Tensor Recovery Based on A Novel Non-convex Function Minimax Logarithmic Concave Penalty Function
Hongbing Zhang, Xinyi Liu, Chang Liu, Hongtao Fan, Yajing Li, Xinyun Zhu

Abstract—Non-convex relaxation methods have been widely used in tensor recovery problems, and compared with convex relaxation methods, can achieve better recovery results. In this paper, a new non-convex function, Minimax Logarithmic Concave Penalty (MLCP) function, is proposed, and some of its intrinsic properties are analyzed, among which it is interesting to find that the Logarithmic function is an upper bound of the MLCP function. The proposed function is generalized to tensor cases, yielding tensor MLCP and weighted tensor $L_\gamma$ norm. Consider that its explicit solution cannot be obtained when applying it directly to the tensor recovery problem. Therefore, the corresponding equivalence theorems to solve such problem are given, namely, tensor equivalent MLCP theorem and equivalent weighted tensor $L_\gamma$-norm theorem. In addition, we propose two EMLCP-based models for classic tensor recovery problems, namely low-rank tensor completion (LRTC) and tensor robust principal component analysis (TRPCA), and design proximal alternate linearization minimization (PALM) algorithms to solve them individually. Furthermore, based on the Kurdyka–Łojasiewicz property, it is proved that the solution sequence of the proposed algorithm has finite length and converges to the critical point globally. Finally, Extensive experiments show that the proposed algorithm achieve good results, and it is confirmed that the MLCP function is indeed better than the Logarithmic function in the minimization problem, which is consistent with the analysis of theoretical properties.

Index Terms—Minimax logarithmic concave penalty (MLCP), equivalent weighted Tensor $L_\gamma$-norm, low-rank tensor completion (LRTC), tensor robust principal component analysis (TRPCA).

I. INTRODUCTION

DATA structures become more complex, and the processing required by many applications becomes more difficult as the dimensionality of the data increases. As a representation of multi-dimensional data, tensors have played an important role in many high-dimensional data applications in recent years, such as color image/video (CI/CV) processing [1, 2, 3, 4], hyperspectral/multispectral image (HSI/MSI) processing [5, 6, 7, 8], magnetic resonance imaging (MRI) data recovery [9, 10, 11, 12], video foreground and background subtraction [13, 14, 15, 16], video rain stripe removal [17, 18], and signal reconstruction [19, 20].

These practical application problems above can be transformed into tensor recovery problems. For different observation data, the tensor recovery problem can usually be modeled as a low-rank tensor completion (LRTC) problem and a tensor robust principal component analysis (TRPCA) problem. Their corresponding models are as follows:

$$\min_{\mathbf{Z}} \text{rank}(\mathbf{Z}) \quad \text{s.t.} P_{\Omega}(\mathbf{T}) = P_{\Omega}(\mathbf{Z})$$

$$\min_{\mathbf{Z}, \mathbf{E}} \text{rank}(\mathbf{Z}) + \tau \|\mathbf{E}\|_1 \quad \text{s.t.} \mathbf{T} = \mathbf{Z} + \mathbf{E},$$

where $\mathbf{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ is the observation; $\mathbf{Z}$ is the initial tensor; $\mathbf{E}$ is a sparsity tensor; $P_{\Omega}(\mathbf{Z})$ is a projection operator that keeps the entries of $\mathbf{Z}$ in $\Omega$ and sets all others to zero. Let

$$\Phi_{\Omega}(\mathbf{Z}) := \begin{cases} 0, & \text{if } \mathbf{Z} \in \mathbf{G}, \\ \infty, & \text{otherwise} \end{cases}$$

where $\mathbf{G} := \{ \mathbf{Z} \in \mathbb{R}^{d_1 \times d_2 \times d_3} : P_{\Omega}(\mathbf{Z} - \mathbf{T}) = 0 \}$.

It is not difficult to find that (1) and (2) are the problem of solving tensor rank minimization. As we all know, the most popular tensor recovery method is nuclear norm minimization. However, the definition of the rank of a tensor is not unique, different tensor rank and corresponding nuclear norm can be induced based on different tensor decomposition. The CANDECOMP/PARAFAC (CP) rank is equal to the smallest number of rank-1 tensors to achieve CP decomposition [21], but generally NP-hard to estimate accurately [22]. Another popular rank is the Tucker rank derived from the Tucker decomposition [23], which is defined as a vector whose $i$th element corresponds to the rank of the mode $i$ unfolding matrix of tensor. Liu et al. [24] first proposed sum of nuclear norms (SNN) as a convex surrogate of Tucker rank, which significantly facilitated the development of the tensor recovery problem. But the SNN is not compact convex relaxation of Tucker rank, and this matrixing technique cannot fully exploit tensor structure information [25]. Furthermore, tensor tubal rank and multi-rank are obtained from tensor singular value decomposition (t-SVD) [26]. Since there is no need tensor matrixization in the calculation process, this allows better utilization of the tensor’s internal structural information. Many multidimensional data in the real world can be well approximated by low-rank tensors, due to the fact that the singular values of the corresponding tensors are relatively

This work was supported by the National Natural Science Foundation of China (Nos. 11701456, 11801452, 11571004), Fundamental Research Project of Natural Science in Shaanxi Province General Project (Youth) (Nos. 2019JQ-415, 2019JQ-196), the Initial Foundation for Scientific Research of Northwest A&F University (Nos.2452017219, 2452018017), and Innovation and Entrepreneurship Training Program for College Students of Shaanxi Province (201910712132).

