTC-RE in one case (i.e., on 100 × 100 × 100 ($R = 5$) with SR = 80%). These results verify our assumption: accurate RE can improve the recovery performance for t -SVD-based TC methods. TC-RE achieves much better recovery results than RE_{TNN} because RE_{TNN} focuses on RE only while TC-RE explicitly uses the correct rank to improve the completion accuracy via the relaxing strategy.

2) *Robust Tensor PCA*: As TRPCA is the best performing existing robust tensor PCA algorithm, we report its results compared with ours in Table II. Our RTPCA-RE clearly outperforms TRPCA in all cases even with data that have a large ratio of corruptions (CR = 50%), where it is difficult to estimate the correct tubal-rank. Moreover, using the refinement

TABLE IV
COMPARISON OF COMPLETION RESULTS MEASURED IN RSE, TIME COST (SECOND), AND ESTIMATED RANK (EST. R) ON REAL IMAGES AND VIDEOS

Problem	HaLRTC [3]			TNCP [56]		TNN [29]		t-TNN [32]			TCTF [20]		STTC [57]			TRBU [58]			TCTF-RE		RE _{TNN}		TC-RE				
	SR	RSE	Time	RSE	Time	RSE	Time	RSE	Time	RSE	Time	Est. R	RSE	Time	RSE	Time	RSE	Time	RSE	Time	RSE	Time	RSE	Time	Est. R		
House (256×256×3) $R = 26$	40	<u>1.54E-03</u>	256.6	2.16E-02	327.1	2.38E-03	441.6	4.11E-02	27.0	1.17E-01	34.4	36	5.58E-03	156.9	3.19E-03	17.3	–	–	–	–	4.95E-02	52.2	4.50E-02	13.5	9.65E-07	234.2	26
	60	<u>1.18E-07</u>	233.3	1.38E-02	477.2	2.07E-04	246.4	2.12E-02	23.2	6.80E-02	37.2	7	4.12E-03	219.9	3.39E-03	16.3	–	–	–	–	1.90E-02	54.7	2.99E-02	13.6	1.03E-13	303.0	26
	80	<u>5.68E-08</u>	224.7	8.29E-03	469.4	7.17E-05	94.0	8.39E-03	27.6	4.44E-02	49.1	3	3.00E-03	162.6	4.21E-03	14.0	–	–	–	–	2.91E-04	58.2	2.14E-02	4.7	4.49E-15	55.3	26
Lenna (256×256×3) $R = 29$	40	1.44E-02	56.1	3.75E-02	298.0	<u>2.38E-03</u>	186.3	4.97E-02	32.9	1.36E-01	22.5	71	4.60E-02	193.5	3.12E-02	20.9	–	–	–	–	6.77E-03	29.2	6.70E-03	8.2	2.55E-06	106.4	29
	60	<u>2.58E-05</u>	62.6	2.27E-02	299.6	2.07E-04	199.6	2.71E-02	20.5	2.71E-02	14.3	61	3.58E-02	161.3	3.40E-02	15.4	–	–	–	–	1.43E-02	30.7	4.47E-02	4.1	1.16E-14	75.5	29
	80	<u>7.96E-08</u>	50.4	1.31E-02	304.4	7.17E-05	64.2	1.23E-02	22.7	5.13E-02	13.8	18	2.69E-02	209.6	4.01E-02	17.0	–	–	–	–	4.76E-04	30.1	3.20E-02	4.3	3.86E-15	47.1	29
River Otter (321×481×3) $R = 36$	40	1.14E-02	247.9	4.28E-02	423.2	<u>7.41E-03</u>	246.2	5.06E-02	23.1	8.71E-01	60.6	39	8.75E-03	345.9	–	–	–	–	–	–	9.33E-02	73.6	4.90E-02	15.8	9.42E-08	183.6	36
	60	<u>1.65E-07</u>	227.9	2.73E-02	490.0	4.45E-04	290.5	2.60E-02	20.3	1.98E-01	51.1	28	7.75E-03	320.9	–	–	–	–	–	–	2.67E-02	75.2	3.21E-02	10.1	1.53E-14	135.9	36
	80	<u>7.85E-08</u>	153.6	1.58E-02	347.9	8.27E-05	121.2	9.99E-03	28.0	7.64E-02	39.5	4	5.11E-03	332.4	–	–	–	–	–	–	1.18E-02	82.4	2.27E-02	5.3	6.59E-15	50.6	36
Akiyo (146×176×150) $R = 13$	40	2.85E-02	308.1	2.38E-02	299.0	<u>5.66E-03</u>	2350.0	1.61E-02	429.7	8.71E-01	568.4	27	9.88E-03	977.9	–	–	–	–	–	–	8.96E-02	447.1	1.28E-02	219.8	1.75E-03	1611.4	13
	60	1.19E-02	314.1	1.85E-02	303.7	<u>1.78E-04</u>	1583.0	8.71E-03	426.3	1.98E-01	397.4	19	8.72E-03	660.5	–	–	–	–	–	–	1.09E-02	461.1	8.48E-03	91.3	5.15E-07	1741.5	13
	80	5.65E-03	338.1	1.26E-02	254.0	<u>8.19E-05</u>	350.9	4.22E-03	456.1	7.64E-02	271.9	3	5.05E-03	664.8	–	–	–	–	–	–	5.75E-04	396.7	5.99E-03	52.5	1.82E-14	825.7	13
Bus (144×120×150) $R = 14$	40	1.73E-01	224.9	1.48E-01	178.0	1.81E-03	1419.0	8.57E-02	217.1	2.62E-01	520.8	25	1.87E-02	819.1	–	–	–	–	–	–	<u>6.35E-04</u>	199.8	1.28E-02	225.2	1.39E-09	1874.3	14
	60	1.10E-01	248.5	1.19E-01	174.8	<u>1.80E-04</u>	203.2	5.28E-02	241.3	2.66E-02	505.0	23	1.54E-02	657.7	–	–	–	–	–	–	6.35E-04	208.3	8.48E-03	109.6	8.11E-15	892.2	14
	80	6.58E-02	245.0	8.38E-02	161.2	<u>8.35E-05</u>	103.6	2.87E-02	260.6	5.20E-02	373.8	13	1.07E-02	639.3	–	–	–	–	–	–	6.35E-04	217.1	5.99E-03	55.9	3.43E-15	375.6	14
Windmill (304×480×40) $R = 17$	40	2.47E-02	553.8	3.21E-02	605.8	<u>5.66E-04</u>	2735.1	1.16E-02	216.8	2.62E-01	723.7	36	3.54E-03	611.7	–	–	–	–	–	–	4.56E-02	587.3	1.94E-02	244.4	5.29E-07	2046.9	17
	60	9.70E-03	598.5	2.47E-02	598.2	1.78E-04	627.9	6.29E-03	210.7	2.66E-02	579.5	19	2.37E-03	724.1	–	–	–	–	–	–	<u>4.56E-08</u>	644.7	1.31E-02	99.1	1.81E-14	1679.4	17
	80	3.57E-03	468.7	1.70E-02	599.8	8.19E-05	305.8	3.13E-03	215.4	5.20E-02	565.2	2	1.74E-03	1596.0	–	–	–	–	–	–	8.18E-16	641.3	9.26E-03	54.9	<u>3.17E-15</u>	385.1	17

