Robust Low-Rank Tensor Completion Based on Tensor Ring Rank via $\ell_{p,\epsilon}$-Norm

Xiao Peng Li and Hing Cheung So, *Fellow, IEEE*

***Abstract*—Tensor completion aims to recover missing entries given incomplete multi-dimensional data by making use of the prior low-rank information, and has various applications because many real-world data can be modeled as low-rank tensors. Most of the existing methods are designed for noiseless or Gaussian noise scenarios, and thus they are not robust to outliers. One popular approach to resist outliers is to employ ℓ_p-norm. Yet nonsmoothness and nonconvexity of ℓ_p-norm with $0 < p \leq 1$ bring challenges to optimization. In this paper, a new norm, named $\ell_{p,\epsilon}$-norm, is devised where $\epsilon > 0$ can adjust the convexity of $\ell_{p,\epsilon}$-norm. Compared with ℓ_p-norm, $\ell_{p,\epsilon}$-norm is smooth and convex even for $0 < p \leq 1$, which converts an intractable nonsmooth and nonconvex optimization problem into a much simpler convex and smooth one. Then, combining tensor ring rank and $\ell_{p,\epsilon}$-norm, a robust tensor completion formulation is proposed, which achieves outstanding robustness. The resultant robust tensor completion problem is decomposed into a number of robust linear regression (RLR) subproblems, and two algorithms are devised to tackle RLR. The first method adopts gradient descent, which has a low computational complexity. While the second one employs alternating direction method of multipliers to yield a fast convergence rate. Numerical simulations show that the two proposed methods have better performance than those based on the ℓ_p-norm in RLR. Experimental results from applications of image inpainting, video restoration and target estimation demonstrate that our robust tensor completion approach outperforms state-of-the-art methods in terms of recovery accuracy.**

***Index Terms*—Tensor completion, tensor ring rank, linear regression, outlier, robust recovery, gradient descent, alternating direction method of multipliers.**

I. Introduction

MISSING value recovery from incomplete data has been a significant research topic, which includes reconstructing one-dimensional, two-dimensional and multi-dimensional data. For one-dimensional data, it refers to compressed sensing which utilizes sparsity to recover a sparse vector from a few measured entries [1], [2]. From vectors to matrices, it becomes matrix recovery problem which aims to restore a matrix from an incomplete matrix with low-rank characteristic [3], [4]. While tensor completion refers to recovering a low-rank tensor from an incomplete tensor. It is considered to be a powerful technique to process multi-dimensional data, such as color images and videos, as it can excavate more latent correlations. For example, compared with matrix completion in processing a gray-scale image [5], tensor completion can use RGB channel information to obtain a better recovered image. Hence, tensor completion has become the core problem in many applications, including visual data recovery [6], [7], multiple-input multiple-output (MIMO) radar [8], [9] and machine learning [10], [11].

Unlike matrix rank which has a unique definition, the tensor rank has several definitions, namely n-rank [12], CANDECOMP/PARAFAC (CP) rank [13], [14], Tucker rank [15], tensor tubal rank [16], tensor train (TT) rank [17] and tensor ring (TR) rank [18]. The performance of many existing tensor completion methods is, to a large extent, affected by their own adopted tensor ranks. For tensor completion methods based on n-rank, an nth-order tensor is unfolded into n matrices in n ways and then the sum of the ranks of these n matrices is minimized [19], [20]. Since this approach just unfolds a tensor into matrices without excavating the tensor structure, an accurate solution may not be attained [21]. CP rank can make use of the tensor structure, which aims to find a linear combination of rank-one tensors [22], [23]. Yet it cannot directly process color image data and video data in which one dimension is much smaller than the other dimensions. Because the low-rank attribute cannot be satisfied in the minimum dimension, the performance will degrade when recovering color images and videos. Otherwise, it may not find a reasonable result since the best low-rank approximation based on CP rank is ill-posed [24]. Tucker rank is defined as a set of all ranks of the matrices factorized by Tucker decomposition [25], [26], but it is not applicable for high-dimensional tensors because its storage memory grows exponentially with the data dimension. Tensor tubal rank is closest to matrix rank, which is calculated from the diagonal tensor factorized by tensor singular value decomposition [27]. It has been shown that tensor tubal rank is more suitable than Tucker rank [28]–[30] for low-rank tensor completion. TT rank utilizes two matrices and a few 3rd-order tensors to represent the target tensor. It has been reported that TT rank can exploit more correlations and obtain better performance than CP rank and Tucker rank in tensor completion [31], [32]. Nevertheless, TT rank requires that factors on both sides are matrices, which results in large intermediate factors and small boundary factors. This problem is solved by TR rank in which two boundary factors can be 3rd-order tensors. Moreover, it has

Manuscript received July 29, 2020; revised February 26, 2021 and April 27, 2021; accepted May 25, 2021. Date of publication May 31, 2021; date of current version July 2, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jarvis Haupt. *(Corresponding author: Hing Cheung So.)*

The authors are with the Department of Electrical Engineering, City University of Hong Kong, Kowloon Tong, Hong Kong (e-mail: x.p.li.frank@gmail.com; hcso@ee.cityu.edu.hk).

Digital Object Identifier 10.1109/TSP.2021.3085116

been shown that the TR rank outperforms the TT rank in tensor completion [34].

On the other hand, conventional tensor completion methods employ Frobenius norm, indicating that they are designed for the Gaussian noise and noise-free scene. Data, in practice, may contain outliers, including impulsive noise in communication channels and salt-and-pepper noise in images [35], [36]. Because ℓ_2-space optimization, which is derived based on Gaussian noise assumption, is unable to resist outliers, the performance of these methods will be degraded when the observed entries are corrupted by outliers. To resist outliers, ℓ_p-norm with $0 < p < 2$ has been adopted, such as ℓ_p-parallel factor analysis (ℓ_p-PARAFAC) [37] and iteratively reweighted tensor singular value decomposition (IR-t-SVD) [38]. Since ℓ_p-PARAFAC employs CP rank, its performance is affected by the tensor model. It is worth noting that the noise in IR-t-SVD must obey the random tubal distribution which means that noise in different frontal slices has the same coordinate. Moreover, ℓ_p-norm is nonsmooth and nonconvex with $0 < p \leq 1$, leading to complicated optimization and possibly local solutions.

In this paper, $\ell_{p,\epsilon}$-norm is devised to replace ℓ_p-norm for resisting outliers. Then we employ TR rank and $\ell_{p,\epsilon}$-norm to reformulate the robust tensor completion problem. The reason for adopting TR rank is that it has better stability and adaptability than other tensor ranks. Besides, block coordinate descent (BCD) is applied to solve the resultant multi-variable optimization problem in which one tensor variable is updated at each iteration while the remaining variables are fixed. In addition, two algorithms are developed for solving RLR. The first one utilizes gradient descent (GD) to yield a low computational complexity. While the second one uses alternating direction method of multipliers (ADMM), which has a fast convergence rate. Yet only the ADMM-based method is adopted to tackle robust tensor completion since the appropriate value of the step-size of the GD-based approach is difficult to obtain. Our main contributions are summarized as:

1) We devise $\ell_{p,\epsilon}$-norm to achieve robustness. Compared with ℓ_p-norm, $\ell_{p,\epsilon}$-norm is smooth and convex with $0 < p \leq 1$.
2) The convexity of $\ell_{p,\epsilon}$-norm with $0 < p < 1$ is proved. Moreover, we clearly describe how to determine ϵ which gives convex $\ell_{p,\epsilon}$-norm. We also analyze that $\ell_{p,\epsilon}$-norm has better performance in resisting outliers than ℓ_p-norm with $0 < p \leq 1$.
3) Based on $\ell_{p,\epsilon}$-norm, we employ TR rank for robust tensor completion. The proposed approach exhibits better performance than existing methods in processing different data including color image, color video, and MIMO radar data in the presence of outliers.

The remainder of this paper is organized as follows. In Section II, notations and preliminaries are provided, and the TR decomposition is reviewed. The $\ell_{p,\epsilon}$-norm is derived, and its associated properties are analyzed in Section III. In Section IV, the robust tensor completion problem based on TR rank and $\ell_{p,\epsilon}$-norm is formulated, and then we decompose the problem into a number of $\ell_{p,\epsilon}$-norm-based RLR subproblems. Two algorithms, namely, GD-$\ell_{p,\epsilon}$-norm and ADMM-$\ell_{p,\epsilon}$-norm, are developed in Section V. In Section VI, we first compare the two proposed algorithms with existing methods based on ℓ_p-norm to deal with RLR. Moreover, numerical examples based on real-world and synthetic data including color image, video and MIMO radar data demonstrate that our robust tensor completion method outperforms the state-of-the-art approaches. Finally, concluding remarks are included in Section VII.

II. Notations and Preliminaries

In this section, notations, basic definitions and TR decomposition are reviewed.

A. Notations

Scalars, vectors, matrices and tensors are represented by italic, bold lower-case, bold upper-case and bold calligraphic letters, respectively. For instance, $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$ denotes an nth-order tensor and its $(i_1, i_2, \ldots, i_n)$ entry is denoted as $\boldsymbol{\mathcal{A}}(i_1, i_2, \ldots, i_n)$. The $\odot$ represents the Hadamard product operation. Pseudo-inverse operator, transpose operator and conjugate transpose operator are denoted by $(\cdot)^\dagger$, $(\cdot)^T$ and $(\cdot)^H$, respectively. Besides, $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2}$ is the Frobenius norm of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ where a_{ij} is the (i, j) entry of $\boldsymbol{A}$. The trace of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is represented by $\mathrm{tr}(\boldsymbol{A}) = \sum_{i=1}^n a_{ii}$. For vectors, matrices and tensors, $\| \cdot \|_p^p$ with $0 < p < 2$ corresponds to the ℓ_p-norm which is calculated by the sum of p power of all elements. Moreover, $\| \cdot \|_2$, $\| \cdot \|_0$ and $\dim(\cdot)$ stand for the ℓ_2-norm, ℓ_0-norm and dimension of a vector, respectively. Given $\boldsymbol{a}$ and $\boldsymbol{b}$, $\boldsymbol{a} \circ \boldsymbol{b}$ denotes the outer product, and $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ represents the inner product. The vectorization of $\boldsymbol{\mathcal{A}}$ is described by $\mathrm{vec}(\boldsymbol{\mathcal{A}})$, while $\mathrm{res}(\cdot)$ is the reshape operator which returns an array of specified dimensions with the same entries as the input data. For a 3rd-order tensor, $\boldsymbol{\mathcal{A}}(:, i_2, i_3)$, $\boldsymbol{\mathcal{A}}(i_1, :, i_3)$ and $\boldsymbol{\mathcal{A}}(i_1, i_2, :)$ represent horizontal fiber, vertical fiber and depth fiber, respectively. In addition, its horizontal slice, lateral slice and frontal slice are denoted as $\boldsymbol{\mathcal{A}}(i_1, :, :)$, $\boldsymbol{\mathcal{A}}(:, i_2, :)$ and $\boldsymbol{\mathcal{A}}(:, :, i_3)$, respectively.

B. Basic Operations

Definition 1 (Mode-i unfolding [39]): Mode-i unfolding of $\boldsymbol{\mathcal{A}}$ is denoted as $\boldsymbol{\mathcal{A}}_{[i]}$, which converts an nth-order $\boldsymbol{\mathcal{A}}$ into a matrix:

$$\boldsymbol{\mathcal{A}}_{[i]} \in \mathbb{R}^{I_i \times (I_1 \cdots I_{i-1} I_{i+1} \cdots I_n)}. \tag{1}$$

Definition 2 (Left unfolding and right unfolding [40]): For $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, left unfolding is denoted as:

$$L(\boldsymbol{\mathcal{A}}) = (\boldsymbol{\mathcal{A}}_{[3]})^T \in \mathbb{R}^{(I_1 I_2) \times I_3}. \tag{2}$$

Similar to left unfolding, right unfolding is:

$$R(\boldsymbol{\mathcal{A}}) = \boldsymbol{\mathcal{A}}_{[1]} \in \mathbb{R}^{I_1 \times (I_2 I_3)}. \tag{3}$$

Definition 3 (Tensor connect product [34]): Given $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{R_0 \times I_1 \times R_1}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{R_1 \times I_2 \times R_2}$, then the tensor connect product between these two tensors is defined as:

$$\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{B}} = \mathrm{res}(L(\boldsymbol{\mathcal{A}}) \times R(\boldsymbol{\mathcal{B}})) \in \mathbb{R}^{R_0 \times (I_1 I_2) \times R_2}. \tag{4}$$

Tensor connect product which is similar to matrix product provides the product rule for two 3rd-order tensors. Note that

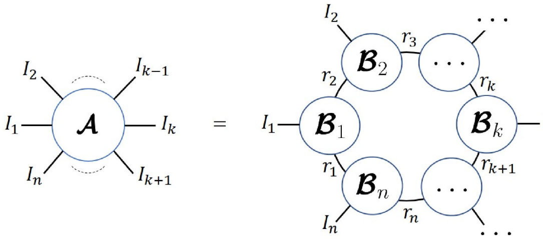


Fig. 1. TR decomposition [18].

the last dimension of the former tensor must be equal to the first dimension of the latter.