H. Zhang, X. Liu, C. Liu, H. Fan and Y. Li are with Department of information and Computing Science, College of Science, Northwest A&F University, Yangling, Shaanxi 712100, China (e-mail: zhanghh@nwafu.edu.cn; Lxy601@163.com; changh_0923@163.com; fahnht17@nwafu.edu.cn; Hliya-jing@163.com)

X. Zhu is with the Department of Mathematics, University of Texas of the Permian Basin, Odessa, TX 79762, USA (e-mail: zhu_x@utpb.edu).
small, while a few large ones contain the main information. On this basis, the tensor nuclear norm (TNN) has been proposed as a convex relaxation of tubal-rank \([27]\). Recently, Zheng et al. \([28]\) proposed a new form of rank (\(N\)-tubal rank) based on tubal rank, which adopts a new unfold method of higher-order tensors into third-order tensors in various directions. Benefiting from this, t-SVD can be applied to higher-order situations by solving third-order tensors of various directions forms. This approach not only enables t-SVD to be applied to higher order cases, but also makes good use of the properties of tensor tubal rank. In view of the excellent properties of \(N\)-tubal rank, we will use \(N\)-tubal rank to construct the model in this paper.

However, although nuclear norm relaxation is becoming a popular solution to the low rank tensor recovery problem, it still suffers from some drawbacks. TNN is a convex relaxation approximation of tensor tubal rank, and there is still a certain distance from tensor tubal rank minimization, which usually leads to the solution of the optimization problem being suboptimal solution to the original problem. Recently, to break the limitation of biased estimation of convex relaxation methods, some non-convex relaxation strategies have been adopted to solve the tensor recovery problem. Non-convex methods are able to penalize larger singular values less and smaller singular values more. In \([29]\), a t-Schatten-p norm was proposed to approximate tensor tubal rank by extending the Schatten-p norm. Another non-convex approach to approximating the tensor tubal rank is by transforming each element as a sum of t-TNNs with the Laplace function \([30]\). Besides, Logarithmic function \([31]\), MCP function \([32]\), \([33]\), \([34]\), SCAD function \([35]\) are also applied to carry out non-convex relaxation. To further explore the superiority of non-convex functions and improve the accuracy of tensor recovery, we propose a new non-convex function in this paper, namely the Minimax Logarithmic Concave Penalty (MLCP) function. Interestingly, it is found that the Logarithmic function is an upper bound of the MLCP function. The proposed function is then generalized to higher dimensional cases, yielding vector MLCP, matrix MLCP, tensor MLCP, and weighted tensor \(L_\gamma\)-norm. However, when the proposed function is directly applied to the tensor recovery problem, the explicit solution cannot be obtained, which is very unfavorable for the solution of the algorithm. Therefore, we further put forward the corresponding equivalent theorems, namely vector equivalent MLCP theorem, matrix equivalent MLCP theorem, tensor equivalent MLCP theorem, and equivalent weighted tensor \(L_\gamma\)-norm theorem. The properties of the tensor EMLCP and the equivalent weighted tensor \(L_\gamma\)-norm are analyzed. Furthermore, the proximal operator for the equivalent weighted tensor \(L_\gamma\)-norm is given, so as to make the tensor recovery model easier to solve.

Second, we construct corresponding EMLCP-based models for two typical problems of tensor recovery, and design a Proximal Alternating Linearization Minimization Algorithm (PALM) to solve these two EMLCP-based models. In particular, we adopt a model that removes mixed noise for the TRPCA problem, which is more realistic. Furthermore, based on the Kurdyka-Łojasiewicz property, it is proved that the solution sequence of the proposed algorithm has finite length and converges to the critical point globally.

Third, we conduct experiments on both LRTC and TRPCA using real data. The LRTC experiments on HSI, MRI, CV and the TRPCA experiments on HSI demonstrate the effectiveness of our proposed new non-convex relaxation method. This method yields better results than the Logarithmic relaxation method, which is consistent with our theoretical analysis.

The summary of this article is as follows: In Section II, some preliminary knowledge and background are given. The definitions and theorems of the MLCP function are presented in Section III. In Section IV, we establish the EMLCP-based models and algorithms. In Section V, we give the theoretical convergence analysis of the proposed algorithms. The results of extensive experiments and discussion are presented in Section VI. Conclusions are drawn in Section VII.

II. Preliminaries

A. Tensor Notations and Definitions

In this section, we give some basic notations and briefly introduce some definitions used throughout the paper. Generally, a lowercase letter and an uppercase letter denote a vector and a matrix, respectively. An \(N\)-th order tensor is denoted by a calligraphic upper case letter \(\mathbf{Z} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}\) and its \(i_1, i_2, \ldots, i_N\)-th element is \(z_{i_1, i_2, \ldots, i_N}\). The Frobenius norm of a tensor is defined as \(\|\mathbf{Z}\|_F = \sqrt{(\sum_{i_1, i_2, \ldots, i_N} z_{i_1, i_2, \ldots, i_N}^2)^{1/2}}\). For a three order tensor \(\mathbf{Z} \in \mathbb{R}^{I_1 \times I_2 \times I_3}\), we use \(\tilde{\mathbf{Z}}\) to denote the discrete Fourier transformation (DFT) along each tubal of \(\mathbf{Z}\), i.e., \(\tilde{\mathbf{Z}} = \text{fft}(\mathbf{Z}, [\cdot, \cdot, 3])\). The inverse DFT is computed by command \(\text{ifft}\) satisfying \(\mathbf{Z} = \text{ifft}(\tilde{\mathbf{Z}}, [\cdot, \cdot, 3])\). More often, the frontal slice \(\mathbf{Z}(\cdot, \cdot, i)\) is denoted compactly as \(\mathbf{Z}^{(i)}\).

Definition 1 (Mode-\(k_1\)\(k_2\) slices \([28]\)) For an \(N\)-th order tensor \(\mathbf{Z} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}\), its mode-\(k_1\)\(k_2\) slices \((\mathbf{Z}^{(k_1,k_2)}, 1 \leq k_1 < k_2 \leq N, k_1, k_2 \in \mathbb{Z})\) are two-dimensional sections, defined by fixing all but the mode-\(k_1\) and the mode-\(k_2\) indexes.

Definition 2 (Tensor Mode-\(k_1\)\(k_2\) Unfolding and Folding \([28]\)) For an \(N\)-th order tensor \(\mathbf{Z} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}\), its mode-\(k_1\)\(k_2\) unfolding is a three order tensor denoted by \(\mathbf{Z}_{(k_1,k_2)} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}\).
Mathematically, the \( k \)th frontal slices of which are the lexicographic orderings of the mode-\( k_1k_2 \) slices of \( Z \). The mode-\( k_1k_2 \) unfolding operator and its inverse operation are respectively denoted as \( Z_{(k_1k_2)} := t - unfold(Z, k_1, k_2) \) and \( Z := t - fold(Z_{(k_1k_2)}, k_1, k_2) \).