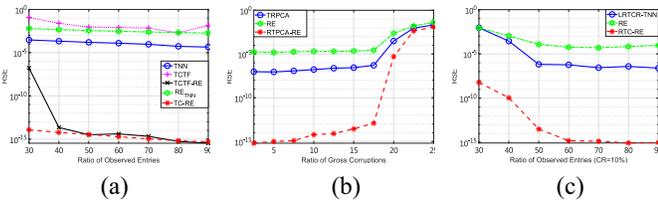


Fig. 1. Comparison of recovery results the proposed methods with other t-SVD methods for tensor $100 \times 100 \times 100$ ($R = 5$) with various ratios of observed entries (SR) and/or gross corruptions (CR). (a) TC. (b) Robust tensor PCA. (c) Robust TC.

scheme, the recovery accuracies of RTPCA-RE are improved significantly over those of the RE step.

3) *Robust Tensor Completion*: By applying our RTC-RE on the synthetic data with both missing entries and gross corruptions, we further demonstrate the superiority of making use of the true rank information in Table III: RTC-RE not only successfully determines the true ranks in all cases, but also outperforms the state-of-the-art robust t-SVD method LRTC-TNN by several orders.

In addition, we also show the comparison in more settings on the $100 \times 100 \times 100$ ($R = 5$) with $SR = 30\% - 90\%$ and/or $CR = 5\% - 50\%$ in Fig. 1, where our methods are the best.

In summary, the proposed methods can estimate the true tubal rank of tensors under mild conditions (e.g., $CR < 50\%$) and outperform the existing competing t-SVD methods with significant improvements. Our methods are also faster on the whole, particularly with efficient RE. Besides, with correct estimated ranks, not only our methods improve the recovery performance (comparing TC-RE with RE_{TNN}, and RTPCA-RE/RTC-RE with their RE step) but also achieve much better results (e.g., TCTF-RE) than other methods (e.g., TCTF). These results support our claims that current t-SVD-based methods do not make use of rank information resulting in degraded performance, while modeling tensors with accurate

ranks significantly improves the recovery results. In the following, the proposed methods will be applied in various real-world applications and compared with other methods to further demonstrate the superior capability of our methods.