Definition 4 (Tensor permutation [18]): For $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$, the ith tensor permutation is denoted as:

$$\boldsymbol{\mathcal{A}}^{p_i} \in \mathbb{R}^{I_i \times I_{i+1} \times \cdots \times I_n \times I_1 \times I_2 \times \cdots I_{i-1}}. \tag{5}$$

The relationship between $\boldsymbol{\mathcal{A}}^{p_i}$ and $\boldsymbol{\mathcal{A}}$ obeys:

$$\boldsymbol{\mathcal{A}}^{p_i}(i_i, \ldots, i_n, i_1, \cdots i_{i-1}) = \boldsymbol{\mathcal{A}}(i_1, i_2, \ldots, i_n). \tag{6}$$

Definition 5 (Inverse operator of TR decomposition [34]): Given $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{R_0 \times (I_1 I_2 \cdots I_n) \times R_0}$, the inverse operator, denoted as U, is to organize $\boldsymbol{\mathcal{A}}$ as an nth-order tensor:

$$\boldsymbol{\mathcal{B}} = U(\boldsymbol{\mathcal{A}}) \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n} \tag{7}$$

where the entry of $\boldsymbol{\mathcal{B}}$ is calculated by

$$\begin{aligned} &\boldsymbol{\mathcal{B}}(i_1, i_2, \ldots, i_n) \\ &\quad = \operatorname{tr}(\boldsymbol{\mathcal{A}}(:, i_1 + (i_2 - 1)I_1 + \cdots + (i_n - 1)I_{n-1}, :)). \end{aligned} \tag{8}$$

Lemma 1: Given $\boldsymbol{\mathcal{B}} = U(\boldsymbol{\mathcal{A}}_1 \cdots \boldsymbol{\mathcal{A}}_n)$, then

$$\boldsymbol{\mathcal{B}}^{p_i} = U(\boldsymbol{\mathcal{A}}_i \boldsymbol{\mathcal{A}}_{i+1} \cdots \boldsymbol{\mathcal{A}}_n \boldsymbol{\mathcal{A}}_1 \cdots \boldsymbol{\mathcal{A}}_{i-1}). \tag{9}$$

C. TR Decomposition

TR decomposition represents an nth-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$ by using multiple 3rd-order tensors:

$$\begin{aligned} \boldsymbol{\mathcal{A}} = \sum_{\alpha_1, \ldots, \alpha_n = 1}^{r_1, \ldots, r_n} &\boldsymbol{\mathcal{B}}_1(\alpha_1, :, \alpha_2) \circ \boldsymbol{\mathcal{B}}_2(\alpha_2, :, \alpha_3) \circ \cdots \\ &\circ \boldsymbol{\mathcal{B}}_n(\alpha_n, :, \alpha_1) \end{aligned} \tag{10}$$

where $\boldsymbol{\mathcal{B}}_i \in \mathbb{R}^{r_i \times I_i \times r_{i+1}}$, $\boldsymbol{\mathcal{B}}_i(\alpha_i, :, \alpha_{i+1})$ is the vertical fiber and $\boldsymbol{r} = [r_1, \ldots, r_n]$ is called TR rank. It is indicated in (10) that $\boldsymbol{\mathcal{A}}$ can be represented by a sum of rank-one tensors. The entry of $\boldsymbol{\mathcal{A}}$ is computed as:

$$\boldsymbol{\mathcal{A}}(i_1, \ldots, i_n) = \operatorname{tr}\left(\prod_{k=1}^{n} \boldsymbol{\mathcal{B}}_k(:, i_k, :)\right) \tag{11}$$

where $\boldsymbol{\mathcal{B}}_k(:, i_k, :)$ denotes the i_kth lateral slice with dimensions $r_k \times r_{k+1}$. It is worth noting that $r_{n+1} = r_1$. Fig. 1 depicts the TR decomposition of an nth-order tensor. The circle represents a tensor, and the tensor dimension is denoted by the number of edges. The line connecting two circles stands for the tensor connect product. The factorization format looks like a ring, giving rise to the name of TR decomposition.

Note that all r_i with $i \in [1, n]$ can be different in the standard TR decomposition. In our work, $r_1 = r_2 = \cdots = r_n = R$ so the TR rank is R.

III. $\ell_{p,\epsilon}$-NORM

In this section, $\ell_{p,\epsilon}$-norm is proposed, then its basic properties are introduced and analyzed. Subsequently, the difference between $\ell_{p,\epsilon}$-norm and ℓ_p-norm is illustrated.

A. Definition and Properties of $\ell_{p,\epsilon}$-Norm

Definition 6: Given $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$, the $\ell_{p,\epsilon}$-norm of $\boldsymbol{x}$ with $0 < p < 2$ is defined as:

$$\|\boldsymbol{x}\|_{p,\epsilon}^p = \sum_{i=1}^{n} \left[(x_i^2 + \epsilon^2)^{p/2} - \epsilon^p\right] \tag{12}$$

where ϵ enables $\ell_{p,\epsilon}$-norm to be convex for $0 < p < 1$. For $0 < p \le 1$, $\epsilon > 0$. Otherwise, $\epsilon = 0$.

Lemma 2: The $\ell_{p,\epsilon}$-norm has following properties:

(i) Given x_1 and x_2, if $|x_1|^p \le |x_2|^p$, then $\left(x_1^2 + \epsilon^2\right)^{p/2} - \epsilon^p \le \left(x_2^2 + \epsilon^2\right)^{p/2} - \epsilon^p$.

(ii) For $1 < p < 2$, $\ell_{p,\epsilon}$-norm equals ℓ_p-norm.

(iii) For $p = 1$, $\lim_{\epsilon \to 0} \|\boldsymbol{x}\|_{1,\epsilon}^1 = \|\boldsymbol{x}\|_1^1$.

(iv) $\ell_{1,\epsilon}$-norm with $\epsilon > 0$ is both convex and smooth.

(v) For $0 < p \le 1$, $\ell_{p,\epsilon}$-norm is the lower convex envelope of ℓ_p-norm with $\epsilon > \sqrt{1-p}\|\boldsymbol{x}\|_\infty$.

Proof: The properties of (i), (ii), (iii) and (iv) are easily verified. We focus on proving (v).

For (v), we first show that $\ell_{p,\epsilon}$-norm with $\epsilon > \sqrt{1-p}\|\boldsymbol{x}\|_\infty$ is a convex function, and then prove that $\ell_{p,\epsilon}$-norm is the lower bound of ℓ_p-norm. Although we utilize the vector of length 2 to analyze, the proof can be extended to different lengths.

Given a bounded $\boldsymbol{x} = [x_1, x_2]^T \in \mathbb{R}^2$ with $|x_1| \le a$ and $|x_2| \le b$ where $a > 0$ and $b > 0$, the $\ell_{p,\epsilon}$-norm function of $\boldsymbol{x}$ is defined as:

$$\begin{aligned} f(x_1, x_2) &= \|\boldsymbol{x}\|_{p,\epsilon}^p \\ &= (x_1^2 + \epsilon^2)^{p/2} - \epsilon^p + (x_2^2 + \epsilon^2)^{p/2} - \epsilon^p. \end{aligned} \tag{13}$$

The gradient of $f(x_1, x_2)$ is:

$$\begin{aligned} \nabla f(x_1, x_2) &= \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right]^T \\ &= [p x_1 (x_1^2 + \epsilon^2)^{p/2-1}, p x_2 (x_2^2 + \epsilon^2)^{p/2-1}]^T \end{aligned} \tag{14}$$

and the Hessian matrix of $f(x_1, x_2)$ is:

$$\begin{aligned} G(x_1, x_2) &= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} \\ &= \begin{bmatrix} T(x_1) & 0 \\ 0 & T(x_2) \end{bmatrix} \begin{bmatrix} P(x_1) & 0 \\ 0 & P(x_2) \end{bmatrix} \end{aligned} \tag{15}$$

where $T(x_i) = p(x_i^2 + \epsilon^2)^{p/2-2}$ and $P(x_i) = (p-1)x_i^2 + \epsilon^2$. It is well known that the convexity condition of a multi-variable $f(x_1, x_2)$ is the Hessian matrix being positive definite in the whole feasible region of $\boldsymbol{x}$. For $p = 1$, $G(x_1, x_2)$ is positive definite for positive ϵ. That is to say, $\ell_{1,\epsilon}$-norm is convex with $\epsilon > 0$. For $0 < p < 1$, to make $G(x_1, x_2)$ positive definite, the terms of both $P(x_1)$ and $P(x_2)$ must be greater than zero for all values of $\boldsymbol{x}$. Mathematically, ϵ satisfies:

$$\begin{cases} (p-1)x_1^2 + \epsilon^2 > 0 \\ (p-1)x_2^2 + \epsilon^2 > 0 \end{cases} \tag{16}$$

which leads to $\epsilon > \sqrt{1-p}\max(a,b) = \sqrt{1-p}\|\boldsymbol{x}\|_\infty$. Hence, the $\ell_{p,\epsilon}$-norm of $\boldsymbol{x}$ is convex with $\epsilon > \sqrt{1-p}\|\boldsymbol{x}\|_\infty$ for $0 < p \leq 1$. Now, we prove that $\ell_{p,\epsilon}$-norm is the lower bound of ℓ_p-norm via constructing:

$$\begin{aligned} g(x_1, x_2) &= \|\boldsymbol{x}\|_{p,\epsilon}^p - \|\boldsymbol{x}\|_p^p \\ &= (x_1^2 + \epsilon^2)^{p/2} - \epsilon^p - |x_1|^p \\ &\quad + (x_2^2 + \epsilon^2)^{p/2} - \epsilon^p - |x_2|^p. \end{aligned} \tag{17}$$

Since $g(x_1, x_2)$ is an even function, we only need to show that $g(x_1, x_2) \leq 0$ for $x_1 \geq 0$ and $x_2 \geq 0$. According to the inequality of $\sqrt{c^2 + d^2} \leq c + d$ for all nonnegative c and d, it can be obtained:

$$\begin{aligned} g(x_1, x_2) &\leq h(x_1, x_2) \\ &= (x_1 + \epsilon)^p - \epsilon^p - x_1^p + (x_2 + \epsilon)^p - \epsilon^p - x_2^p. \end{aligned} \tag{18}$$

For $p = 1$, it is easy to get:

$$h(x_1, x_2) = (x_1 + \epsilon) - \epsilon - x_1 + (x_2 + \epsilon) - \epsilon - x_2 = 0 \tag{19}$$

which means $\|\boldsymbol{x}\|_{1,\epsilon}^1 \leq \|\boldsymbol{x}\|_1^1$ where the equal sign holds if and only if $\boldsymbol{x} = \mathbf{0}$. For $0 < p < 1$, each partial derivative of $h(x_1, x_2)$ is:

$$p\left((x_i + \epsilon)^{p-1} - x_i^{p-1}\right) \tag{20}$$

which is less than zero for $x_i > 0$, implying that $h(x_1, x_2) \leq h(0, 0) = 0$. Combining with $g(x_1, x_2) \leq h(x_1, x_2)$, we have $g(x_1, x_2) \leq 0$, indicating $\|\boldsymbol{x}\|_{p,\epsilon}^p \leq \|\boldsymbol{x}\|_p^p$ where the equal sign is satisfied if and only if $\boldsymbol{x} = \mathbf{0}$. Hence, it has been proved that the $\ell_{p,\epsilon}$-norm is the lower convex envelope of ℓ_p-norm with $\epsilon > \sqrt{1-p}\|\boldsymbol{x}\|_\infty$ for $0 < p \leq 1$.

Lemma 3: The $\ell_{p,\epsilon}$-norm satisfies following properties:

(i) $\ell_{p,\epsilon}$-norm is more robust to outliers than ℓ_p-norm with $0 < p \leq 1$ and $\epsilon > \sqrt{1-p}\|\boldsymbol{x}\|_\infty$.

(ii) The unique minimum of the $\ell_{p,\epsilon}$-norm with $0 < p < 1$ and $\epsilon > \sqrt{1-p}\|\boldsymbol{x}\|_\infty$ is equivalent to the global solution to minimizing ℓ_p-norm.

Proof: Before proving (i), we first introduce the definition of robustness. Robustness refers to the ability to deal with deviations from the distributional assumptions [33], [35], while outliers correspond to the data with large anomaly. Hence, a method which is robust to outliers means that it can achieve small performance loss even when the data largely deviate from the assumed model. Compared with ℓ_2-norm that works perfectly in additive Gaussian noise, ℓ_1-norm has better performance under impulsive noise as ℓ_1-norm does not amplify outliers. To prove (i), we adopt the inequality of $(e^2 + \epsilon^2)^{p/2} - \epsilon^p < |e|^p$ for $0 < p \leq 1$. This means that $\ell_{p,\epsilon}$-norm has a stronger ability than ℓ_p-norm to suppress outliers.

In (v) of Lemma 2, we have analyzed that $\ell_{p,\epsilon}$-norm is the lower bound of ℓ_p-norm. For a residual e related to outlier, it can be obtained that $(e^2 + \epsilon^2)^{p/2} - \epsilon^p < |e|^p$ for $0 < p \leq 1$. Hence, the $\ell_{p,\epsilon}$-norm has better performance than ℓ_p-norm on robustness to outliers with $0 < p \leq 1$ and $\epsilon > \sqrt{1-p}\|\boldsymbol{x}\|_\infty$.