For a three order tensor \( Z \in \mathbb{R}^{l_1 \times l_2 \times l_3} \), the block circulation operation is defined as
\[
bcirc(Z) := \begin{pmatrix}
Z^{(1)} & Z^{(3)} & \cdots & Z^{(2)} \\
Z^{(2)} & Z^{(1)} & \cdots & Z^{(3)} \\
\vdots & \vdots & \ddots & \vdots \\
Z^{(l_3)} & Z^{(l_3-1)} & \cdots & Z^{(1)}
\end{pmatrix} \in \mathbb{R}^{l_1 \times l_3 \times l_2}.
\]

The block diagonalization operation and its inverse operation are defined as
\[
bdig(Z) := \begin{pmatrix}
Z^{(1)} \\
Z^{(2)} \\
\vdots \\
Z^{(l_3)}
\end{pmatrix} \in \mathbb{R}^{l_1 \times l_3 \times l_2},
\]
\[
bdfold(bdig(Z)) := Z.
\]

The vectorization operation and its inverse operation are defined as
\[
bvec(Z) := \begin{pmatrix}
Z^{(1)} \\
Z^{(2)} \\
\vdots \\
Z^{(l_3)}
\end{pmatrix} \in \mathbb{R}^{l_1 \times l_3 \times l_2},
\]
\[
bvfold(bvec(Z)) := Z.
\]

**Definition 3 (t-product [26]):** Let \( A \in \mathbb{R}^{l_1 \times l_2 \times l_3} \) and \( B \in \mathbb{R}^{l_3 \times l_4 \times l_5} \). Then the t-product \( A \ast B \) is defined to be a tensor of size \( l_1 \times l_4 \times l_5 \),
\[
A \ast B := bvfold(bcirc(A)bvec(B)).
\]

Since that circular convolution in the spatial domain is equivalent to multiplication in the Fourier domain, the t-product between two tensors \( C = A \ast B \) is equivalent to
\[
\hat{C} = bdfold(bdig(\hat{A})bdig(\hat{B})).
\]

**Definition 4 (Tensor conjugate transpose [26]):** The conjugate transpose of a tensor \( A \in \mathbb{C}^{l_1 \times l_2 \times l_3} \) is the tensor \( A^H \in \mathbb{C}^{l_3 \times l_2 \times l_1} \) obtained by conjugate transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through 1.

**Definition 5 (identity tensor [26]):** The identity tensor \( I \in \mathbb{R}^{l_1 \times l_1 \times l_1} \) is the tensor whose first frontal slice is the \( I_1 \times I_1 \) identity matrix, and whose other frontal slices are all zeros.

It is easy to see that \( bcirc(I) \) is the \( I_1I_2 \times I_1I_2 \) identity matrix. So it is easy to get \( A \ast I = A \) and \( I \ast A = A \).

**Definition 6 (orthogonal tensor [26]):** A tensor \( Q \in \mathbb{R}^{l_1 \times l_1 \times l_1} \) is orthogonal if it satisfies
\[
Q \ast Q^H = Q^H \ast Q = I.
\]
where \( \partial f(z) \) denotes the Fréchet subdifferen- tial of \( f \) at \( z \in \text{dom}(f) \), which is the set of all \( y \) satisfying
\[
\lim \inf_{x \to z, x \neq z} \frac{f(x) - f(z) - y \cdot (x - z)}{\|x - z\|} \geq 0.
\] (8)

For any \( Z \in \mathbb{R} \), the distance from \( Z \) to \( \mathcal{F} \) is defined by
\[
dist(Z, \mathcal{F}) := \inf\{\|Z - Y\|_F : Y \in \mathcal{F}\},
\]
where \( \mathcal{F} \) is a subset of \( \mathbb{R} \). Next, we recall the Kurdyka–Łojasiewicz (KL) property, which plays a pivotal role in the analysis of the convergence of proximal alternating linearized minimization (PALM) algorithm for the nonconvex problems.

Definition 13 (KL function [40]): Let \( f : \mathbb{R}^n \to (-\infty, +\infty] \) be a proper and lower semicontinuous function.

(a): The function \( f \) is said to have the KL property at \( z \in \text{dom}(\partial f) \) if there exist \( \eta \in (0, +\infty] \), a neighborhood \( \mathcal{F} \) of \( z \) and a continuous concave function \( \phi : [0, \eta) \to [0, +\infty) \) such that:

\[
\text{(a) } \phi(0) = 0; \quad \text{(b) } \phi \text{ is continuously differentiable on } (0, \eta), \text{ and continuous at } 0; \quad \text{(c) } \phi'(s) > 0 \text{ for all } s \in (0, \eta); \quad \text{(d) for all } y \in \mathcal{F} \cap \{y \in \mathbb{R}^n : f(z) < f(y) < f(z) + \eta\}, \text{the following KL inequality holds:}
\]
\[
\phi'(f(y) - f(z)) \cdot \text{dist}(0, \partial f(y)) \geq 1.
\]

(b): If \( f \) satisfies the KL property at each point of \( \text{dom}(\partial f) \), then \( f \) is called a KL function.

III. MINIMAX LOGARITHMIC CONCAVE PENALTY (MLCP) FUNCTION AND EQUIVALENT MINIMAX LOGARITHMIC CONCAVE PENALTY (EMLP)

In this section, we first define the definition of the Minimax Logarithmic Concave Penalty (MLCP) function.