E. Application to Image and Video Completion

To evaluate the performance of the proposed RE_{TNN} and TC-RE, we compare them against the five competing algorithms in terms of recovery accuracy (RSE) and the RE results for six real-world images and videos. Note that these natural-color images/videos are approximately low rank, as a small number of their leading singular values dominate the information and a large number of singular values are very close to zero (refer to Fig. 1 in the supplementary material). Therefore, to directly evaluate the RE for the real data, we examine their singular values and truncate the t-SVD to derive the tensors with exact low tubal-ranks: {26 (House), 29 (Lenna), 36 (River Otter), 13 (Akiyo), 14 (Bus), 17 (Windmill)}.⁷ We test all of the methods on all tensors with 10%–90% observed entries, and our TC-RE outperforms other methods in all cases. For simplicity, we report the results of $SR = \{40\%, 60\%, 80\%\}$ in Table IV.⁸

1) *Tensor Rank Estimation*: We compare the RE results obtained by our RE_{TNN} and TCTF (the other four TC methods do not have the RE step). Table IV shows that RE_{TNN} correctly determines the tubal ranks in all cases, while TCTF underestimates or overestimates the true ranks. RE_{TNN} is robust to the initial rank, while TCTF obtains different ranks using different

⁷Image/Videos with truncated exact low-rank are widely studied in the literature [62]–[65]. In this work, we follow this line to contribute better alternatives for robust tensor recovery with accurate RE.

⁸In Table IV, “–” refers to that TRBU cannot handle the tensors with arbitrary size as it requires a good tensor ring structure, while it can process the House and Lenna with size $2^8 \times 2^8 \times 3$ more efficiently.



Fig. 2. Recovery results on the Windmill video and the *Lenna* image. The first and second column figures are the low tubal-rank tensors and the incomplete tensors (SR = 50%), respectively. The third column to the tenth figures are recovered tensors by HaLRTC, TNCP, TNN, t-TNN, TCTF, TCTF-RE, RE_{TNN}, and TC-RE, respectively, (best viewed on screen or see larger figures in Appendix D of the supplementary material).

rank initializations. These results demonstrate the superiority of our RE model.

2) *Image and Video Completion*: Table IV shows that TC-RE consistently recovers all the tensors successfully (RSE < 10^{-3}) with much smaller reconstruction errors except in one case (Akiyo video with SR = 40%), where the observed entries are not sufficient. Nevertheless, TC-RE consistently outperforms all other methods by several orders of magnitude in all cases, excepting one (i.e., the Windmill video with SR = 80%), where TCTF-RE obtains the best results, benefiting from the accurate RE of our RE_{TNN}. HaLRTC and TNN are the best performing existing algorithms in the cases of recovering images and videos, respectively. Moreover, TNCP and TCTF successfully predict the missing entries in only a few cases because their inaccurate CP/tubal-RE deteriorates their recoverability. Although TCTF-RE consistently outperforms TCTF in all cases in Tables I and IV, it fails to retain its good completion performance on real tensors with fewer observed entries (e.g., SR < 80%), where TC-RE still yields good recovery results. This verifies that TC-RE can achieve the optimal recovery solutions with the correctly estimated ranks and sufficient observations. In addition, TC-RE also consistently achieves much better results than RE_{TNN} in real-world tensors such as in synthetic cases, which further confirms the effectiveness and superiority of our relaxing strategy with accurate RE.

Fig. 2 shows examples of recovery results on the Windmill video and the *Lenna* image with 50% missing entries.

3) *Time Cost*: Table IV shows that the proposed RE_{TNN} is, on the whole, the fastest among all methods including the compared *t*-SVD-based TC methods. That is, because RE_{TNN} avoids computing the full *t*-SVD, which leads to less CPU time cost. TC-RE is not as fast as RE_{TNN} as it needs more iterations to compute the truncated *t*-SVD of \mathcal{X} after RE, but it is not the slowest and achieves much better recovery results than other methods with similar computational costs (e.g., TNN).

F. Application to Video Denoising

To evaluate the performance of the proposed RTPCA-RE, we apply it to denoising of the Basketball video ($144 \times 146 \times 40$, $R = 15$) corrupted by CR = 25% sparse noise. Table V shows that RTPCA-RE achieves much better recovery accuracy than the compared six methods. As RPCA and SNN simply assume the noise to be Gaussian, they thus cannot handle gross corruptions successfully, although they are faster than other methods. PSSV shows good results for matrix data on the

TABLE V
COMPARISON OF VIDEO DENOISING ON THE BASKETBALL VIDEO

	RPCA	SNN	PSSV	IRTPCA	OR-TPCA	RKCA	TRPCA	RTPCA-RE (RE)
	[52], [59]	[41]	[60]	[61]	[35]	[21]	[30]	
RSE	5.65E-01	3.14E-01	1.81E-01	5.19E-03	1.85E-01	8.45E-02	2.76E-01	5.61E-12 (9.34E-05)
Time	97.3	272.7	59.5	864.1	856.8	1345.9	1671.9	511.6 (188.5)