To prove (ii), we first define:

$$f^p(\boldsymbol{x}) = \sum_{i=1}^{n} |x_i|^p \tag{21}$$

where $\boldsymbol{x}$ is a vector with length n. Given a convex set $\mathcal{X}$ of $\boldsymbol{x}$, suppose $\boldsymbol{x}^* \in \mathcal{X}$ is the global solution to $\min_{\boldsymbol{x}} f^p(\boldsymbol{x})$, we obtain:

$$f^p(\boldsymbol{x}^*) < f^p(\boldsymbol{x}) \tag{22}$$

where $\boldsymbol{x} \in \mathcal{X}$ but $\boldsymbol{x} \neq \boldsymbol{x}^*$. Because $\boldsymbol{x}^*$ is the global solution, for x_i with $i \in [1, n]$, we have:

$$f^p(x_1^*, \ldots, x_i^*, \ldots, x_n^*) < f^p(x_1, \ldots, x_i, \ldots, x_n). \tag{23}$$

Since $\partial f^p / \partial x_i > 0$ for $x_i > 0$, and $\partial f^p / \partial x_i < 0$ for $x_i < 0$, (23) leads to $|x_i| > |x_i^*|$. Moreover, according to the Property (i) of Lemma 2, we obtain:

$$f(x_1^*, \ldots, x_i^*, \ldots, x_n^*) < f(x_1, \ldots, x_i, \ldots, x_n). \tag{24}$$

This results in $f(\boldsymbol{x}^*) < f(\boldsymbol{x})$, indicating that $\boldsymbol{x}^*$ is the minimizer of $\min_{\boldsymbol{x}} f(\boldsymbol{x})$. Moreover, since $f(\boldsymbol{x})$ is convex, $\boldsymbol{x}^*$ is the unique minimizer. Therefore, the unique minimizer of $\ell_{p,\epsilon}$-norm is equivalent to the global solution to minimizing ℓ_p-norm.

B. Comparison Between $\ell_{p,\epsilon}$-Norm and ℓ_p-Norm

Fig. 2 shows a comparison illustration between the $\ell_{p,\epsilon}$-norm and ℓ_p-norm with $p = 1$ and $p = 0.5$. The left one indicates that $\ell_{1,\epsilon}$-norm is smooth. Moreover, $\ell_{1,\epsilon}$-norm is the lower bound of ℓ_1-norm. As shown by the right one, there exists great diversity between $\ell_{0.5,\epsilon}$-norm and $\ell_{0.5}$-norm. The difference implies that $\ell_{0.5,\epsilon}$-norm has a better ability than $\ell_{0.5}$-norm to resist outliers. In addition, the $\ell_{0.5,\epsilon}$-norm is convex and has the same minimum as the $\ell_{0.5}$-norm.

C. Existing Alternative Proposals of ℓ_p-Norm

Our $\ell_{p,\epsilon}$-norm is inspired by the prior work [41] which proposes a ℓ_p-approximation:

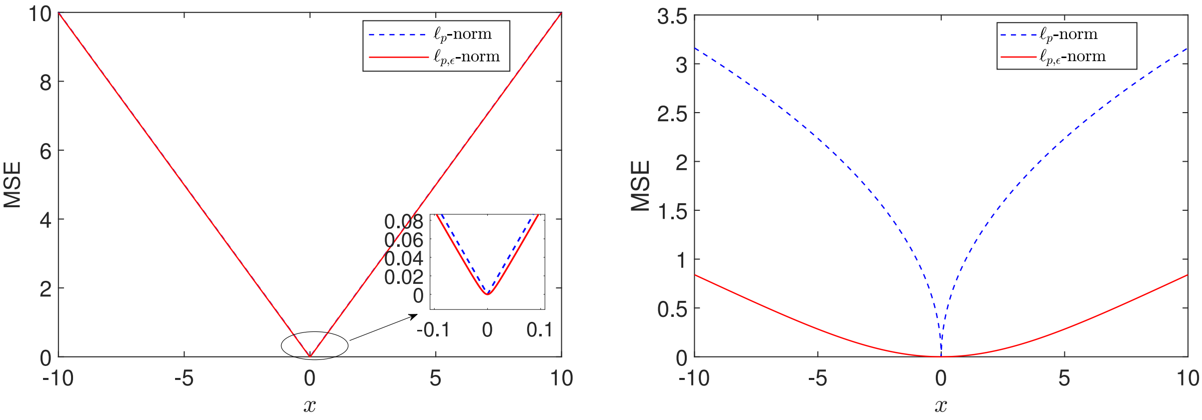
$$u^e(\boldsymbol{x}) = \sum_{i=1}^{n} (x_i^2 + e^2)^{p/2} \tag{25}$$

which is utilized to solve the best linear ℓ_p-approximation of a function on a discrete point set for $p \in (1, 2)$. However, our $\ell_{p,\epsilon}$-norm based on the improvement of the prior ℓ_p-approximation is to solve the problem caused by ℓ_p-norm with $0 < p \leq 1$.

Another method to replace ℓ_p-norm is proposed in [42]:

$$|x|_1^1 \approx x\tanh(\gamma x) \tag{26}$$

where $\gamma > 0$ adjusts the approximate accuracy. However, this method focuses on approximating ℓ_1-norm, which does not solve the nonconvex problem of ℓ_p-norm with $0 < p < 1$.



(a) $p = 1$

(b) $p = 0.5$

Fig. 2. $\ell_{p,\epsilon}$-norm and ℓ_p-norm in one-dimensional case. Left: $\epsilon = \sqrt{1-p} \times \max(x) + 10^{-2} = 10^{-2}$. Right: ϵ is calculated by $\epsilon = \sqrt{1-p} \times \max(x) + 10^{-2} = \sqrt{0.5} \times 10 + 10^{-2} = 7.081$.

IV. $\ell_{p,\epsilon}$-Norm for Robust Tensor Completion

To facilitate the presentation, we first review the low-rank matrix completion problem, which aims to recover a low-rank matrix from a partially observed matrix, formulated as:

$$\begin{aligned}&\min_{\boldsymbol{M}} \text{rank}(\boldsymbol{M})\\&\text{s.t.}\, \boldsymbol{M} \odot \boldsymbol{\Omega} = \boldsymbol{Y}_\Omega\end{aligned} \tag{27}$$

where $\boldsymbol{Y}_\Omega \in \mathbb{R}^{m\times n}$ is an observed matrix with missing entries, $\boldsymbol{\Omega}$ is a binary matrix in which 1 and 0 mean that the corresponding entries in $\boldsymbol{Y}_\Omega$ are available and missing, respectively. Unfortunately, minimizing rank is a NP-hard problem since the rank function is discrete and nonconvex. There are two popular approaches to solve (27), namely nuclear norm minimization [43], [44] and matrix factorization [45], [46]. The matrix factorization scheme can avoid computing singular value decomposition [45], which acquires a lower computational complexity than nuclear norm minimization, corresponding to the following problem:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|(\boldsymbol{U}\boldsymbol{V}) \odot \boldsymbol{\Omega} - \boldsymbol{Y}_\Omega\|_F^2 \tag{28}$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r\times n}$ with r being the rank of the target matrix. After obtaining $\boldsymbol{U}$ and $\boldsymbol{V}$, the target matrix is computed as $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{V}$.

According to the matrix completion formulation, the low-rank tensor completion problem can be written as:

$$\begin{aligned}&\min_{\boldsymbol{\mathcal{M}}} \text{rank}(\boldsymbol{\mathcal{M}})\\&\text{s.t.}\, \boldsymbol{\mathcal{M}} \odot \boldsymbol{\Omega} = \boldsymbol{\mathcal{Y}}_\Omega\end{aligned} \tag{29}$$

where $\boldsymbol{\mathcal{Y}}_\Omega \in \mathbb{R}^{I_1\times\cdots\times I_n}$ is a partially observed tensor. Herein, $\boldsymbol{\Omega}$ is a binary tensor having the same dimensions with $\boldsymbol{\mathcal{Y}}_\Omega$. Inspired by matrix factorization approach, Wang *et al.* have proposed utilizing TR decomposition to solve low-rank tensor completion [34], leading to the following optimization problem:

$$\min_{\boldsymbol{\mathcal{X}}_i: i=1,\ldots,n} \|U(\boldsymbol{\mathcal{X}}_1\cdots\boldsymbol{\mathcal{X}}_n) \odot \boldsymbol{\Omega} - \boldsymbol{\mathcal{Y}}_\Omega\|_F^2 \tag{30}$$

where $\boldsymbol{\mathcal{X}}_i \in \mathbb{R}^{R\times I_i\times R}$ and R is the predefined TR rank. However, impulsive noise appears in many practical scenarios [36], such as imperfection in sensors and communication channels. It is well known that ℓ_2-space optimization works well in the noiseless and Gaussian noise cases [47], while the performance of ℓ_2-space optimization will significantly deteriorate in the presence of outliers. One popular method to resist outliers is to employ ℓ_p-norm ($0 < p < 2$). Whereas the ℓ_p-norm is nonsmooth and nonconvex with $0 < p \leq 1$. Nonsmoothness introduces difficulty to optimize, and nonconvexity cannot guarantee the global solution.

In order to suppress impulsive noise, we apply the $\ell_{p,\epsilon}$-norm instead of the ℓ_p-norm in (30):

$$\min_{\boldsymbol{\mathcal{X}}_i: i=1,\ldots,n} \|U(\boldsymbol{\mathcal{X}}_1\cdots\boldsymbol{\mathcal{X}}_n) \odot \boldsymbol{\Omega} - \boldsymbol{\mathcal{Y}}_\Omega\|_{p,\epsilon}^p. \tag{31}$$

Since (31) includes n tensor variables, we adopt BCD [48], [49] as the solver. The BCD method cyclically optimizes one $\boldsymbol{\mathcal{X}}_i$ at each iteration while fixing the remaining variables:

$$\begin{aligned}\boldsymbol{\mathcal{X}}_1^{k+1} &= \arg\min_{\boldsymbol{\mathcal{X}}_1} \|U(\boldsymbol{\mathcal{X}}_1\boldsymbol{\mathcal{X}}_2^k\cdots\boldsymbol{\mathcal{X}}_n^k) \odot \boldsymbol{\Omega} - \boldsymbol{\mathcal{Y}}_\Omega\|_{p,\epsilon}^p\\ \boldsymbol{\mathcal{X}}_2^{k+1} &= \arg\min_{\boldsymbol{\mathcal{X}}_2} \|U(\boldsymbol{\mathcal{X}}_1^{k+1}\boldsymbol{\mathcal{X}}_2\cdots\boldsymbol{\mathcal{X}}_n^k) \odot \boldsymbol{\Omega} - \boldsymbol{\mathcal{Y}}_\Omega\|_{p,\epsilon}^p\\ &\vdots\\ \boldsymbol{\mathcal{X}}_n^{k+1} &= \arg\min_{\boldsymbol{\mathcal{X}}_n} \|U(\boldsymbol{\mathcal{X}}_1^{k+1}\boldsymbol{\mathcal{X}}_2^{k+1}\cdots\boldsymbol{\mathcal{X}}_n) \odot \boldsymbol{\Omega} - \boldsymbol{\mathcal{Y}}_\Omega\|_{p,\epsilon}^p\end{aligned} \tag{32}$$

where k represents the iteration number in the BCD scheme. It can be seen from (32) that n subproblems have the same structure. Therefore, we focus on one of them, say, $\boldsymbol{\mathcal{X}}_i$, without loss of generality. According to Lemma 1, minimizing $\boldsymbol{\mathcal{X}}_i$ can be rewritten as:

$$\boldsymbol{\mathcal{X}}_i^{k+1} = \arg\min_{\boldsymbol{\mathcal{X}}_i} \|U(\boldsymbol{\mathcal{X}}_i\boldsymbol{\mathcal{X}}^k) \odot \boldsymbol{\Omega}^{P_i} - \boldsymbol{\mathcal{Y}}_\Omega^{P_i}\|_{p,\epsilon}^p \tag{33}$$

where $\boldsymbol{\mathcal{X}}^k = \boldsymbol{\mathcal{X}}_{i+1}^k\cdots\boldsymbol{\mathcal{X}}_n^k\boldsymbol{\mathcal{X}}_1^k\cdots\boldsymbol{\mathcal{X}}_{i-1}^k \in \mathbb{R}^{R\times(I_{i+1}\cdots I_n I_1\cdots I_{i-1})\times R}$. Then, based on mode-$i$ unfolding,

Algorithm 1: $\ell_{p,\epsilon}$-RTRC.

```
Input: Tensor with missing entries $\boldsymbol{\mathcal{Y}}_\Omega \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_n}$,
  binary tensor $\boldsymbol{\Omega} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_n}$, TR rank $R$
  Initialize: Randomize $\boldsymbol{\mathcal{X}}_i =\in \mathbb{R}^{R\times I_i\times R}$ with $i \in [1,n]$
  for $k = 1, 2, \cdots$ do
    for $i = 1 : n$ do
      for $\delta_i = 1 : I_i$ do
        1) Utilize Algorithm 3 to calculate
   $\boldsymbol{\mathcal{X}}_i^{k+1}(:,\delta_i,:) = \arg\min_{\boldsymbol{X}_i^{\delta_i}} \|\boldsymbol{A}\mathrm{vec}(\boldsymbol{X}_i^{\delta_i}) - \boldsymbol{y}_{\Omega_{\delta_i}}\|_{p,\epsilon}^p$
      end for
    end for
    Stop if stopping criterion is met.
  end for
Output: $\boldsymbol{\mathcal{M}} = U(\boldsymbol{\mathcal{X}}_1^{k+1}\boldsymbol{\mathcal{X}}_2^{k+1}\cdots\boldsymbol{\mathcal{X}}_n^{k+1})$
```

(34) is equivalent to the following matrix completion problem:

$$\boldsymbol{\mathcal{X}}_i^{k+1} = \arg\min_{\boldsymbol{\mathcal{X}}_i} \|U(\boldsymbol{\mathcal{X}}_i\boldsymbol{\mathcal{X}}^{k})_{[i]} \odot \boldsymbol{\Omega}_{[i]}^{P_i} - \boldsymbol{\mathcal{Y}}_{\Omega\ [i]}^{P_i}\|_{p,\epsilon}^p. \tag{34}$$