Definition 14 (Minimax Logarithmic Concave Penalty (MLCP) function): Let \( \lambda > 0, \gamma > 0, \varepsilon > 0 \). The MLCP function \( f_{L,\gamma,\lambda} : \mathbb{R} \to \mathbb{R}_{\geq 0} \) is defined as
\[
f_{L,\gamma,\lambda}(z) = \begin{cases}
\lambda \log \left( \frac{|z|}{\varepsilon} + 1 \right) - \log \left( \frac{|z| + 1}{2\varepsilon} \right), & |z| \leq \varepsilon e^\gamma \lambda - \varepsilon, \\
\frac{\varepsilon L}{2}, & |z| > \varepsilon e^\gamma \lambda - \varepsilon.
\end{cases}
\] (9)

The MLCP function is a symmetric function, so we only discuss its functional properties on \([0, +\infty)\).

Proposition 1: The MLCP function defined in (9) satisfies the following properties:

(a): \( f_{L,\gamma,\lambda}(z) \) is continuous, smooth and
\[
f_{L,\gamma,\lambda}(0) = 0, \quad \lim_{z \to +\infty} \frac{f_{L,\gamma,\lambda}(z)}{z} = 0;
\]

(b): \( f_{L,\gamma,\lambda}(z) \) is monotonically non-decreasing and concave on \([0, +\infty)\);

(c): \( f'_{L,\gamma,\lambda}(z) \) is non-negativity and monotonicity non-increasing on \([0, +\infty)\). Moreover, it is Lipschitz bounded, i.e., there exists constant \( L(f) \) such that
\[
|f'_{L,\gamma,\lambda}(x) - f'_{L,\gamma,\lambda}(y)| \leq L(f) |x - y|;
\]

(d): Especially, for the MLCP function, it is increasing in parameter \( \gamma \), and
\[
\lim_{\gamma \to +\infty} f_{L,\gamma,\lambda}(z) = \lambda \log \left( \frac{|z|}{\varepsilon} + 1 \right). \] (10)

Proof: The proof is provided in Appendix A.

Definition 15 (Vector MLCP): Let \( z \in \mathbb{R}^n \) and \( \lambda > 0, \gamma > 0, \varepsilon > 0 \). The vector MLCP \( f_{L,\gamma,\lambda} : \mathbb{R}^n \to \mathbb{R}_{\geq 0} \) is defined as
\[
f_{L,\gamma,\lambda}(z) = \sum_{i=1}^n f_{L,\gamma,\lambda}(z_i),
\] (11)

where \( z_i \) denotes the \( i \)th entry of the vector \( z \) and \( f_{L,\gamma,\lambda}(z_i) \) is defined in (9).

Definition 16 (Matrix MLCP): Let \( Z \in \mathbb{R}^{m \times n} \) and \( \lambda > 0, \gamma > 0, \varepsilon > 0 \). The matrix MLCP \( f_{L,\gamma,\lambda} : \mathbb{R}^{m \times n} \to \mathbb{R}_{\geq 0} \) is defined as
\[
f_{L,\gamma,\lambda}(Z) = \sum_{i=1}^m \sum_{j=1}^n f_{L,\gamma,\lambda}(Z_{ij}),
\] (12)

where \( Z_{ij} \) denotes the \((i, j)\) element of \( Z \), and \( f_{L,\gamma,\lambda} \) is the same as in (9).

Definition 17 (Tensor MLCP): Let \( Z \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_n} \) and \( \lambda > 0, \gamma > 0, \varepsilon > 0, \bar{\lambda} \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_n} \). The tensor MLCP \( f_{L,\gamma,\lambda} : \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_n} \to \mathbb{R}_{\geq 0} \) is defined as
\[
f_{L,\gamma,\lambda}(\bar{Z}) = \sum_{i_1=1}^{l_1} \cdots \sum_{i_n=1}^{l_n} f_{L,\gamma,\lambda}(\bar{Z}_{i_1,i_2,\ldots,i_n}(\bar{Z}_{i_1,i_2,\ldots,i_n}))
\]

where \( \bar{Y}_{i_1,i_2,\ldots,i_n} \) denotes the \((i_1, i_2, \ldots, i_n)\)-th element of \( Y \), and \( h_{L,\gamma,\lambda} \) is defined in (9).

Definition 18 (Matrix Ly-norm): The Ly norm of a rank-r matrix \( Z \in \mathbb{R}^{m \times n} \), denoted by \( ||Z||_{L,\gamma,\lambda} \), is defined in terms of the singular values \( \sigma_i, i = 1, 2, \ldots, r \) as follows:
\[
||Z||_{L,\gamma,\lambda} := f_{L,\gamma,\lambda}(\sigma) = \sum_{i=1}^r f_{L,\gamma,\lambda}(\sigma_i),
\] (14)

where \( \sigma \) is singular value vector of matrix \( Z \).

Similarly, the weighted matrix Ly-norm is a generalization of weighted MLCP for matrix and is defined as follows.

Definition 19 (Weighted matrix Ly-norm): The weighted matrix Ly-norm of \( Z \in \mathbb{R}^{m \times n} \), denoted by \( ||Z||_{L,\gamma,\lambda} \), is defined as follows:
\[
||Z||_{L,\gamma,\lambda} \leq f_{L,\gamma,\lambda}(\sigma) = \sum_{i=1}^r f_{L,\gamma,\lambda}(\sigma_i). \]
(15)

where \( r = \min(m, n) \) denotes the maximum rank of \( Z \).

Definition 20 (Weighted tensor Ly-norm): The weighted tensor Ly-norm of \( Z \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_n} \), denoted by \( ||Z||_{L,\gamma,\lambda} \), is defined as follows:
\[
||Z||_{L,\gamma,\lambda} \leq \sum_{i_1=1}^{l_1} \cdots \sum_{i_n=1}^{l_n} f_{L,\gamma,\lambda}(\sigma_j(\bar{Z}(i))).
\] (16)

where \( R = \min(l_1, l_2) \).

Further, we convert \( \lambda \) from a constant to a variable, for which we propose some equivalent MLCP theorems.