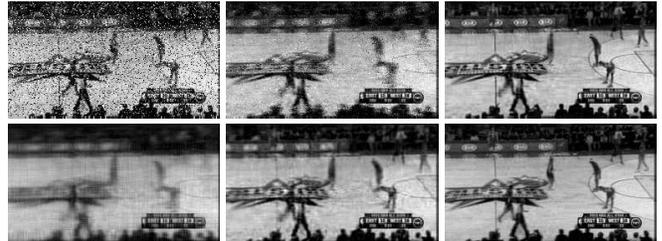


Fig. 3. Recovery comparison on basketball video (20th frame shown). Upper left: Noisy video (CR = 25%). Upper middle: PSSV [60] recovery. Upper right: TRPCA [30] recovery. Lower left: OR-TPCA [35] recovery. Lower middle: RKCA [21] recovery. Lower right: RTPCA-RE recovery.

original paper while it is not capable for higher order tensors. RKCA obtains better results than these three methods although it performs much worse than ours because it breaks the tensor structure and lost information resulting from the unfolding of the data. Furthermore, TRPCA and OR-TPCA achieve limited results, and are even worse than that of RTPCA-RE without the refinement scheme (i.e., RE step in RTPCA-RE), although TRPCA is the best performing existing method. This supports our assumption that current *t*-SVD-based robust tensor PCA methods do not utilize true tubal-rank information, resulting in degraded recovery results.

We illustrate the recovery results by showing the 20th frame of the video in Fig. 3 (for simplicity, we do not show the results of three worse methods: 1) RPCA; 2) SNN; and 3) IRTPCA).⁹

G. Application to Robust Completion for Images With Both Missing and Corrupted Entries

To evaluate RTC-RE in the presence of both missing data and sparse noise, we test on 100 images from the Berkeley

⁹We have also evaluated these seven RPCA methods on Berkeley images with CR = {10%, 30%}, while the recovery performance of these competing approaches is much worse than RTPCA-RE. Due to space limitation, we omit the image denoising results and show the robust completion of Berkeley images in Table VI and Fig. 4.

TABLE VI
ROBUST COMPLETION FOR EXAMPLE IMAGES FROM THE BERKELEY

Image Index (<i>True R</i>)	BRTF	GWLRTF-Tucker	LRTCR-TNN	RTC-RE (RE)	
	[6]	[24]	[34]		Est. <i>R</i>
1 (22)	RSE 1.84E-01 Time 470.5	6.25E-02 1829.9	<u>2.35E-02</u> 271.7	8.73E-10 (4.31E-02)	22 168.9 (50.0)
2 (35)	RSE 3.53E-01 Time 1051.8	1.37E-01 1389.7	<u>5.10E-02</u> 271.6	5.94E-08 (6.54E-02)	35 181.7 (54.3)
3 (36)	RSE 2.54E-01 Time 1251.3	7.49E-02 1778.3	<u>5.97E-02</u> 271.1	1.00E-07 (6.64E-2)	36 182.9 (54.0)
4 (39)	RSE 5.56E-01 Time 1301.5	1.47E-01 1818.9	<u>1.37E-01</u> 193.7	1.32E-07 (1.50E-01)	39 184.5 (56.9)
5 (41)	RSE 1.76E-01 Time 1460.7	2.11E-01 1749.6	<u>5.17E-02</u> 193.3	5.11E-07 (6.14E-02)	41 184.0 (56.2)
6 (47)	RSE 4.45E-01 Time 2217.4	1.49E-01 1773.6	<u>1.43E-01</u> 274.3	2.01E-4 (1.58E-01)	47 183.6 (56.6)
Average of 100 images	RSE 3.31E-01 Time 1128.5	9.65E-02 1696.4	<u>5.11E-02</u> 253.2	3.28E-6 (5.88E-02)	N/A 182.4 (51.3) N/A

Segmentation Dataset [66]. The sizes of these images are $481 \times 321 \times 3$. We truncate them into the exact low-tubal rank images for RE evaluation, where the ranks range from 12 to 53. The second column of Fig. 4 shows the observed images, which are generated by randomly selecting 40% of the corresponding image pixels as missing and randomly setting $CR = 10\%$ of pixels to random values in $[0, 255]$.

We report the robust recovery results of six examples in Table VI and illustrate the corresponding figures in Fig. 4. As Table VI shows, RTC-RE outperforms all of the competing methods with significant improvements in all images and successfully determines the tubal-ranks. This conclusion is also consistent for all the 100 tested images. With these correct ranks, RTC-RE refines the model and improves the recovery, which demonstrates the importance of accurate RE and also confirms the effectiveness of the refinement scheme. BRTF fails to recover these color images because it requires accurate CP-RE while it under-estimates their tensor ranks. GWLRTF-Tucker does not achieve good recovery as it needs to unfold the observed tensors, which can destroy the intrinsic structure of tensorial data and lose vital information. Furthermore, LRTCR-TNN is the best performing existing method but its achieved accuracies are much lower than ours because it does not utilize rank information.