Since $\boldsymbol{\mathcal{X}}_i$ can be decomposed into I_i lateral slices, denoted as $\boldsymbol{\mathcal{X}}_i(:,\delta_i,:)$ with $0 < \delta_i < I_i$, (34) can be split into I_i subproblems:

$$\begin{aligned}\boldsymbol{\mathcal{X}}_i^{k+1}(:,\delta_i,:) = \ &\arg\min_{\boldsymbol{\mathcal{X}}_i(:,\delta_i,:)} \|U(\boldsymbol{\mathcal{X}}_i(:,\delta_i,:)\boldsymbol{\mathcal{X}}^{k})_{[i]}\\ &\odot \boldsymbol{\Omega}_{[i]}^{P_i}(\delta_i,:) - \boldsymbol{\mathcal{Y}}_{\Omega\ [i]}^{P_i}(\delta_i,:)\|_{p,\epsilon}^p.\end{aligned} \tag{35}$$

Note that all of $U(\boldsymbol{\mathcal{X}}_i(:,\delta_i,:)\boldsymbol{\mathcal{X}}^k)_{[i]}$, $\boldsymbol{\Omega}_{[i]}^{P_i}(\delta_i,:)$ and $\boldsymbol{\mathcal{Y}}_{\Omega\ [i]}^{P_i}(\delta_i,:)$ are vectors. Because the result of (35) is only influenced by the observed entries, (35) is equivalent to:

$$\boldsymbol{\mathcal{X}}_i^{k+1}(:,\delta_i,:) = \arg\min_{\boldsymbol{\mathcal{X}}_i(:,\delta_i,:)} \|U(\boldsymbol{\mathcal{X}}_i(:,\delta_i,:)\boldsymbol{\mathcal{X}}_{\Omega_{\delta_i}}^{k})_{[i]} - \boldsymbol{y}_{\Omega_{\delta_i}}\|_{p,\epsilon}^p \tag{36}$$

herein $\boldsymbol{\mathcal{X}}_{\Omega_{\delta_i}}^k \in \mathbb{R}^{R\times\|\boldsymbol{\Omega}_{\delta_i}\|_0\times R}$ and $\boldsymbol{y}_{\Omega_{\delta_i}} \in \mathbb{R}^{\|\boldsymbol{\Omega}_{\delta_i}\|_0}$ include the observed entries of $\boldsymbol{\mathcal{X}}^k$ and $\boldsymbol{\mathcal{Y}}_{\Omega\ [i]}^{P_i}(\delta_i,:)$, respectively, where $\boldsymbol{\Omega}_{\delta_i}$ is the subset of $\boldsymbol{\Omega}_{[i]}^{P_i}(\delta_i,:)$, and only contains entries of 1. The tensor $\boldsymbol{\mathcal{X}}_i(:,\delta_i,:) \in \mathbb{R}^{R\times 1\times R}$ can be represented by a matrix $\boldsymbol{X}_i^{\delta_i} \in \mathbb{R}^{R\times R}$. Then, based on (11), (36) can be rewritten as a sum of all entries, leading to:

$$\begin{aligned}\boldsymbol{\mathcal{X}}_i^{k+1}(:,\delta_i,:) = \arg\min_{\boldsymbol{X}_i^{\delta_i}} \sum_{j=1}^{\|\boldsymbol{\Omega}_{\delta_i}\|_0} &\|\mathrm{tr}(\boldsymbol{X}_i^{\delta_i} \times \boldsymbol{\mathcal{X}}_{\Omega_{\delta_i}}^k(:,j,:))\\ &- \boldsymbol{y}_{\Omega_{\delta_i}}(j)\|_{p,\epsilon}^p\end{aligned} \tag{37}$$

Lemma 4: Suppose $\boldsymbol{A} \in \mathbb{R}^{I_1\times I_2}$ and $\boldsymbol{B} \in \mathbb{R}^{I_2\times I_1}$, we have:

$$\mathrm{tr}(\boldsymbol{A}\times\boldsymbol{B}) = \mathrm{vec}(\boldsymbol{B}^{\mathrm{T}})^{\mathrm{T}}\mathrm{vec}(\boldsymbol{A}). \tag{38}$$

According to Lemma 4, (37) can be simplified as:

$$\boldsymbol{\mathcal{X}}_i^{k+1}(:,\delta_i,:) = \arg\min_{\boldsymbol{X}_i^{\delta_i}} \|\boldsymbol{A}\mathrm{vec}(\boldsymbol{X}_i^{\delta_i}) - \boldsymbol{y}_{\Omega_{\delta_i}}\|_{p,\epsilon}^p \tag{39}$$

where $\boldsymbol{A} \in \mathbb{R}^{\|\boldsymbol{\Omega}_{\delta_i}\|_0\times R^2}$ and $\boldsymbol{A}(j,:) = \mathrm{vec}(\boldsymbol{\mathcal{X}}_{\Omega_{\delta_i}}^k(:,j,:)^T)^T$ with $j \in [1, \|\boldsymbol{\Omega}_{\delta_i}\|_0]$. In our work, (39) is called $\ell_{p,\epsilon}$-RLR. In the next section, we propose two efficient methods to solve $\ell_{p,\epsilon}$-RLR.

The robust TR completion ($\ell_{p,\epsilon}$-RTRC) is summarized in Algorithm 1. The stopping condition depends on the tolerance parameter:

$$\eta = \frac{\|U(\boldsymbol{\mathcal{X}}_1^{k+1}\boldsymbol{\mathcal{X}}_2^{k+1}\cdots\boldsymbol{\mathcal{X}}_n^{k+1})_{[1]} - U(\boldsymbol{\mathcal{X}}_1^{k}\boldsymbol{\mathcal{X}}_2^{k}\cdots\boldsymbol{\mathcal{X}}_n^{k})_{[1]}\|_F^2}{\|U(\boldsymbol{\mathcal{X}}_1^{k}\boldsymbol{\mathcal{X}}_2^{k}\cdots\boldsymbol{\mathcal{X}}_n^{k})_{[1]}\|_F^2}. \tag{40}$$

When $\eta < 10^{-4}$ is reached, we terminate the algorithm. Our experiments show that the proposed algorithm usually converges within a dozen iterations. To facilitate proving the local convergence of Algorithm 1, we define a cost function:

$$\mathcal{C}(\boldsymbol{\mathcal{X}}_1,\ldots,\boldsymbol{\mathcal{X}}_i,\ldots\boldsymbol{\mathcal{X}}_n) = \|U(\boldsymbol{\mathcal{X}}_1\ldots\boldsymbol{\mathcal{X}}_i\ldots\boldsymbol{\mathcal{X}}_n)\odot\boldsymbol{\Omega} - \boldsymbol{\mathcal{Y}}_\Omega\|_{p,\epsilon}^p. \tag{41}$$

Then we prove that $\ell_{p,\epsilon}$-RTRC keeps $\mathcal{C}(\boldsymbol{\mathcal{X}}_1,\ldots,\boldsymbol{\mathcal{X}}_i,\ldots,\boldsymbol{\mathcal{X}}_n)$ nonincreasing.

Proof:

$$\begin{aligned}&\mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k+1},\ldots,\boldsymbol{\mathcal{X}}_i^{k+1},\ldots,\boldsymbol{\mathcal{X}}_n^{k+1}) - \mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k},\ldots,\boldsymbol{\mathcal{X}}_i^{k},\ldots,\boldsymbol{\mathcal{X}}_n^{k})\\ &=\mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k+1},\boldsymbol{\mathcal{X}}_2^{k},\ldots,\boldsymbol{\mathcal{X}}_n^{k}) - \mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k},\boldsymbol{\mathcal{X}}_2^{k},\ldots,\boldsymbol{\mathcal{X}}_n^{k})\\ &+\ldots\\ &+\mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k+1},\ldots,\boldsymbol{\mathcal{X}}_i^{k+1},\ldots,\boldsymbol{\mathcal{X}}_n^{k}) - \mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k+1},\ldots,\boldsymbol{\mathcal{X}}_i^{k},\ldots,\boldsymbol{\mathcal{X}}_n^{k})\\ &+\ldots\\ &+\mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k+1}\ldots,\boldsymbol{\mathcal{X}}_i^{k+1},\ldots,\boldsymbol{\mathcal{X}}_n^{k+1}) - \mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k+1},\ldots,\boldsymbol{\mathcal{X}}_i^{k+1},\ldots,\boldsymbol{\mathcal{X}}_n^{k}).\end{aligned} \tag{42}$$

We know that each block $\boldsymbol{\mathcal{X}}_i$ is updated by the same scheme, namely (39). If $\mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k+1},\ldots,\boldsymbol{\mathcal{X}}_i^{k+1},\ldots,\boldsymbol{\mathcal{X}}_n^{k}) - \mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k+1},\ldots,\boldsymbol{\mathcal{X}}_i^{k},\ldots,\boldsymbol{\mathcal{X}}_n^{k}) \le 0$ holds, we can obtain $\mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k+1},\ldots,\boldsymbol{\mathcal{X}}_i^{k+1},\ldots,\boldsymbol{\mathcal{X}}_n^{k+1}) - \mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k},\cdot,\boldsymbol{\mathcal{X}}_i^{k},\ldots,\boldsymbol{\mathcal{X}}_n^{k}) \le 0$. Problem (39) is a convex optimization problem because $\|\cdot\|_{p,\epsilon}^p$ with $0<p<2$ is a convex operator. In addition, $\boldsymbol{\mathcal{X}}_i^{k+1}$ is determined from Algorithm 3, and the convergence of Algorithm 3 is guaranteed. Thus, $\boldsymbol{\mathcal{X}}_i^{k+1}$ minimizes $\mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k+1},\ldots,\boldsymbol{\mathcal{X}}_i,\ldots,\boldsymbol{\mathcal{X}}_n^{k})$, resulting in $\mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k+1},\ldots,\boldsymbol{\mathcal{X}}_i^{k+1},\ldots,\boldsymbol{\mathcal{X}}_n^{k}) - \mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k+1},\ldots,\boldsymbol{\mathcal{X}}_i^{k},\ldots,\boldsymbol{\mathcal{X}}_n^{k}) \le 0$. Therefore, $\mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k+1},\ldots,\boldsymbol{\mathcal{X}}_i^{k+1},\ldots,\boldsymbol{\mathcal{X}}_n^{k+1}) - \mathcal{C}(\boldsymbol{\mathcal{X}}_1^{k},\ldots,\boldsymbol{\mathcal{X}}_i^{k},\ldots,\boldsymbol{\mathcal{X}}_n^{k}) \le 0$, which implies that $\mathcal{C}(\boldsymbol{\mathcal{X}}_1,\ldots,\boldsymbol{\mathcal{X}}_i,\ldots,\boldsymbol{\mathcal{X}}_n)$ is nonincreasing. It is clear that the loss function is upper bounded by $\mathcal{C}(\boldsymbol{\mathcal{X}}_1^{1},\ldots,\boldsymbol{\mathcal{X}}_i^{1},\ldots,\boldsymbol{\mathcal{X}}_n^{1})$ while $\mathcal{C}(\boldsymbol{\mathcal{X}}_1,\ldots,\boldsymbol{\mathcal{X}}_i,\ldots,\boldsymbol{\mathcal{X}}_n)$ is lower bounded by 0. Given that (31) is a nonconvex optimization problem, the local convergence of $\ell_{p,\epsilon}$-RTRC is thus guaranteed.

V. $\ell_{p,\epsilon}$-Norm for Robust Linear Regression

In this section, two efficient approaches are presented to solve the $\ell_{p,\epsilon}$-RLR problem, namely GD-$\ell_{p,\epsilon}$-norm and ADMM-$\ell_{p,\epsilon}$-norm. We first use a concise expression to replace (39), leading to:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_{p,\epsilon}^p \tag{43}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$.

Algorithm 2: GD-$\ell_{p,\epsilon}$-Norm.

```
Input: $\boldsymbol{A}$, $\boldsymbol{y}$ and $\mu$
  Initialize: randomize $\boldsymbol{x}^1$
  for $t = 1, 2, \cdots$ do
    1) adjust $\epsilon = \sqrt{1-p}\|\boldsymbol{A}\boldsymbol{x}^t - \boldsymbol{y}\|_\infty + 10^{-2}$
    2) compute $\nabla f(\boldsymbol{x}^t)$
    3) $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \mu\nabla f(\boldsymbol{x}^k)$
    Stop if stopping criterion is met.
  end for
Output: $\hat{\boldsymbol{x}} = \boldsymbol{x}^{t+1}$
```

A. GD-Based Method

Since (43) is an unconstrained and convex optimization problem with $0 < p < 2$, the simplest scheme is GD [50], [51]. For (43), the gradient of $f(\boldsymbol{x})$ is:

$$\nabla f(\boldsymbol{x}) = p\boldsymbol{A}^T\boldsymbol{g} \tag{44}$$

where $\boldsymbol{g} \in \mathbb{R}^m$ and $g_i = ((\boldsymbol{a}_i^T\boldsymbol{x} - y_i)^2 + \epsilon^2)^{p/2-1}(\boldsymbol{a}_i^T\boldsymbol{x} - y_i)$ is the ith entry of $\boldsymbol{g}$ with $\boldsymbol{a}_i^T$ and y_i being the ith row of $\boldsymbol{A}$ and ith entry of $\boldsymbol{y}$, respectively. The GD utilizes the negative direction of $\nabla f(\boldsymbol{x})$ to optimize $\boldsymbol{x}$ with step size μ. Algorithm 2 summarizes the GD-$\ell_{p,\epsilon}$-norm method.