Theorem 2 (Scalar EMCP): Let \( \lambda > 0, \gamma > 0, \varepsilon > 0 \) and \( z \in \mathbb{R} \). The scalar EMCP \( f_{L,\gamma,\lambda} : \mathbb{R} \to \mathbb{R}_{\geq 0} \) is the solution of the following optimization problem:
\[
f_{L,\gamma,\lambda}(z) = \min_{\omega \in \mathbb{R}_{\geq 0}} \{ \omega \log \left( \frac{|z|}{\varepsilon} + 1 \right) + \frac{\gamma}{2} (\omega - \lambda)^2 \}. \] (17)

Proof: The proof is provided in Appendix B.
Theorem 3 (Vector EMLCP): Let $\psi > 0$, $\phi > 0$, $\omega \in \mathbb{R}_{\geq 0}^n$, $\lambda \in \mathbb{R}_{\geq 0}^m$, and $z \in \mathbb{R}^n$. The vector MCP is the solution of the following optimization problem:

$$f_{L, \gamma, \phi}(z) = \min_{\omega \in \mathbb{R}_{\geq 0}^n} \left\{ \|z\|_{L, \omega} + \frac{\gamma}{2} \|\omega - \lambda\|_2^2 \right\},$$

(18)

where $\|z\|_{L, \omega}$ is defined as

$$\|z\|_{L, \omega} = \sum_{i=1}^n \omega_i \log\left(\frac{\|z_i\|}{\epsilon}\right) + 1, \omega_i \geq 0,$$

and $\{\omega_i, i = 1, 2, \ldots, n\}$ denote the weights.

Proof: The proof is provided in Appendix C.

Theorem 4 (Matrix EMLCP): Let $\phi > 0$, $\beta > 0$, $\Omega \in \mathbb{R}_{\geq 0}^{m \times n}$, and $Z \in \mathbb{R}_{\geq 0}^{m \times n}$. The matrix MCP is the solution of the following optimization problem:

$$f_{L, \gamma, \phi}(Z) = \min_{\Omega \in \mathbb{R}_{\geq 0}^{m \times n}} \left\{ \|Z\|_{L, \Omega} + \frac{\gamma}{2} \|\Omega - \Lambda\|_2^2 \right\},$$

(19)

where $\|Z\|_{L, \Omega}$ is defined as

$$\|Z\|_{L, \Omega} = \sum_{i=1}^m \sum_{j=1}^n \Omega_{ij} \log\left(\frac{\|Z_{ij}\|}{\epsilon}\right) + 1,$$

where $\{\Omega_{ij} \geq 0, i = 1, 2, \ldots, m, j = 1, 2, \ldots, n\}$ denote the weights.

Proof: The proof is provided in Appendix D.

Theorem 5 (Tensor EMLCP): Let $\phi > 0$, $\beta > 0$, $\Omega \in \mathbb{R}_{\geq 0}^{I_1 \times I_2 \times \cdots \times I_N}$ and $Z \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. The tensor MCP is the solution of the following optimization problem:

$$f_{L, \gamma, \phi}(Z) = \min_{\Omega \in \mathbb{R}_{\geq 0}^{I_1 \times I_2 \times \cdots \times I_N}} \left\{ \|Z\|_{L, \Omega} + \frac{\gamma}{2} \|\Omega - \Lambda\|_2^2 \right\},$$

(20)

where $\|Z\|_{L, \Omega}$ is defined as

$$\|Z\|_{L, \Omega} = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \Omega_{i_1i_2\cdots i_N} \log\left(\frac{\|Z_{i_1i_2\cdots i_N}\|}{\epsilon}\right) + 1,$$

where $W$ is weight tensor, and $t = i_1, i_2, \ldots, i_N$.

Proof: The proof is provided in Appendix E.

Remark 1: As the order of the tensor decreases, tensor EMLCP can degenerate into the form of matrix EMLCP, vector EMLCP, and scalar EMLCP respectively.

Theorem 6 (Equivalent weighted tensor $L_1$-norm): Consider a rank-$r$ matrix $Z \in \mathbb{R}^{m \times n}$ with the SVD: $Z = Udiag(\sigma)V^T$, where $\sigma = [\sigma_1, \sigma_2, \ldots, \sigma_r]^T$. Let $\Omega, \Lambda \in \mathbb{R}_{\geq 0}^r$, and $\gamma > 0, \phi > 0$. The matrix $L_1$-norm is obtained equivalently as

$$\|Z\|_{L, \gamma, \phi, \Omega} = \min_{\Omega} \left\{ \|Z\|_{L, \Omega} + \frac{\gamma}{2} \|\Omega - \Lambda\|_2^2 \right\},$$

(22)

where $\|Z\|_{L, \Omega} = \sum_{i=1}^r \Omega_i \log\left(\frac{\|Z_i\|}{\epsilon}\right) + 1$.

Proof: The proof is provided in Appendix F.

Theorem 7 (Equivalent weighted tensor $L_2$-norm): For a third-order tensor $Z \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, its SVD is decomposed into $Z = U \ast S \ast V$, where $S \in \mathbb{R}^{R \times R \times I_3}$ and $R = \min\{I_1, I_2\}$. Let $W, \Lambda \in \mathbb{R}^{R \times I_3}$, and $\gamma > 0, \phi > 0$. The weighted tensor $L_2$-norm is obtained equivalently as

$$\|Z\|_{L, \gamma, \phi, \Omega} = \min_{W} \left\{ \|Z\|_{L, W} + \frac{\gamma}{2} \|W - \Lambda\|_F^2 \right\},$$

(23)

where

$$\|Z\|_{L, W} := \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \sum_{i_3=1}^{I_3} W_{i_1i_2i_3} \log\left(\frac{\|Z_{i_1i_2i_3}\|}{\epsilon}\right) + 1.$$
Algorithm 1 EMLCPTC

**Input:** An incomplete tensor \( T \), the index set of the known elements \( \Omega \), convergence criteria \( \epsilon \), maximum iteration number \( \tau \).

**Initialization:** \( \mathcal{Z}^0 = \mathcal{T}_\Omega, \mathcal{M}^0_{k_1k_2} = \mathcal{X}^0, \mu^0_{k_1k_2} > 0, \rho > 0, \tau > 1 \).