H. Application to Video Background Modeling

We now further apply our methods to the real-world application of surveillance videos with the aim of subtracting the foreground objects from the background (i.e., background modeling problems). As video background is highly correlated along the frames, they can be represented by a low-rank tensor, while the foreground objects are moving along frames, and thus can be represented by a sparse component [6]. We conduct experiments on two popular video sequences: 1) shopping mall sequences with 100 frames, where each frame size is 256×320 and 2) airport hall sequences with 300 frames, where each frame size is 144×176 . To illustrate the capability of simultaneous robust completion and RPCA of our RTC-RE, we randomly remove 90% pixels from these two

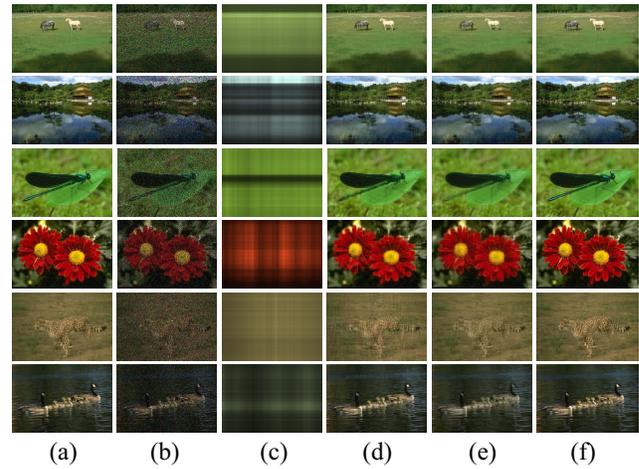


Fig. 4. Recovery performance comparison of six example images corrupted by 40% missing entries and 10% sparse noise (best viewed on screen). (a) Original low tubal-rank images. (b) Observed images. (c)–(e) Recovered images by BRTF, GWLRTF-Tucker, LRTCR-TNN, and RTC-RE, respectively.

videos and evaluate the performance of background modeling under this missing data scenario, which may occur in reality (e.g., the camera is broken and obtains incomplete surveillance videos). We further study two cases: one is the common case, in which we are given all frames of videos, and in the other, we are given partial frames of videos (assume that we could not obtain enough video sequences, or partial frames might be missing).

Fig. 5(a) shows that: given all of the frames from the Airport Hall and Shopping Mall videos, our method is superior to the compared methods. Although BRTF can obtain slightly clearer results than ours, it loses partial RGB color. That is because BRTF must reshape the input to $[256 \times 320, 3, 100]$ format, for example, it fails to work if it retains the original size of videos as the input. This reshaping limits the estimated rank to up to three, and results in the compression (lost partial) of RGB information, particularly with fewer frames of videos, as shown in Fig. 5(b) and 5(c). With fewer frames from the original videos, the ghost effects are more severe with other methods, particularly BRTF, which fails to work on the Shopping Mall video with only 25 frames [1/4 of total frames, shown in Fig. 5(c)], while RTC-RE can still obtain relatively good results. This indicates that RTC-RE has better robustness to missing data, and has less (missing) frames and gross corruptions.

I. Summary of Experimental Results

- 1) The proposed four methods outperform all competing methods with significant improvements in all cases of RE, TC, and robust tensor recovery in various applications and settings on synthetic and real-world tensors. With correct tubal-ranks determined by RE_{TNN} , TC-RE can achieve optimal solutions via the relaxing strategy. RTPCA-RE and RTC-RE not only inherit the ability of RE_{TNN} to achieve accurate RE but also share a similar spirit with TC-RE by explicitly utilizing estimated tubal-ranks to further refine their models (the

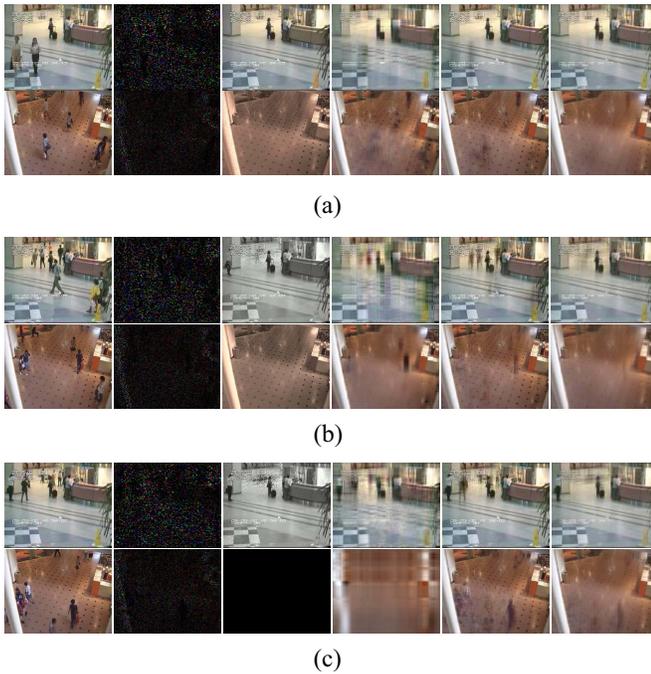


Fig. 5. Background modeling when 90% pixels of the Airport Hall and Shopping Mall videos are missing. The {200th, 100th}, {100th, 50th}, {50th, 25th} frames of results are shown as examples from the first to sixth rows, respectively. (a) Given all the frames of the videos as training. (b) Given only 1/2 of the total frames of the videos. (c) Given only 1/4 of the total frames of the videos. The first and second column figures are the original and the observed video frames (SR = 10%), respectively. The third column to the sixth figures are obtained by BRTF, GWLRTF-Tucker, LRTC-TNN, and RTC-RE, respectively, (best viewed on screen).

refinement schemes) and, thus, obtain successful results. These results confirm the importance of RE and verify the effectiveness and superiority of our methods.