In our study, the stopping condition is:

$$\frac{\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\|_2^2}{\|\boldsymbol{x}^t\|_2^2} \leq 10^{-8}. \tag{45}$$

The GD method is a traditional technique such that the convergence of GD has been analyzed by lots of prior works. One famous book written by Boyd and Vandenberghe provides the detailed proof of the convergence [51].

B. ADMM-Based Method

It is known that ADMM is a simple and powerful method to solve constrained convex optimization problems since it inherits the merits of dual ascent and augmented Lagrangian algorithms, namely the decomposability of dual ascent and the superior convergence property of augmented Lagrangian. Furthermore, the high efficiency of ADMM in the big-scale problem has been revealed [52].

To adopt ADMM, it requires converting the unconstrained problem (43) to a constrained problem via introducing $\boldsymbol{e} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}$, leading to:

$$\begin{aligned}&\min_{\boldsymbol{x},\boldsymbol{e}} \|\boldsymbol{e}\|_{p,\epsilon}^p \\ &\text{s.t.}\, \boldsymbol{e} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}.\end{aligned} \tag{46}$$

Then, the augmented Lagrangian of (46) is:

$$\begin{aligned}\mathcal{L}_\lambda(\boldsymbol{x}, \boldsymbol{e}, \boldsymbol{\Lambda}) = &\|\boldsymbol{e}\|_{p,\epsilon}^p + \langle \boldsymbol{\Lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} - \boldsymbol{e}\rangle \\ &+ \frac{\lambda}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} - \boldsymbol{e}\|_F^2\end{aligned} \tag{47}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^m$ contains the Lagrange multipliers and $\lambda > 0$ is penalty parameter. According to the dual ascent method, (47) can be solved by iterative updating with the following strategy:

$$(\boldsymbol{x}^{t+1}, \boldsymbol{e}^{t+1}) := \arg\min_{\boldsymbol{x},\boldsymbol{e}} \mathcal{L}_\lambda(\boldsymbol{x}, \boldsymbol{e}, \boldsymbol{\Lambda}^t) \tag{48}$$

$$\boldsymbol{\Lambda}^{t+1} = \boldsymbol{\Lambda}^t + \lambda(\boldsymbol{A}\boldsymbol{x}^{t+1} - \boldsymbol{y} - \boldsymbol{e}^{t+1}). \tag{49}$$

Yet, dual ascent cannot directly solve (48) since it is minimized with respect to two optimization variables. To solve this issue, ADMM which adopts the alternating direction approach to update $\boldsymbol{x}$ and $\boldsymbol{e}$ is applied, consisting of the following iterations:

$$\boldsymbol{x}^{t+1} := \arg\min_{\boldsymbol{x}} \mathcal{L}_\lambda(\boldsymbol{x}, \boldsymbol{e}^t, \boldsymbol{\Lambda}^t) \tag{50}$$

$$\boldsymbol{e}^{t+1} := \arg\min_{\boldsymbol{e}} \mathcal{L}_\lambda(\boldsymbol{x}^{t+1}, \boldsymbol{e}, \boldsymbol{\Lambda}^t) \tag{51}$$

$$\boldsymbol{\Lambda}^{t+1} = \boldsymbol{\Lambda}^t + \lambda(\boldsymbol{A}\boldsymbol{x}^{t+1} - \boldsymbol{y} - \boldsymbol{e}^{t+1}) \tag{52}$$

The subproblem (50) is equivalent to the following ℓ_2-norm minimization problem:

$$\min_{\boldsymbol{x}} \left\|\boldsymbol{A}\boldsymbol{x} - \left(\boldsymbol{y} + \boldsymbol{e}^t - \frac{\boldsymbol{\Lambda}^t}{\lambda}\right)\right\|_2^2 \tag{53}$$

which has a closed-form solution: $\boldsymbol{x}^{t+1} = \boldsymbol{A}^\dagger(\boldsymbol{y} + \boldsymbol{e}^t - \frac{\boldsymbol{\Lambda}^t}{\lambda})$. For subproblem (51), it can be simplified as:

$$\min_{\boldsymbol{e}} \frac{1}{\lambda}\|\boldsymbol{e}\|_{p,\epsilon}^p + \frac{1}{2}\|\boldsymbol{e} - \boldsymbol{z}^t\|_2^2 \tag{54}$$

where

$$\boldsymbol{z}^t = \boldsymbol{A}\boldsymbol{x}^{t+1} - \boldsymbol{y} + \frac{\boldsymbol{\Lambda}^t}{\lambda}. \tag{55}$$

It is easy to reveal that (54) can be decomposed into m independent scalar problems:

$$\min_{e_l} \sum_{l=1}^{m} \left[\frac{1}{\lambda}((e_l^2 + \epsilon^2)^{p/2} - \epsilon^p) + \frac{1}{2}(e_l - z_l^k)^2\right] \tag{56}$$

where e_l and z_l^k are the lth entry of $\boldsymbol{e}$ and $\boldsymbol{z}^t$, respectively. To the best of our knowledge, it is the first time to propose the problem (56) which has not been solved with $0 < p \leq 1$. To tackle (56), we first define:

$$J(e_l) = \frac{1}{\lambda}((e_l^2 + \epsilon^2)^{p/2} - \epsilon^p) + \frac{1}{2}(e_l - z_l^t)^2 \tag{57}$$

then the minimizer of $\min_{e_l} J(e_l)$ can be calculated via:

$$J'(e_l) = \frac{p}{\lambda}e_l(e_l^2 + \epsilon^2)^{p/2-1} + e_l - z_l^t = 0. \tag{58}$$

For $z_l^t \neq 0$, it is found that $J'(|z_l^t|) > 0$ and $J'(-|z_l^t|) < 0$. That is to say, $J'(|z_l^t|)J'(-|z_l^t|) < 0$ which means that the root of $J'(e_l) = 0$ lies in $[-|z_l^t|, |z_l^t|]$. Moreover, the second derivative of $J(e_l)$ is:

$$J''(e_l) = \frac{p}{\lambda}(e_l^2 + \epsilon^2)^{p/2-2}((p-1)e_l^2 + \epsilon^2) + 1. \tag{59}$$

According to the Property (v) of $\ell_{p,\epsilon}$-norm, $(p-1)e_l^2 + \epsilon^2 > 0$ under a given ϵ. Hence, $J''(e_l) > 0$ holds for e_l in $[-|z_l^t|, |z_l^t|]$ which implies that $J'(e_l)$ is monotonically increasing in $[-|z_l^t|, |z_l^t|]$. Therefore, the root of $J'(e_l) = 0$ is unique in $[-|z_l^t|, |z_l^t|]$, and can be quickly calculated via the bisection method with a complexity of $\mathcal{O}(1)$ [55], [56]. On the other

```
Algorithm 3: ADMM-$\ell_{p,\epsilon}$-Norm.
Input: $\boldsymbol{A}$, $\boldsymbol{y}$ and $\lambda$
  Initialize: randomize $\boldsymbol{e}^0$ and $\boldsymbol{\Lambda}^0$
  for $t = 1, 2, \cdots$ do
    1) $\boldsymbol{x}^{t+1} = \boldsymbol{A}^{-1}(\boldsymbol{y} + \boldsymbol{e}^t - \frac{\boldsymbol{\Lambda}^t}{\lambda})$
    2) calculate $\boldsymbol{z}^t = \boldsymbol{A}\boldsymbol{x}^{t+1} - \boldsymbol{y} + \frac{\boldsymbol{\Lambda}^t}{\lambda}$
    3) adjust $\epsilon = \sqrt{1-p}\max(\|\boldsymbol{e}^t\|_\infty, \|\boldsymbol{z}^t\|_\infty) + 10^{-2}$
    4) $\boldsymbol{e}^{t+1} := \arg\min_{\boldsymbol{e}} \mathcal{L}_\lambda(\boldsymbol{x}^{t+1}, \boldsymbol{e}, \boldsymbol{\Lambda}^t)$
    5) $\boldsymbol{\Lambda}^{t+1} = \boldsymbol{\Lambda}^t + \lambda(\boldsymbol{A}\boldsymbol{x}^{t+1} - \boldsymbol{y} - \boldsymbol{e}^{t+1})$
    Stop if stopping criterion is met.
  end for
Output: $\hat{\boldsymbol{x}} = \boldsymbol{x}^{t+1}$
```

hand, for $z_l^t = 0$, the minimizer and the minimum of $J(e_l)$ are obviously 0.

For $1 < p < 2$, (56) is the same as the standard ℓ_p-norm problem. Then, $J(e_l)$ is simplified as:

$$J(e_l) = \frac{1}{\lambda}|e_l|_p^p + \frac{1}{2}(e_l - z_l^t)^2 \tag{60}$$

which has been tackled [57]:

$$e_l^* = \begin{cases} \arg \min\limits_{e_l \in \{0, e_l^+\}} J(e_l), & \text{if } z_l^t \geq 0 \\ \arg \min\limits_{e_l \in \{0, e_l^-\}} J(e_l), & \text{if } z_l^t < 0 \end{cases} \tag{61}$$

where e_l^+ and e_l^- are the solutions corresponding to $J(e_l)$ in the cases of $z_l^t \geq 0$ and $z_l^t < 0$, respectively. For $z_l^t \geq 0$, $J'(e_l)$ is monotonically increasing in $[0, +\infty)$. Furthermore, $J'(0) < 0$ and $J'(z_l^t) > 0$ indicate that a unique root e_l^+ lies in $[0, z_l^t]$, and can be found by the bisection method. In the same manner, the unique root $e_l^- \in [z_l^t, 0]$ is obtained from solving $J'(e_l) = 0$ for $z_l^t < 0$ via the bisection approach as well since $J'(e_l)$ monotonically increases in $[z_l^t, 0]$, and $J'(0)J'(z_l^t) < 0$.

For (52), the gradient of $\mathcal{L}_\lambda(\boldsymbol{x}^{t+1}, \boldsymbol{e}^{t+1}, \boldsymbol{\Lambda})$ with respect to $\boldsymbol{\Lambda}$ is

$$\frac{\partial \mathcal{L}_\lambda(\boldsymbol{x}^{t+1}, \boldsymbol{e}^{t+1}, \boldsymbol{\Lambda})}{\partial \boldsymbol{\Lambda}} = \boldsymbol{A}\boldsymbol{x}^{t+1} - \boldsymbol{y} - \boldsymbol{e}^{t+1}. \tag{62}$$

In summary, ADMM solves the dual problem of (47):

$$\max_{\boldsymbol{\Lambda}} \min_{\boldsymbol{x}, \boldsymbol{e}} \mathcal{L}_\lambda(\boldsymbol{x}, \boldsymbol{e}, \boldsymbol{\Lambda}). \tag{63}$$

The ADMM-$\ell_{p,\epsilon}$-norm method is summarized in Algorithm 3. The stopping criterion is the same as that of Algorithm 2.

Regarding ADMM analysis, the theoretical convergence of closed, proper and convex functions has been proved in [52]. The $\ell_{p,\epsilon}$-norm with $0 < p < 2$ is also closed, proper, and convex. In addition, it has been reported that employing ADMM to optimize the function with more than two variables may not ensure convergence [53]. Whereas ADMM-$\ell_{p,\epsilon}$-norm just updates two parameters, namely $\boldsymbol{x}$ and $\boldsymbol{e}$. Therefore, the convergence of the ADMM-$\ell_{p,\epsilon}$-norm is guaranteed.

C. Parameter Setting

We first discuss the step size μ of the GD-$\ell_{p,\epsilon}$-norm. If μ is large, the algorithm will oscillate rather than converge. Conversely, if μ is small, the algorithm requires a large number of iterations to converge. Hence, two schemes to optimize μ in each iteration are proposed, including exact line search and backtracking line search [51]. Both aim to search for a relatively appropriate μ. However, GD-$\ell_{p,\epsilon}$-norm is not adopted to solve $\ell_{p,\epsilon}$-RTRC, because it is difficult to select an appropriate μ in practice. The reason is that large size of $\boldsymbol{A}$ in tensor completion results in very large magnitude of the gradient, which makes convergence precision extremely sensitive to μ. For instance, a change of 10^{-5} in μ may affect the convergence precision, while 10^{-4} can cause oscillation.

Regarding the penalty parameter λ of ADMM-$\ell_{p,\epsilon}$-norm, through our experiments, we have found that λ can be flexibly selected as a constant throughout the iterative process. Nevertheless, the algorithm with a smaller λ will converge faster than that with a larger λ. On the other hand, it has been suggested that a time-varying λ can speed up convergence [52], [54]. But the convergence of ADMM with varying λ has not been proved. Therefore, a fixed value of λ is adopted.

D. Computational Complexity

For the GD-$\ell_{p,\epsilon}$-norm method, the computational complexity is $\mathcal{O}(Tmn)$ where T is the iteration number. Whereas the ADMM-$\ell_{p,\epsilon}$-norm has a higher computational complexity: $\mathcal{O}(Tm^2n)$ because it needs to calculate the inverse of $\boldsymbol{A}$. If RLR is solved by two popular approaches, namely, iteratively reweighted least squares-based ℓ_p-norm (IRLS-ℓ_p-norm) and ADMM-ℓ_p-norm, their computational complexities are $\mathcal{O}(Tm^2n)$.

VI. Simulation Results

All simulations are run on a computer with 3.2 GHz CPU and 16 GB memory.