1. while not converged and \( k < K \) do
   2. Updating \( W^k_{k_1k_2} \) via (32);
   3. Updating \( \mathcal{M}^k_{k_1k_2} \) via (33);
   4. Updating \( \mathcal{Z}^k \) via (36);
   5. Updating the multipliers \( Q^k_{k_1k_2} \) via (37);
   6. \( \mu^k_{k_1k_2} = \tau \mu^{k-1}_{k_1k_2}, \rho = \tau \rho, k = k + 1; \)
   7. Check the convergence conditions \( \| \mathcal{Z}^{k+1} - \mathcal{Z}^k \|_\infty \leq \epsilon \).
8. end while

9. return \( \mathcal{Z}^{k+1} \).

**Output:** Completed tensor \( \mathcal{Z} = \mathcal{Z}^{k+1} \).

is given by
\[
S_{L,Y,\bar{\Lambda}} = \begin{cases} 
W_{j,i} = \max\{\bar{\Lambda}_{j,i} - \frac{\log (\sigma_j(\bar{Y}^{(i)})) + 1}{\gamma}, 0\}, \\
\bar{L} = \mathcal{U} + S_1 + \mathcal{V}^H,
\end{cases}
\]
where \( \mathcal{U} \) and \( \mathcal{V} \) are derived from the t-SVD of \( \mathcal{Y} = \mathcal{U} \ast S_2 \ast \mathcal{V}^H \). More importantly, the \( i \)th front slice of DFT of \( S_1 \) and \( S_2 \), i.e., \( S_1^{(i)} = \sigma(\bar{L}^{(i)}) \) and \( S_2^{(i)} = \sigma(\bar{Y}^{(i)}) \), has the following relationship
\[
\sigma_j(\bar{L}^{(i)}) = \begin{cases} 
0, & \text{if } \sigma_j(\bar{Y}^{(i)}) \leq 2\sqrt{\alpha} - \epsilon, \\
\frac{l_1}{\rho}, & \text{if } \sigma_j(\bar{Y}^{(i)}) > 2\sqrt{\alpha} - \epsilon,
\end{cases}
\]
where \( l_1 = \sigma_j(\bar{Y}^{(i)}) - \epsilon, l_2 = \sqrt{(\sigma_j(\bar{Y}^{(i)}) + \epsilon)^2 - 4\alpha}, \alpha = W_{j,i}/\rho \).

**Proof:** The proof is provided in Appendix J.

IV. EMLCPTC-BASED MODELS AND SOLVING ALGORITHMS

In this section, we apply the EMLCPTC to low rank tensor completion (LRTC) and tensor robust principal component analysis (TRPCA) and propose the EMLCPTC-based models with proximal alternating linearized minimization algorithms.

A. EMLCPTC-based LRTC model

Tensor completion aims at estimating the missing elements from an incomplete observation tensor. Considering an \( N \)-order tensor \( \mathcal{Z} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \), the proposed EMLCPTC-based LRTC model is formulated as follow
\[
\min_{\mathcal{Z},W} \sum_{1 \leq k_1 < k_2 < N} \beta_{k_1k_2} (\|\mathcal{Z}(k_1k_2)\|_LW + \frac{\gamma}{2}\|W(k_1k_2) - \bar{\Lambda}\|_F^2) + \Phi_{\Omega}(\mathcal{Z}) \tag{27}
\]
where \( \mathcal{Z} \) is the reconstructed tensor and \( T \) is the observed tensor, \( \Omega \) is the index set for the known entries, and \( \Phi_{\Omega}(\mathcal{Z}) \) is a projection operator that keeps the entries of \( \mathcal{Z} \) in \( \Omega \) and sets all others to zero, \( \beta_{k_1k_2} \geq 0 \) (\( 1 \leq k_1 < k_2 \leq N, k_1, k_2 \in \mathbb{Z} \)) and \( \sum_{1 \leq k_1 < k_2 < N} \beta_{k_1k_2} = 1 \). Let
\[
\Phi_{\Omega}(\mathcal{Z}) = \begin{cases} 
0, & \text{if } \mathcal{Z} \in \Omega, \\
\infty, & \text{otherwise}
\end{cases}
\]

where \( \mathbb{G} := \{ \mathcal{Z} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}, \mathcal{P}_{\Omega}(\mathcal{Z} - T) = 0 \} \).

Next, we exploit the PALM to solve (27). We first introduce auxiliary variables \( \mathcal{M}_{k_1k_2} \), and then rewrite (27) as the following equivalent constrained problem:
\[
\min_{\mathcal{Z},W} \sum_{1 \leq k_1 < k_2 < N} \beta_{k_1k_2} (\|\mathcal{M}(k_1k_2)\|_LW + \frac{\gamma}{2}\|W(k_1k_2) - \bar{\Lambda}\|_F^2) + \Phi_{\Omega}(\mathcal{Z}) \tag{29}
\]
subject to
\[
\mathcal{Z} = \mathcal{M}_{k_1k_2}, 1 \leq k_1 < k_2 \leq N, k_1, k_2 \in \mathbb{Z}.
\]

The augmented Lagrangian function of (29) can be expressed in the following concise form:
\[
LAG(\mathcal{Z}, M, W, \bar{\Lambda}, Q) = \sum_{1 \leq k_1 < k_2 < N} \beta_{k_1k_2} (\|\mathcal{M}(k_1k_2)\|_LW + \frac{\gamma}{2}\|W(k_1k_2) - \bar{\Lambda}\|_F^2)
+ \Phi_{\Omega}(\mathcal{Z}) + \frac{\mu_{k_1k_2}}{2} \|\mathcal{M} - \mathcal{M}_{k_1k_2} - \frac{Q_{k_1k_2}}{\mu_{k_1k_2}}\|_F^2,
= H_1(M, W) + H_2(W, \bar{\Lambda}) + \Phi_{\Omega}(\mathcal{Z}) + H_3(M, \mathcal{Z}), \tag{30}
\]
where \( Q_{k_1k_2} \) (\( 1 \leq k_1 < k_2 \leq N \)) are the Lagrange multipliers, \( \mu_{k_1k_2} \) are positive scalars. For the sake of convenience, we denote the variable updated by the iteration as \( (\cdot)^{\ast} \), the last iteration result as \( (\cdot)^{\ast\ast} \), and omit the specific number of iterations. With the proximal linearization of each subproblem, the PALM algorithm on the four blocks \( (\mathcal{Z}, M, W, \bar{\Lambda}) \) for solving (29) yields the iteration scheme alternatingly as follows:
\[
W_{j,i}^{\ast\ast} = \max(\frac{\gamma \bar{\Lambda}_{j,i}^{\ast\ast} + \rho^{\ast\ast}W_{j,i}^{\ast\ast} - \log (\sigma_j(\bar{Y}^{(i)})) + 1}{\gamma + \rho^{\ast\ast}}, 0), \tag{32}
\]
\[
M^{\ast\ast} = \mathcal{U} \ast S_1 \ast \mathcal{V}^H, \tag{33}
\]
where \( \mathcal{U} \) and \( \mathcal{V} \) are derived from the t-SVD of \( M^{\ast\ast} \).