- 2) The compared CP-based (e.g., TNCP and BRTF) and Tucker-based methods (e.g., HaLRTC and GWLRTF-Tucker) do not achieve as good results as the t-SVD models (e.g., TNN, TRPCA, and our methods) in most cases, which verifies our predictions: T-SVD models have more advantages than CP and Tucker models. CP-based methods rely on accurate RE while existing techniques often underestimate or overestimate the ranks, and the Tucker-based methods require unfolding of the observed tensors, which destroys the intrinsic tensor structure and loses vital information. These drawbacks lead to more degraded recovery performances of the CP-/Tucker-based methods than t-SVD-based methods.
- 3) Current t-SVD-based methods (e.g., TNN, t-TNN, TRPCA, and OR-TPCA) obtain a few good results but have much worse performance than ours. With the correct estimated ranks by RE_{TNN}, TCTF-RE consistently outperforms TCTF. These results confirm the advantage of t-SVD models with accurate RE and verify our claim that existing t-SVD-based methods *neither* make use of the low-rank prior (e.g., TNN and t-TNN) *nor* provide accurate RE (e.g., TCTF), resulting in limited recovery results.

- 4) The proposed RE techniques (e.g., RE_{TNN}) are the fastest on the whole among all compared tensor methods including the compared t-SVD-based methods. ARD_{TNN} avoids computing the full t-SVD and thus costs less CPU time. TC-ARD and RTPCA-RE/RTC-RE are not as fast as ARD_{TNN}/RE steps because they involve the computation of the relaxing strategy and refinement schemes with more iterations, respectively. Nevertheless, they are not the slowest and consistently achieve much better recovery results than other methods with less computational costs (in most cases). Although our implementations are not optimized for efficiency as our focus is on the accuracy, we could speed up our implementation by, for example, updating all $\{\hat{\mathbf{X}}^{(i)}\}_{i=1}^3$ in parallel, to achieve better efficiency in future work.

VII. CONCLUSION

In this article, we have proposed four t-SVD-based methods: 1) RE_{TNN}; 2) TC-RE; 3) RTPCA-RE; and 4) RTC-RE. RE_{TNN} has solved the challenging RE problem by simultaneously minimizing the reconstruction error and the TNN of the f -diagonal tensor of an incomplete tensor, where the TNN of the f -diagonal is equivalent to that of the entire tensor and can be further recast as the ℓ_1 -norm of singular values in the Fourier domain. With accurate RE, TC-RE can obtain optimal completion via the relaxing strategy. RTPCA-RE and RTC-RE have solved the robust tensor PCA and robust TC problems, respectively. By explicitly using the correct estimated rank to further refine the models, RTPCA-RE and RTC-RE can achieve a successful recovery. We have also discussed the generalization of the proposed methods and analyzed some variants for other problems. As demonstrated in the experimental results on synthetic and real-world images/videos in various applications, the proposed methods not only correctly determine the true ranks but also successfully recover the tensors with missing entries and/or sparse noise under mild conditions and, thus, outperform the competing methods in all cases with significant improvements. Nevertheless, the proposed methods cannot keep such significant tensor recovery performance on original data without rank truncation as there is no “correct rank” to estimate, which is worth studying elsewhere as our future work.

REFERENCES

- [1] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
- [2] X. Cao, X. Wei, Y. Han, and D. Lin, “Robust face clustering via tensor decomposition,” *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2546–2557, Nov. 2015.
- [3] 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.
- [4] L. Zhang, L. Song, B. Du, and Y. Zhang, “Nonlocal low-rank tensor completion for visual data,” *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 673–685, Feb. 2021.
- [5] A. J. Tom and S. N. George, “A three-way optimization technique for noise robust moving object detection using tensor low-rank approximation, $l_{1/2}$, and TTV regularizations,” *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 1004–1014, Feb. 2021.