A. Comparison Between $\ell_{p,\epsilon}$-Norm and ℓ_p-Norm

We first compare the two proposed algorithms with the variants based on ℓ_p-norm to tackle RLR. The results are based on synthetic random data. The entries of $\boldsymbol{A} \in \mathbb{R}^{5000 \times 20}$ and $\boldsymbol{x} \in \mathbb{R}^{20}$ obey the standard Gaussian distribution. The independent noise vector $\boldsymbol{n} \in \mathbb{R}^{5000}$ is generated according to the Gaussian mixture model (GMM) noise whose probability density function is:

$$p_v(v) = \sum_{i=1}^{2} \frac{c_i}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{v^2}{2\sigma_i^2}\right) \tag{64}$$

where $c_1 + c_2 = 1$ with $0 < c_i < 1$ and σ_i^2 is variance. To simulate impulsive noise, $\sigma_2^2 \gg \sigma_1^2$ and $c_2 < c_1$ are used, which means that large noise samples with σ_2^2 and c_2 are mixed in Gaussian background noise with small variance σ_1^2. In our simulations, we set $\sigma_2^2 = 100\sigma_1^2$ and $c_2 = 0.1$. The signal-to-noise

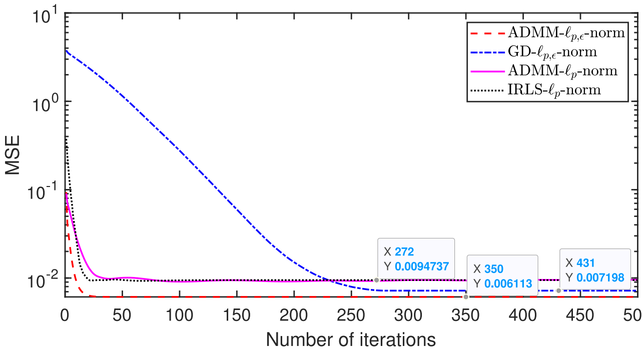


Fig. 3. MSE versus iteration number in GMM noise at SNR $= 3$ dB with $p = 1$ by ADMM-$\ell_{p,\epsilon}$-norm, GD-$\ell_{p,\epsilon}$-norm, ADMM-ℓ_p-norm and IRLS-ℓ_p-norm.

ratio (SNR) is defined as:

$$\text{SNR} = \frac{\|\boldsymbol{x}\|_2^2}{\text{dim}(\boldsymbol{x})\sigma_v^2} \tag{65}$$

where $\sigma_v^2 = \sum_i c_i \sigma_i^2$ is the total noise variance. The observed vector $\boldsymbol{y}$ is generated using $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$. The target is to restore $\boldsymbol{x}$ given $\boldsymbol{A}$ and $\boldsymbol{y}$. Performance is evaluated by the mean square error (MSE), defined as

$$\text{MSE}(\hat{\boldsymbol{x}}) = \text{E}\left\{\frac{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2}{\|\boldsymbol{x}\|_2^2}\right\} \tag{66}$$

where $\hat{\boldsymbol{x}}$ is the recovered vector, and MSE is based on 100 independent trials.

Fig. 3 depicts the convergence performance of ADMM-$\ell_{p,\epsilon}$-norm, GD-$\ell_{p,\epsilon}$-norm, ADMM-ℓ_p-norm and IRLS-ℓ_p-norm with $p = 1$ in additive GMM noise of SNR=3 dB. The steady-state MSEs of ADMM-$\ell_{p,\epsilon}$-norm and GD-$\ell_{p,\epsilon}$-norm are 0.006 113 and 0.007 198, respectively. The ADMM-ℓ_p-norm and IRLS-ℓ_p-norm have a similar steady-state MSE of 0.009 474. Thus, the $\ell_{1,\epsilon}$-norm has better performance on robustness against outliers than ℓ_1-norm owing to Property (i) of Lemma 3. Compared with the GD-$\ell_{p,\epsilon}$-norm, the convergence rate of ADMM-$\ell_{p,\epsilon}$-norm is faster.

Fig. 4 evaluates the performance of the four approaches at $0 < p < 1$ and 3 dB GMM noise where 100 convergence curves are plotted. Compared with the ADMM-ℓ_p-norm and IRLS-ℓ_p-norm, the ADMM-$\ell_{p,\epsilon}$-norm and GD-$\ell_{p,\epsilon}$-norm have an overwhelming advantage. The reason is that ℓ_p-norm with $0 < p < 1$ is non-convex, so it cannot search for the global solution in every trial, which results in 100 inconsistent steady-state MSEs. In contrast, $\ell_{p,\epsilon}$-norm with $0 < p < 1$ is convex such that ADMM-$\ell_{p,\epsilon}$-norm and GD-$\ell_{p,\epsilon}$-norm converge to the same MSE value each time.

Selection of p: Based on the above simulation, we then study the choice of p at different levels of GMM noise. Table I depicts the effect of p on the recovery accuracy under different SNRs. It can be seen that $p = 1.4$ has the best performance for weak outliers. When the SNR of GMM noise goes down to 15 dB and 12 dB, $p = 1.2$ is the best. As the SNR continues to decrease, $p = 1.0$ results in the best performance. While $p = 0.8$ outperforms other values at SNR=6 dB. For the strongest outliers, it requires $p = 0.6$ to obtain the best performance. In short, when the outliers are stronger, the value of p is required to be smaller.

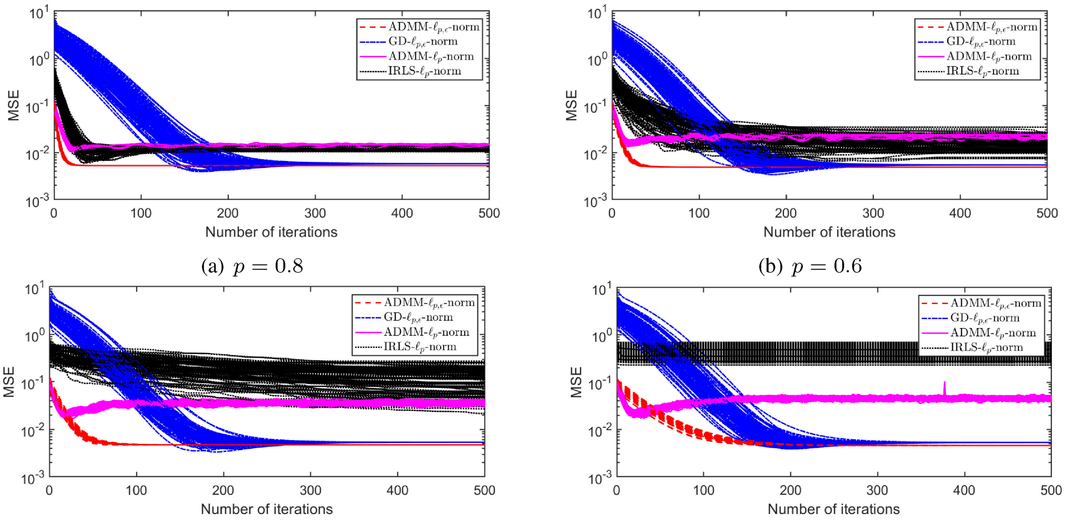
TABLE I
COMPARISON OF p ON RECOVERY ACCURACY AT DIFFERENT SNRs

SNR	$p = 0.6$	$p = 0.8$	$p = 1.0$	$p = 1.2$	$p = 1.4$
18dB	0.000621	0.000514	0.000465	0.000378	**0.000312**
15dB	0.000942	0.000874	0.000805	**0.000591**	0.000904
12dB	0.002621	0.001690	0.001562	**0.001373**	0.002511
9dB	0.003071	0.002669	**0.002461**	0.003424	0.004981
6dB	0.004322	**0.003908**	0.004186	0.005791	0.011644
3dB	**0.004887**	0.005482	0.006112	0.008812	0.013732

B. Color Image Inpainting With Salt-and-Pepper Noise

The first application of low-rank tensor completion is color image inpainting, which is inspired by the widespread use of low-rank matrix completion. In fact, an image may not be fully captured due to damage to the photosensitive device or the shadow from other objects. In addition, the image data will be mixed with impulsive noise during wireless transmission. We know that low-rank matrix completion can deal with gray-scale images represented as matrices. Compared with gray-scale images, color images have RGB channels. Hence, a color image can be viewed as a 3rd-order tensor. In this section, two color images are examined: $Windows(480 \times 500 \times 3)$ and $Einstein(600 \times 600 \times 3)$. To measure the performance of the recovered images, two evaluation indices are utilized, namely peak SNR (PSNR) and structural similarity (SSIM). We directly use the MATLAB commands, namely, 'psnr(recovered, original)' and 'ssim(recovered, original)'. Larger values of PSNR and SSIM mean that the recovery performance is better. Five existing approaches are compared with our ADMM-$\ell_{p,\epsilon}$-RTRC, including tensor ring completion (TRC) [34], fast low rank tensor completion (FaLRTC) [6], simple low rank tensor completion (SiLRTC) [6], ADMM-t-SVD [30], ℓ_p-PARAFAC [37] and IR-t-SVD [38]. In the field of image processing, salt-and-pepper noise is a popular impulsive model and therefore it is added to the incomplete image. Salt-and-pepper noise is generated by the command 'imnoise($\boldsymbol{\mathcal{X}}$, 'salt & pepper,' ρ)' in MATLAB where $\boldsymbol{\mathcal{X}}$ is the image tensor, ρ is the normalized noise intensity. The relationship between ρ and SNR is $\rho = 1/\text{SNR}$.

Fig. 5 depicts the performance of seven methods with 70% data loss and 5 dB salt-and-pepper noise. The case of high intensity salt-and-pepper noise is shown in Fig. 6 where the missing ratio is still 70%, but SNR becomes 2 dB. Combining Figs. 5 and 6, it can be seen that the ADMM-$\ell_{p,\epsilon}$-RTRC can effectively restrain outliers according to slight changes of PSNR and SSIM from 5 dB to 2 dB salt-and-pepper noise. TRC utilizes Frobenius norm so as to have severe degradation when the intensity of salt-and-pepper noise becomes strong. For FaLRTC, SiLRTC and ADMM-t-SVD, these three methods are not designed for noisy situations, hence their performance is unsatisfactory no matter whether the salt-and-pepper noise is weak or strong. It can be observed from Fig. 5 that the images recovered by FaLRTC, SiLRTC and ADMM-t-SVD still retain outliers. Moreover, ℓ_p-PARAFAC and IR-t-SVD employ ℓ_p-norm to resist outliers, but their results are still worse than that



(a) $p = 0.8$

(b) $p = 0.6$

(c) $p = 0.4$

(d) $p = 0.2$

Fig. 4. MSE versus iteration number in GMM noise at SNR = 3 dB with $0 < p >$ where the performance of 100 trials is plotted by ADMM-$\ell_{p,\epsilon}$-norm, GD-$\ell_{p,\epsilon}$-norm, ADMM-ℓ_p-norm and IRLS-ℓ_p-norm.

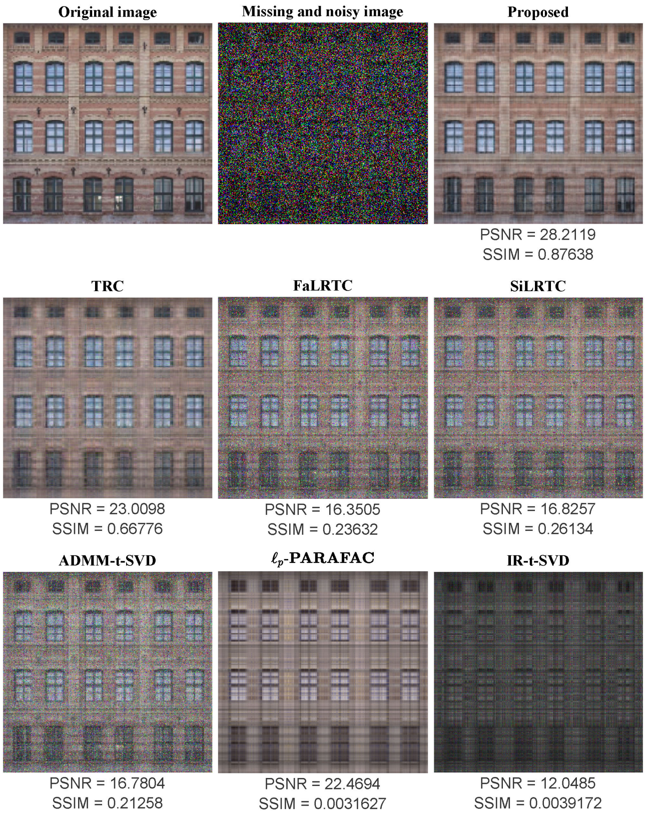


Fig. 5. Performance of different approaches with 70% randomly missing data and SNR = 5 dB salt-and-pepper noise.

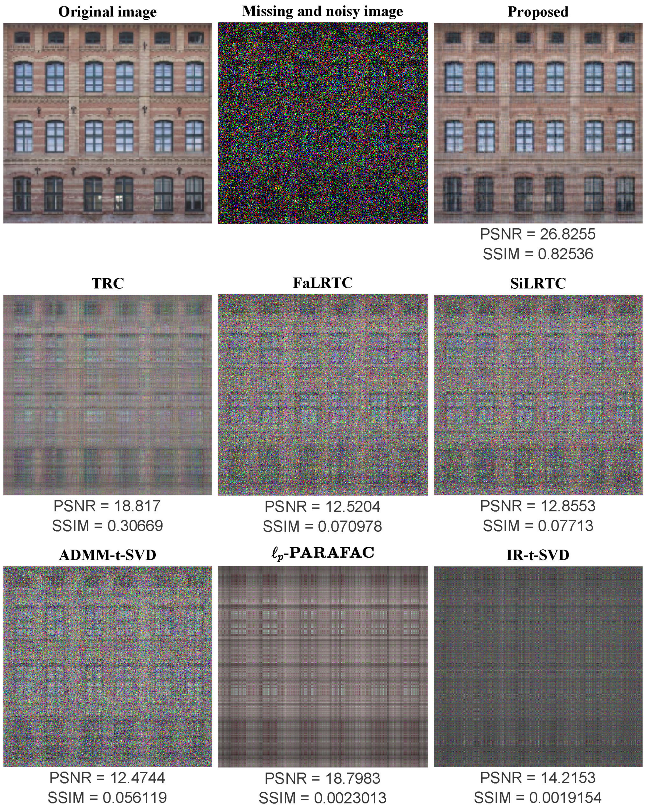


Fig. 6. Performance of different approaches with 70% randomly missing data and 2 dB salt-and-pepper noise.

of ADMM-$\ell_{p,\epsilon}$-RTRC. Furthermore, compared with FaLRTC, SiLRTC and ADMM-t-SVD, we see that ℓ_p-PARAFAC is able to resist outliers, but it cannot ideally recover the image. The reason is that CP rank is not suitable for image data. On the other hand, the reason for poor performance of IR-t-SVD is that tensor tubal rank is only applicable to the outliers following the random tubal distribution.