The update for \( \bar{\Lambda} \) turns out to be straightforward:
\[
\bar{\Lambda}^{\ast\ast} = \min(\frac{\gamma W^{\ast\ast} - \bar{\Lambda}^{\ast\ast}}{2}, \frac{\rho^{\ast\ast}W^{\ast\ast} - \bar{\Lambda}^{\ast\ast}}{2}, 0), \tag{34}
\]
Fixed \( W_{j,i}^{\ast\ast}, M_{k_1k_2}^{\ast\ast}, \bar{\Lambda}_{k_1k_2}^{\ast\ast} \) and \( Q_{k_1k_2}^{\ast\ast} \), the minimization problem of \( \mathcal{Z} \) is as follows:
\[
\min_{\mathcal{Z}} \sum_{1 \leq k_1 < k_2 < N} \frac{\mu_{k_1k_2}}{2} \|\mathcal{M}^{\ast\ast}_{k_1k_2} + \frac{Q_{k_1k_2}^{\ast\ast}}{\mu_{k_1k_2}}\|_F^2 + \Phi_{\Omega}(\mathcal{Z}) + \frac{\rho^{\ast\ast}}{2} ||\mathcal{Z} - \mathcal{Z}^{\ast\ast}||_F^2. \tag{35}
\]
The closed form of $\mathcal{Z}$ can be derived by setting the derivative of (35) to zero. We can now update $\mathcal{Z}$ by the following equation:

$$
\mathcal{Z}^+ = \mathcal{P}_D(T) + \mathcal{P}_D \left( \frac{\sum_{1 \leq k_1 < k_2 \leq N} \mu_{k_1, k_2} M_{k_1 k_2}^* - Q_{k_1 k_2}^* + \rho^* \mathcal{Z}^*}{\sum_{1 \leq k_1 < k_2 \leq N} \mu_{k_1, k_2} + \rho^*} \right).
$$

(36)

Finally, multipliers $Q_{k_1 k_2}$ are updated as follows:

$$
Q_{k_1 k_2}^+ = Q_{k_1 k_2}^* + \mu_{k_1, k_2} (\mathcal{Z}^* - M_{k_1 k_2}^*). 
$$

(37)

The EMLCP-based LRTC model computation is given in Algorithm 1. The main per-iteration cost lies in the update of $M_{k_1 k_2}$, which requires computing t-SVD. The per-iteration complexity is $O(LE(\sum_{1 \leq k_1 < k_2 \leq N} [\log(t(e_{k_1}) + \min(I_{k_1}, I_{k_2})])$, where $LE = \prod_{1 \leq i \leq N} I_i$ and $te_{k_2} = LE/(I_{k_1}I_{k_2})$.

B. EMLCP-based TRPCA model

Tensor robust PCA (TRPCA) aims to recover the tensor from grossly corrupted observations. Using the proposed EMLCP, we can get the following EMLCP-based TRPCA model:

$$
\min_{L, \xi, N} \sum_{1 \leq k_1 < k_2 \leq N} \beta_{k_1 k_2} \left( ||L_{k_1 k_2}||_L W + \frac{\gamma}{2} ||W_{k_1 k_2} - \bar{\Lambda}||_F^2 \right) 
$$

$$
+ \tau_1 ||E||_1 + \tau_2 ||N||_F 
$$

where $T$ is the corrupted observation tensor, $L$ is the low-rank component, $E$ is the sparse noise component, $N$ is the Gaussian noise component, and $\tau_1, \tau_2$ are tuning parameters compromising $L, E$ and $N$. Similarly, we introduce auxiliary variables $\mathcal{G}_{k_1 k_2}$, and then rewrite (38) as the following equivalent constrained problem:

$$
\min_{L, \xi, N} \sum_{1 \leq k_1 < k_2 \leq N} \beta_{k_1 k_2} \left( ||G_{k_1 k_2}||_L W + \frac{\gamma}{2} ||W_{k_1 k_2} - \bar{\Lambda}||_F^2 \right) 
$$

$$
+ \tau_1 ||E||_1 + \tau_2 ||N||_F 
$$

s.t. $T = L + E + N,$

$$
L = \mathcal{G}_{k_1 k_2}, 1 \leq k_1 < k_2 \leq N, k_1, k_2 \in \mathbb{Z}. 
$$

(39)