- [6] 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.
- [7] Y. Chen, W. He, N. Yokoya, and T.-Z. Huang, "Hyperspectral image restoration using weighted group sparsity-regularized low-rank tensor decomposition," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3556–3570, Aug. 2020.
- [8] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis," in *Proc. UCLA Phonet.*, 1970, pp. 1–84.
- [9] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart–Young decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.
- [10] L. R. Tucker, "Implications of factor analysis of three-way matrices for measurement of change," *Probl. Meas. Change*, vol. 15, pp. 122–137, May 1963.
- [11] E. Hazan, R. Livni, and Y. Mansour, "Classification with low rank and missing data," in *Proc. ICML*, 2015, pp. 257–266.
- [12] G. Doquire and M. Verleysen, "Feature selection with missing data using mutual information estimators," *Neurocomputing*, vol. 90, pp. 3–11, Aug. 2012.
- [13] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *J. ACM*, vol. 60, no. 6, p. 45, 2013.
- [14] P. Rai, Y. Wang, S. Guo, G. Chen, D. Dunson, and L. Carin, "Scalable Bayesian low-rank decomposition of incomplete multiway tensors," in *Proc. ICML*, 2014, pp. 1800–1808.
- [15] C. Hawkins and Z. Zhang, "Bayesian tensorized neural networks with automatic rank selection," 2019. [Online]. Available: arXiv:1905.10478
- [16] L. Cheng, X. Tong, S. Wang, Y.-C. Wu, and H. V. Poor, "Learning nonnegative factors from tensor data: Probabilistic modeling and inference algorithm," *IEEE Trans. Signal Process.*, vol. 68, pp. 1792–1806, Feb. 2020.
- [17] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low- n -rank tensor recovery via convex optimization," *Inverse Probl.*, vol. 27, no. 2, 2011, Art. no. 025010.
- [18] Z. Fang, X. Yang, L. Han, and X. Liu, "A sequentially truncated higher order singular value decomposition-based algorithm for tensor completion," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1956–1967, May 2019.
- [19] T.-X. Jiang, M. K. Ng, X.-L. Zhao, and T.-Z. Huang, "Framelet representation of tensor nuclear norm for third-order tensor completion," *IEEE Trans. Image Process.*, vol. 29, pp. 7233–7244, 2020.
- [20] P. Zhou, C. Lu, Z. Lin, and C. Zhang, "Tensor factorization for low-rank tensor completion," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1152–1163, Mar. 2018.
- [21] M. Bahri, Y. Panagakis, and S. Zafeiriou, "Robust Kronecker component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2365–2379, Oct. 2019.
- [22] Y. Zhou, H. Lu, and Y.-M. Cheung, "Probabilistic rank-one tensor analysis with concurrent regularizations," *IEEE Trans. Cybern.*, early access, May 20, 2019, doi: [10.1109/TCYB.2019.2914316](https://doi.org/10.1109/TCYB.2019.2914316).
- [23] 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, 2014.
- [24] X. Chen *et al.*, "A generalized model for robust tensor factorization with noise modeling by mixture of Gaussians," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5380–5393, Nov. 2018.
- [25] 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.
- [26] Y. Miao, L. Qi, and Y. Wei, "Generalized tensor function via the tensor singular value decomposition based on the t -product," *Linear Algebra Appl.*, vol. 590, pp. 258–303, Apr. 2020.
- [27] L. Feng, Y. Liu, L. Chen, X. Zhang, and C. Zhu, "Robust block tensor principal component analysis," *Signal Process.*, vol. 166, Jan. 2020, Art. no. 107271.
- [28] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-SVD," in *Proc. CVPR*, 2014, pp. 3842–3849.
- [29] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2017.
- [30] 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.
- [31] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller, "Tensor-based formulation and nuclear norm regularization for multienergy computed tomography," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1678–1693, Apr. 2014.
- [32] W. Hu, D. Tao, W. Zhang, Y. Xie, and Y. Yang, "The twist tensor nuclear norm for video completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 2961–2973, Dec. 2017.
- [33] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization," in *Proc. CVPR*, 2016, pp. 5249–5257.
- [34] C. Lu, J. Feng, S. Yan, and Z. Lin, "A unified alternating direction method of multipliers by majorization minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 527–541, Mar. 2018.
- [35] P. Zhou and J. Feng, "Outlier-robust tensor PCA," in *Proc. IEEE CVPR*, 2017, pp. 3938–3946.
- [36] 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.
- [37] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.
- [38] L. Mirsky, "Symmetric gauge functions and unitarily invariant norms," *Quart. J. Math.*, vol. 11, no. 1, pp. 50–59, 1960.
- [39] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust video denoising using low rank matrix completion," in *Proc. IEEE CVPR*, 2010, pp. 1791–1798.
- [40] M. Bahri, Y. Panagakis, and S. Zafeiriou, "Robust kronecker-decomposable component analysis for low-rank modeling," in *Proc. IEEE CVPR*, 2017, pp. 3352–3361.
- [41] B. Huang, C. Mu, D. Goldfarb, and J. Wright, "Provable models for robust low-rank tensor completion," *Pac. J. Optim.*, vol. 11, no. 2, pp. 339–364, 2015.
- [42] X.-L. Zhao, W.-H. Xu, T.-X. Jiang, Y. Wang, and M. K. Ng, "Deep plug-and-play prior for low-rank tensor completion," *Neurocomputing*, vol. 400, pp. 137–149, Aug. 2020, doi: [10.1016/j.neucom.2020.03.018](https://doi.org/10.1016/j.neucom.2020.03.018).
- [43] J. W. Zheng, P. Yang, X. Yang, and S. Y. Chen, "Truncated low-rank and total p variation constrained color image completion and its moreau approximation algorithm," *IEEE Trans. Image Process.*, vol. 29, pp. 7861–7874, Jul. 2020, doi: [10.1109/TIP.2020.3008367](https://doi.org/10.1109/TIP.2020.3008367).
- [44] Z. Long, C. Zhu, J. Liu, and Y. Liu, "Bayesian low rank tensor ring model for image completion," 2020. [Online]. Available: arXiv:2007.01055
- [45] Y. Liu, L. Jiao, and F. Shang, "A fast tri-factorization method for low-rank matrix recovery and completion," *Pattern Recognit.*, vol. 46, no. 1, pp. 163–173, 2013.
- [46] G. Liu and S. Yan, "Active subspace: Toward scalable low-rank learning," *Neural Comput.*, vol. 24, no. 12, pp. 3371–3394, 2012.
- [47] W. Fang, D. Wei, and R. Zhang, "Stable tensor principal component pursuit: Error bounds and efficient algorithms," *Sensors*, vol. 19, no. 23, p. 5335, 2019.
- [48] Z. Zha, X. Yuan, B. Wen, J. Zhou, J. Zhang, and C. Zhu, "A benchmark for sparse coding: When group sparsity meets rank minimization," *IEEE Trans. Image Process.*, vol. 29, pp. 5094–5109, Mar. 2020, doi: [10.1109/TIP.2020.2972109](https://doi.org/10.1109/TIP.2020.2972109).
- [49] Q. Shi, H. Lu, and Y.-M. Cheung, "Rank-one matrix completion with automatic rank estimation via l_1 -norm regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4744–4757, Oct. 2018.
- [50] Z. Zha *et al.*, "Analyzing the group sparsity based on the rank minimization methods," in *Proc. IEEE ICME*, 2017, pp. 883–888.
- [51] S. Osher, Y. Mao, B. Dong, and W. Yin, "Fast linearized Bregman iteration for compressive sensing and sparse denoising," 2011. [Online]. Available: arXiv:1104.0262
- [52] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, p. 11, 2011.
- [53] H. Fan, Y. Chen, Y. Guo, H. Zhang, and G. Kuang, "Hyperspectral image restoration using low-rank tensor recovery," *Sci. Technol. Arts Res. J.*, vol. 10, no. 10, pp. 4589–4604, 2017.
- [54] Q. Shi, Y.-M. Cheung, Q. Zhao, and H. Lu, "Feature extraction for incomplete data via low-rank tensor decomposition with feature regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1803–1817, Jun. 2018.
- [55] Y. Liu, F. Shang, W. Fan, J. Cheng, and H. Cheng, "Generalized higher order orthogonal iteration for tensor learning and decomposition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2551–2563, Dec. 2016.
- [56] Y. Liu, F. Shang, L. Jiao, J. Cheng, and H. Cheng, "Trace norm regularized candecomp/parafac decomposition with missing data," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2437–2448, Nov. 2015.