The impact of p is studied in Fig. 7 which shows the PSNR versus p. We observe that the proposed algorithm has stable and

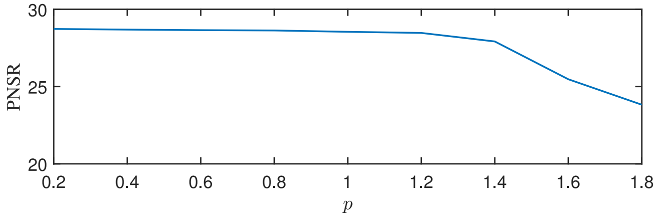


Fig. 7. Performance of p versus PSNR with 70% randomly missing data entries and 5 dB salt-and-pepper noise.

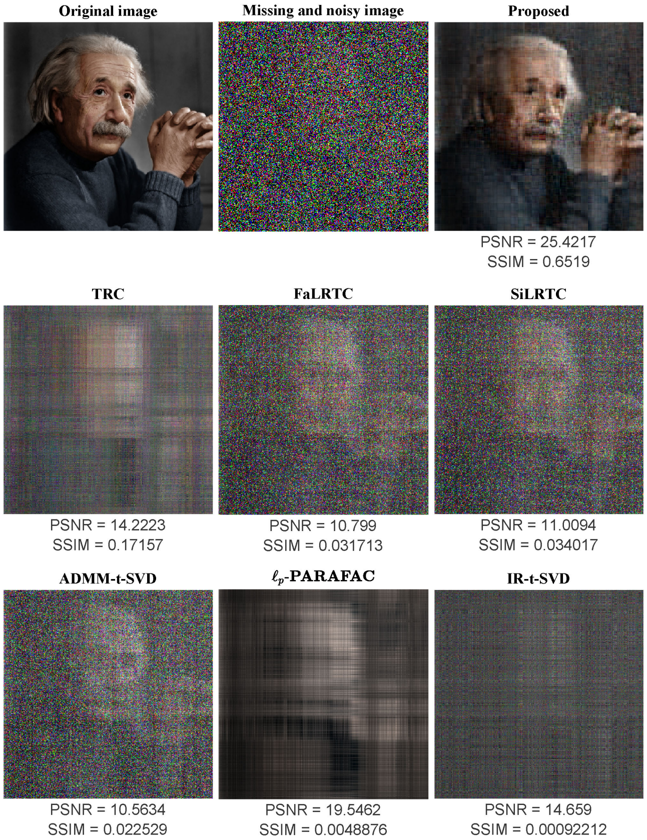


Fig. 8. Results of portrait with 70% missing data and 2 dB salt-and-pepper noise.

good performance in the case of $0 < p < 1$. Yet if robust tensor completion is solved by ℓ_p-norm, the performance will become unstable and poor with $0 < p < 1$ due to the nonconvexity of ℓ_p-norm. On the other hand, the ability to resist outliers becomes weaker with p increasing and hence PSNR reduces as p increases from 1 to 2.

Another image which belongs to the portrait type and has different dimensions from the first image is investigated. The results are shown in Fig. 8. We see that the ADMM-$\ell_{p,\epsilon}$-RTRC with $p = 1$ has the best recovery performance in terms of both objective evaluation indices and subjective observation.

C. Video Inpainting With Salt-and-Pepper Noise

As the extension of color image data, color video is a typical representation of 4th-order tensors, which can be viewed as a set of color images arranged in the time channel in chronological order. The investigated video is about gun shooting [34]. Its dimensions are $100 \times 260 \times 3 \times 85$ which means that it contains 85 color images. Since FaLRTC, SiLRTC, ADMM-t-SVD, ℓ_p-PARAFAC and IR-t-SVD cannot reconstruct the video under a high intensity salt-and-pepper noise, their results are not reported.

Fig. 9 depicts the results recovered by ADMM-$\ell_{p,\epsilon}$-RTRC and TRC under the condition of 30% random observation ratio and 2 dB salt-and-pepper noise. We select 7 representative frames from 85 frames to illustrate the performance, that is, 1st frame, 15th frame, 29th frame, 43 rd frame, 57th frame, 71st frame, 85th frame. It can be seen that the ADMM-$\ell_{p,\epsilon}$-RTRC can effectively resist outliers. The PSNRs in dB corresponding to the seven frames are 21.2830, 21.6936, 22.2226, 22.6552, 22.7389, 22.8456 and 22.8703, respectively. On the other hand, the SSIMs of the seven recovered frames are 0.5723, 0.6083, 0.6654, 0.6972, 0.7180, 0.7338 and 0.7302, respectively. Compared with ADMM-$\ell_{p,\epsilon}$-TRC, the performance of TRC is not satisfactory.

D. Target Estimation With GMM Noise

In this subsection, we apply the proposed approach to deal with the source localization problem in bistatic MIMO radar. In practice, the complete data for the direction-of-departure (DOD) and direction-of-arrival (DOA) estimation may not be collected at the receiver front-end due to various reasons, such as temporarily broken-down transmit or receive antennas, sub-Nyquist sampling, or even intentionally forbidden antennas for energy saving purposes. Here, ADMM-$\ell_{p,\epsilon}$-RTRC is employed to recover the incomplete received data. Four existing methods are also included in this evaluation, which are high accuracy low rank tensor completion (HaLRTC) [6], ℓ_p-PARAFAC, accelerated proximal gradient line-search tensor completion (APGL-TC) [9] and fixed point iterative method for low n-rank tensor pursuit (FP-LRTC) [20].

We assume that the MIMO radar system has the following parameters:

- M_t co-located antennas in the transmit array and M_r co-located antennas in the receive array;
- d_t and d_r are the respective inter-element distances of antennas in the transmit and receive arrays;
- K targets in the range-bin of interest;
- Matrix $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_{M_t}]^T \in \mathbb{R}^{M_t \times L}$ contains the M_t narrowband transmit pulse waveforms, where L is the number of samples per pulse period.
- $\{\beta_k\}$ with $k = 1, 2, \ldots, K$ are the radar cross section fading coefficients;
- $\{\theta_k\}$ and $\{\phi_k\}$ with $k = 1, 2, \ldots, K$ are the DODs and DOAs corresponding to the transmit and receive array normal, respectively.

Then, the baseband received signal at the output of the receive array in the qth pulse after synchronization can be written as:

$$\boldsymbol{X}_q = \boldsymbol{B}\boldsymbol{\Sigma}_q\boldsymbol{A}^T\boldsymbol{S} + \boldsymbol{N}_q \tag{67}$$

where $\boldsymbol{X}_q \in \mathbb{C}^{M_r \times L}$, $\boldsymbol{N}_q \in \mathbb{C}^{M_r \times L}$ is GMM noise matrix, $\boldsymbol{\Sigma}_q = \mathrm{diag}(\boldsymbol{c}_q)$ with $\boldsymbol{c}_q = [\beta_{1q}, \ldots, \beta_{Kq}]$, $\boldsymbol{A} = [\boldsymbol{a}(\theta_1), \ldots, \boldsymbol{a}(\theta_K)]^T \in \mathbb{C}^{M_t \times K}$ and $\boldsymbol{B} =$

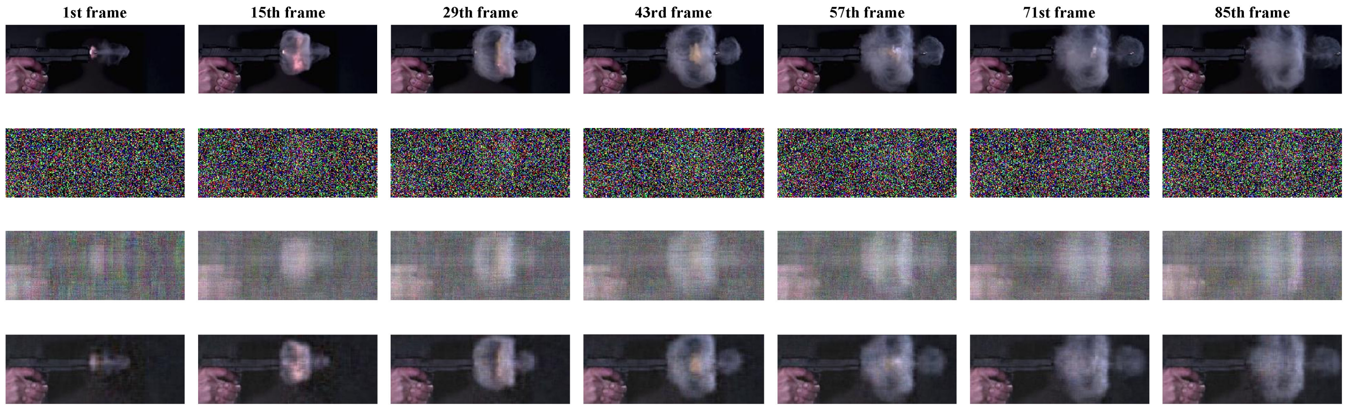
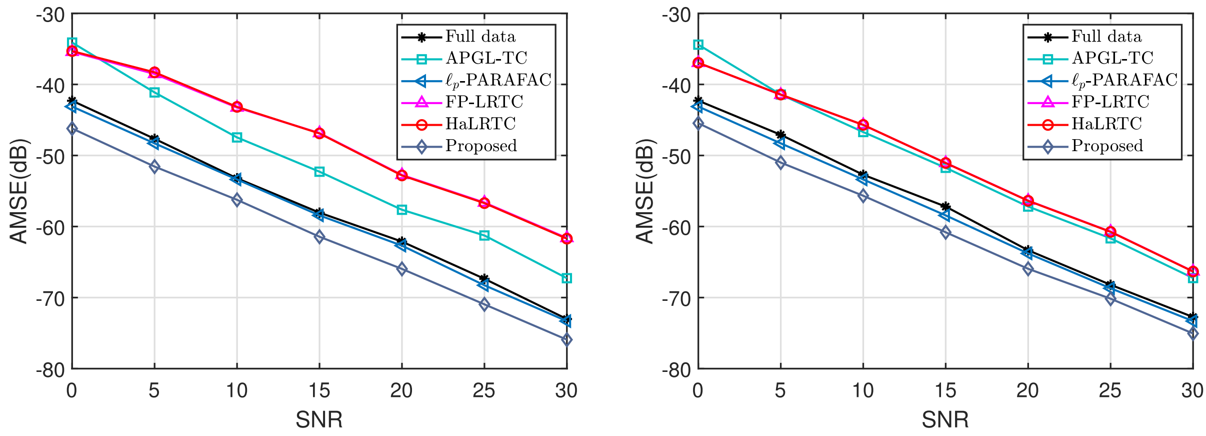


Fig. 9. Performance on video recovery by ADMM-$\ell_{p,\epsilon}$-RTRC and TRC with 70% randomly missing data and SNR = 2 dB salt-and-pepper noise. The first row contains seven frames of the original video. The second row shows the corresponding missing and noisy frames. The results recovered by TRC are shown in the third row. The restored frames by ADMM-$\ell_{1,\epsilon}$-RTRC are represented in the bottom row.



(a) AMSE versus SNR for DOD estimation

(b) AMSE versus SNR for DOA estimation

Fig. 10. AMSE versus different GMM noise levels for different tensor completion methods with 30% observation ratio.

$[\boldsymbol{b}(\phi_1),\ldots,\boldsymbol{b}(\phi_K)]^T \in \mathbb{C}^{M_r\times K}$ are the transmit steering matrix and receive steering matrix, respectively, where

$$\boldsymbol{a}(\theta_k) = [1, e^{j2\pi d_t \sin(\theta_k)/\lambda}, \ldots, e^{j2\pi d_t (M_t-1)\sin(\theta_k)/\lambda}]^T$$
$$\boldsymbol{b}(\phi_k) = [1, e^{j2\pi d_r \sin(\phi_k)/\lambda}, \ldots, e^{j2\pi d_r (M_r-1)\sin(\phi_k)/\lambda}]^T \tag{68}$$

Suppose that there are Q pulses, the received signal can be organized as a 3rd-order tensor, denoted by $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{M_r\times L\times Q}$. The low-rank tensor completion in the DOD and DOA estimation problem assumes that the received signal $\boldsymbol{\mathcal{X}}$ is incomplete. Hence, it requires estimating a received signal $\hat{\boldsymbol{\mathcal{X}}}$ based on the incomplete measurements, and then utilizes $\hat{\boldsymbol{\mathcal{X}}}$ to estimate $\{\theta_k\}$ and $\{\phi_k\}$. In the estimation process, the matched-filter output is:

$$\boldsymbol{\mathcal{Y}}(:,:,q) = \frac{\hat{\boldsymbol{\mathcal{X}}}(:,:,q)\boldsymbol{S}^H}{L} \tag{69}$$

Then, $\{\theta_k\}$ and $\{\phi_k\}$ are estimated based on $\boldsymbol{\mathcal{Y}}$ by the conventional PARAFAC method [8].