- [57] Y. Liu, Z. Long, and C. Zhu, "Image completion using low tensor tree rank and total variation minimization," *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 338–350, Feb. 2019.
- [58] H. Huang, Y. Liu, J. Liu, and C. Zhu, "Provable tensor ring completion," *Signal Process.*, vol. 171, Jun. 2020, Art. no. 107486.
- [59] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," 2010. [Online]. Available: arXiv:1009.5055
- [60] T.-H. Oh, Y.-W. Tai, J.-C. Bazin, H. Kim, and I. S. Kweon, "Partial sum minimization of singular values in robust PCA: Algorithm and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 744–758, Apr. 2016.
- [61] Y. Liu, L. Chen, and C. Zhu, "Improved robust tensor principal component analysis via low-rank core matrix," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1378–1389, Dec. 2018.
- [62] X. Lu, T. Gong, P. Yan, Y. Yuan, and X. Li, "Robust alternative minimization for matrix completion," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 939–949, Jun. 2012.
- [63] S. Zhang, L. Jiao, F. Liu, and S. Wang, "Global low-rank image restoration with Gaussian mixture model," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1827–1838, Jun. 2017.
- [64] M. F. Kaloorazi and J. Chen, "Low-rank matrix approximation based on intermingled randomized decomposition," in *Proc. IEEE ICASSP*, 2019, pp. 7475–7479.
- [65] G. Singh, S. Gupta, M. Lease, and C. Dawson, "Streaming singular value decomposition for big data applications," 2020. [Online]. Available: arXiv:2010.14226
- [66] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. ICCV*, vol. 2, 2001, pp. 416–423.