In our simulation, $\lambda = 0.3$, $M_t = M_r = 30$, $Q = L = 128$, $K = 5$, $\boldsymbol{\theta} = [10°, 20°, 30°, -10°, 0°]$, $\boldsymbol{\phi} = [25°, -30°, -15°, 15°, 5°]$ and observation ratio is 30% which means that only 30% data of $\boldsymbol{\mathcal{X}}$ is collected. To the evaluate performance, average mean square error (AMSE) between the estimated and true values is adopted:

$$\text{AMSE} = \frac{1}{100K}\sum_{m=1}^{100}\sum_{k=1}^{5}(\xi_k - \hat{\xi_k^m})^2 \tag{70}$$

where $\hat{\xi_k^m}$ denotes the estimate of ξ_k in the mth trial. Herein, ξ_k is θ_k or ϕ_k.

Because the proposed method cannot process complex-valued data, the real and imaginary parts of the data are separated, and then these two parts are restored individually. After obtaining the recovered real and imaginary parts, the restored $\hat{\boldsymbol{\mathcal{X}}}$ is obtained by combining these two parts.

Fig. 10 shows the estimation performance when the SNR of GMM noise varies from 0 dB to 30 dB. "Full data" means that $\boldsymbol{\mathcal{X}}$ is utilized to estimate $\{\theta_k\}$ and $\{\phi_k\}$ by PARAFAC. For each of the remaining methods, it first estimates $\hat{\boldsymbol{\mathcal{X}}}$ based on incomplete $\boldsymbol{\mathcal{X}}_{\boldsymbol{\Omega}} = \boldsymbol{\mathcal{X}} \odot \boldsymbol{\Omega}$ by tensor completion, and then uses $\hat{\boldsymbol{\mathcal{X}}}$ to estimate $\{\theta_k\}$ and $\{\phi_k\}$ by PARAFAC. It can be seen that our method and ℓ_p-PARAFAC have better performance than the other three methods since APGL-TC, FP-LRTC and HaLRTC are not designed for outliers. Moreover, ADMM-$\ell_{p,\epsilon}$-RTRC outperforms ℓ_p-PARAFAC. Note that both the proposed method and ℓ_p-PARAFAC is even superior to the "full data" scheme because they can remove the outliers when constructing $\hat{\mathcal{X}}$.

VII. Conclusion

In this paper, we devise the $\ell_{p,\epsilon}$-norm to replace ℓ_p-norm for resisting outliers. Then we reformulate the robust tensor completion problem by using TR rank and $\ell_{p,\epsilon}$-norm. To solve RLR, two algorithms are developed. The first one adopts GD which has a low computational complexity. In the second method, we utilize ADMM to solve the $\ell_{p,\epsilon}$-RLR problem, yielding a fast convergence rate. It is shown that our methods outperform the algorithms based on ℓ_p-norm to solve RLR. Simulations and experiments based on real-world and synthetic data demonstrate that the proposed robust tensor completion approach is superior to TCR, FaLRTC, SiLRtC, APGL-TC, FP-LRTC, HaLRTC, ℓ_p-PARAFAC and IR-t-SVD in terms of outlier-robustness.

Although the two algorithms are designed for RLR, only ADMM-$\ell_{p,\epsilon}$-norm is utilized to solve the robust tensor completion problem. In tensor completion problem, GD-$\ell_{p,\epsilon}$-norm may occur oscillation rather than convergence. The reason is that the large size of tensor can result in enormous magnitude of the gradient in GD-$\ell_{p,\epsilon}$-norm. The recovery accuracy is quite sensitive to the step-size. Yet the aforementioned methods for step-size selection cannot solve this issue. Therefore, a future work is to appropriately determine the step-size of the GD-$\ell_{p,\epsilon}$-norm in each iteration.

References

[1] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[2] E. J. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[3] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, Apr. 2009.

[4] E. J. Candès and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010.

[5] X. P. Li, Q. Liu, and H. C. So, "Rank-one matrix approximation with ℓ_p-norm for image inpainting," *IEEE Signal Process. Lett.*, vol. 27, pp. 680–684, Apr. 2020.

[6] 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.

[7] 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.

[8] D. Nion and N. D. Sidiropoulos, "Tensor algebra and multidimensional harmonic retrieval in signal processing for MIMO radar," *IEEE Trans. Signal Process.*, vol. 58, no. 11, pp. 5693–5705, Nov. 2010.

[9] L. T. Huang, A. L. F. de Almeida, and H. C. So, "Target estimation in bistatic MIMO radar via tensor completion," *Signal Process.*, vol. 120, pp. 654–659, Mar. 2016.

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

[11] A. Cichocki *et al.*, "Tensor decompositions for signal processing applications: From two-way to multi-way component analysis, " *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, Mar. 2015.

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

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

[14] R. Bro, "PARAFAC. Tutorial and applications," *Chemom. Intell. Lab. Syst.*, vol. 38, no. 2, pp. 149–171, Oct. 1997.

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

[16] M. Kilmer, K. Braman, N. Hao, and R. 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, Jan. 2013.

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

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

[19] Y. Liu and F. Shang, "An efficient matrix factorization method for tensor completion," *IEEE Signal Process. Lett.*, vol. 20, no. 4, pp. 307–310, Apr. 2013.

[20] L. Yang, Z. H. Huang, and X. Shi, "A fixed point iterative method for low-n-rank tensor pursuit," *IEEE Trans. Signal Process.*, vol. 61, no. 11, pp. 2952–2962, Jun. 2013.

[21] Y. Ming and C. H. Zhang, "On tensor completion via nuclear norm minimization," *Found. Comput. Math.*, vol. 16, no. 4, pp. 1031–1068, Aug. 2016.

[22] C. Mu, B. Huang, J. Wright, and D. Goldfarb, "Square deal: Lower bounds and improved relaxations for tensor recovery," in *Proc. 31st Int. Conf. Mach. Learn.*, Beijing, China, Jun. 2014, pp. 73–81.

[23] P. Jain and S. Oh, "Provable tensor factorization with missing data," in *Proc. Adv. Neural Inform. Process. Syst.*, Montreal, Canada, Dec. 2014, pp. 1431–1439.

[24] V. De Silva and L. H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1084–1127, Sep. 2008.

[25] J. Shi, W. Yang, L. Yong, and X. Zheng, "Low-rank tensor completion via tucker decompositions," *J. Comput. Inform. Syst.*, vol. 11, no. 10, pp. 3759–3768, May 2015.

[26] L. Yang, J. Fang, H. Li, and B. Zeng, "An iterative reweighted method for tucker decomposition of incomplete tensors," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4817–4829, Sep. 2016.

[27] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, Aug. 2011.

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

[29] W. Sun, Y. Chen, L. Huang, and H. C. So, "Tensor completion via generalized tensor tubal rank minimization using general unfolding," *IEEE Signal Process. Lett.*, vol. 25, no. 6, pp. 868–872, Jun. 2018.

[30] A. Wang, Z. Lai, and Z. Jin, "Noisy low-tubal-rank tensor completion," *Neurocomputing*, vol. 330, pp. 267–279, Nov. 2019.

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

[32] L. Yuan, Q. Zhao, L. Gui, and J. Cao, "High-order tensor completion via gradient-based optimization under tensor train format," *Signal Process.: Image Commun.*, vol. 73, pp. 53–61, Apr. 2019.

[33] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma, *Robust Statistics for Signal Processing*. Cambridge, U.K.: Cambridge Univ. Press, 2018.

[34] W. Wang, V. Aggarwal, and S. Aeron, "Efficient low rank tensor ring completion," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 5697–5705.

[35] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma, "Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts," *IEEE Signal Process. Mag.*, vol. 29, no. 4, pp. 61–80, Jul. 2012.

[36] V. S. Bhadouria and D. Ghoshal, "A study on genetic expression programming based approach for impulse noise reduction in images," *Signal Image Video Process.*, vol. 10, no. 3, pp. 575–584, Mar. 2016.

[37] X. Lin, L. Huang, and W. Sun, "ℓ_p-PARAFAC for joint DOD and DOA estimation in bistatic MIMO radar," in *Proc. CIE Int. Conf. Radar (RADAR)*, Guangzhou, China, Oct. 2016, pp. 1–5.

[38] W. Sun, X. Lin, H. C. So, L. Huang, and Q. Li, "Iteratively reweighted tensor SVD for robust multi-dimensional harmonic retrieval," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Apr. 2016, pp. 4318–4322.

[39] P. Comon, "Tensors: A brief introduction," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 44–53, Apr. 2014.

[40] S. Holtz, T. Rohwedder, and R. Schneider, “On manifolds of tensors of fixed TT-rank,” *Numer. Math.*, vol. 120, no. 4, pp. 701–731, Apr. 2012.

[41] H. Eckblom, “Calculation of linear best ℓ_p approximation,” *BIT*, vol. 13, pp. 292–300, Jan. 1973.

[42] S. Jawad, I. M. Qureshi, Y. Deng, and K. Kadir. “Reconstruction of sparse signals and compressively sampled images based on smooth ℓ_1-norm approximation.” *J. Signal. Process. Syst.*, vol. 88, no. 3, pp. 333–344, Sep. 2017.

[43] J.-F. Cai, E. J. Candès, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010.

[44] S. Ma, D. Goldfarb, and L. Chen, “Fixed point and bregman iterative methods for matrix rank minimization,” *Math. Program.*, vol. 128, no. 1, pp. 321–353, Sep. 2011.

[45] Z. Wen, W. Yin, and Y. Zhang, “Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm,” *Math. Program. Comput.*, vol. 4, no. 4, pp. 333–361, Dec. 2012.

[46] P. Jain, P. Netrapalli, and S. Sanghavi, “Low-rank matrix completion using alternating minimization,” in *Proc. 45th Annu. ACM Symp. Theory Comput.*, California, USA, Jun. 2013, pp. 665–674.

[47] E. J. Candès and Y. Plan, “Matrix completion with noise,” *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010.

[48] P. Tseng, “Convergence of a block coordinate descent method for non-differentiable minimization,” *J. Optim. Theory Appl.*, vol. 103, no. 9, pp. 475–494, Jun. 2001.

[49] Y. Xu and W. Yin, “A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion,” *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 1758–1789, Sep. 2013.

[50] S. Yun and K.-C. Toh, “A coordinate gradient descent method for ℓ_1-regularized convex minimization,” *Comput. Optim. Appl.*, vol. 48, no. 2, pp. 273–307, Mar. 2011.

[51] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[52] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[53] C. Chen, B. He, Y. Ye, and X. Yuan, “The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent,” *Math. Program. Ser. A.*, vol. 155, no. 1, pp. 57–79, Jan. 2016.

[54] J. Eckstein, “Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results,” *Rutcor, Res. Rep. RRR*, vol. 32, no. 3, pp. 44–78, Dec. 2012.

[55] A. Eiger, K. Sikorski, and F. Stenger, “A bisection method for systems of nonlinear equations,” *ACM Trans. Math. Softw.*, vol. 10, no. 4, pp. 367–377, Dec. 1984.

[56] C. Gutierrez, F. Gutierrez, and M. C. Rivara, “Complexity of the bisection method,” *Theor. Comput. Sci.*, vol. 382, no. 2, pp. 131–138, Aug. 2007.

[57] W. J. Zeng, and H. C. So, “Outlier-robust matrix completion via ℓ_p-minimization,” *IEEE Trans. Signal Process.*, vol. 66, no. 5, pp. 1125–1140, Mar. 2018.

Xiao Peng Li was born in Hebei, China. He received the B.Eng. degree in electronic science and technology from Yanshan University, Qinhuangdao, China, in 2015, and the M.Sc. degree in 2018 in electronic information engineering from the City University of Hong Kong, Hong Kong, where he is currently working toward the Ph.D. degree with the Department of Electrical Engineering. From 2018 to 2019, he was a Research Assistant with the College of Information Engineering, Shenzhen University, Shenzhen, China. His research interests include robust signal processing and sparse approximation.

Hing Cheung So (Fellow, IEEE) was born in Hong Kong. He received the B.Eng. degree in electronic engineering from the City University of Hong Kong, Hong Kong, in 1990 and the Ph.D. degree in electronic engineering from The Chinese University of Hong Kong, Hong Kong, in 1995. From 1990 to 1991, he was an Electronic Engineer with the Research and Development Division, Everex Systems Engineering Ltd., Hong Kong. During 1995–1996, he was a Postdoctoral Fellow with The Chinese University of Hong Kong. From 1996 to 1999, he was a Research Assistant Professor with the Department of Electronic Engineering, City University of Hong Kong, where he is currently a Professor. His research interests include detection and estimation, fast and adaptive algorithms, multidimensional harmonic retrieval, robust signal processing, source localization, and sparse approximation.

He is on the editorial boards of IEEE SIGNAL PROCESSING MAGAZINE during 2014–2017, IEEE TRANSACTIONS ON SIGNAL PROCESSING during 2010–2014, Signal Processing from 2010–Present, and Digital Signal Processing from 2011–Present. He was also the Lead Guest Editor of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, special issue on ADVANCES IN TIME/FREQUENCY MODULATED ARRAY SIGNAL PROCESSING in 2017. In addition, he was an Elected Member in Signal Processing Theory and Methods Technical Committee during 2011–2016 of the IEEE Signal Processing Society, where he was chair in the awards subcommittee during 2015–2016.