The augmented Lagrangian function of (39) can be expressed in the following concrete form:

$$
LAG(L, G, W, \bar{\Lambda}, E, N, T) = 
$$

$$
\sum_{1 \leq k_1 < k_2 \leq N} \beta_{k_1 k_2} \left( ||G_{k_1 k_2}||_L W + \frac{\gamma}{2} ||W_{k_1 k_2} - \bar{\Lambda}||_F^2 \right) 
$$

$$
+ \tau_1 ||E||_1 + \tau_2 ||N||_F + \frac{\mu_{k_1 k_2}}{2} ||L - G_{k_1 k_2} + R_{k_1 k_2}||_F^2 
$$

$$
+ \frac{\tau}{2} ||T - L - E - N + \frac{\rho}{\tau}||_F^2 
$$

$$
= H_1(G, W) + H_2(W, \bar{\Lambda}) + H_3(E) 
$$

$$
+ H_4(N) + H_5(L, G) + H_6(L, E, N) 
$$

(40)

where $T$ and $R_{k_1 k_2}$ are the Lagrange multipliers, $\mu_{k_1 k_2}, \tau_1, \tau_2$ and $\tau$ are positive scalars. Similar to EMLCP-based LRTC model, we denote the variable updated by the iteration as $(\cdot)^*$, the last iteration result as $(\cdot)^{+}$, and omit the specific number of iterations. With the proximal linearization of each subproblem, the PALM algorithm on the six blocks $(L, G, W, \bar{\Lambda}, E, N)$ for solving (29) yields the iteration scheme alternatingly as follows:

$$
W^* = \min_W H_1(W) + H_2(W) + \frac{\rho}{\tau} ||W - W^*||_F^2, 
$$

$$
G^* = \min_G H_1(G) + (G - G^*)^T H_5(G^*) + \frac{\rho}{\tau} ||G - G^*||_F^2, 
$$

$$
\bar{\Lambda}^* = \min_{\bar{\Lambda}, \hat{\Lambda}} H_3(\hat{\Lambda}) + H_4(\hat{\Lambda}) + \frac{\rho}{\tau} ||\hat{\Lambda} - \bar{\Lambda}^*||_F^2, 
$$

$$
L^* = \min_L H_2(L) + H_5(L) + \frac{\rho}{\tau} ||L - L^*||_F^2, 
$$

$$
E^* = \min_E H_3(E) + H_4(E) + \frac{\rho}{\tau} ||E - E^*||_F^2, 
$$

$$
N^* = \min_N H_4(N) + H_5(N) + \frac{\rho}{\tau} ||N - N^*||_F^2. 
$$

(41)

Based on Theorem 8, $W$ and $G$ are updated as follows:

$$
W_{j, i} = \max(\frac{\gamma \bar{\Lambda}_{j, i}^* + \rho W_{j, i}}{\gamma + \rho}), 
$$

(42)

$$
G_{i, j} = U * S_1 * \mathcal{V}^H, 
$$

(43)

where $U$ and $\mathcal{V}$ are derived from the t-SVD of $G^* + \mu L^* + R^* - \mu G^* = U * S_2 * \mathcal{V}^H$. The relationship between $S_1$ and $S_2$ is given by Theorem 8.

The update for $\bar{\Lambda}$ turns out to be straightforward:

$$
\bar{\Lambda}^* = \frac{\gamma W^* + \rho \bar{\Lambda}^*}{\gamma + \rho}. 
$$

(44)

Fixed $G_{k_1 k_2}, E, N, R_{k_1 k_2}$ and $T$, the minimization problem $L$ is converted into the following form:

$$
\min_L \sum_{1 \leq k_1 < k_2 \leq N} \beta_{k_1 k_2} \frac{\mu_{k_1 k_2}}{2} ||L - G_{k_1 k_2} + R_{k_1 k_2}||_F^2 
$$

$$
+ \frac{\tau}{2} ||T - L - E - N + \frac{\rho}{\tau}||_F^2 = H_1(G, W) + H_2(W, \bar{\Lambda}) + H_3(E) 
$$

$$
+ H_4(N) + H_5(L, G) + H_6(L, E, N) 
$$

(45)

The closed form of $L$ can be derived by setting the derivative of (45) to zero. We can now update $L$ by the following equation:

$$
L^* = \frac{S}{\sum \mu_{k_1 k_2} + \tau + \rho^*}. 
$$

(46)
where \(S = \sum \mu_{k_1k_2}G^+_{k_1k_2} - R^+_{k_1k_2} + \tau(T - E^* - N^*) + F^* + \rho^*L^*\).

Now, let’s solve \(E\). The minimization problem of \(E\) is as follows:

\[
\min_{\tau_1} \tau_1 \|E\|_1 + \frac{\rho^*}{2} \|E - E^*\|_F^2 + \frac{\tau}{2} \|T - L^* - E - N^* + \frac{F^*}{\tau}\|_F^2.
\]

(47)

Problem (47) has the following closed-form solution:

\[
E^* = S_{\frac{c}{\tau_1}}\left(\frac{\tau(T - L^* + \frac{F^*}{\tau}) + \rho^*E^*}{\tau + \rho^*}\right),
\]

where \(S_{\lambda}(\cdot)\) is the soft thresholding operator (44):

\[
S_{\lambda}(x) = \begin{cases} 
0, & \text{if } |x| \leq \lambda, \\
\text{sign}(x) |x| - \lambda, & \text{if } |x| > \lambda 
\end{cases}
\]

(49)

The minimization problem of \(N\) is as follows:

\[
\min_{\tau_2} \|N\|_F^2 + \frac{\tau}{2} \|T - L^* - E^* - N + \frac{F^*}{\tau}\|_F^2 + \frac{\rho^*}{2} \|N - N^*\|_F^2.
\]

We update \(N\) by the following equation:

\[
N^* = \frac{\tau(T - L^* - E^*) + F^* + \rho^*N^*}{2\tau_2 + \tau + \rho^*}.
\]

(50)

Finally, multipliers \(R_{k_1k_2}\) and \(F\) are updated according to the following formula:

\[
\begin{align*}
R^+_{k_1k_2} &= R^+_{k_1k_2} + \mu_{k_1k_2}(L^* - G^+_{k_1k_2}); \\
F^* &= F^* + \tau(T - L^* - E^* - N^*).
\end{align*}
\]

(51)

EMLCP-based TPRC model computation is given in Algorithm 2. The main per-iteration cost lies in the update of \(G^+_{k_1k_2}\), which requires computing SVD and t-SVD. The per-iteration complexity is \(O(LE(\sum_{k_1k_2} \log(\log(k_{k_1k_2}) + \min(I_{k_1}, I_{k_2})))\)) where \(LE = \prod_{i=1}^N I_i\) and \(\text{le}(k_{k_1k_2}) = \text{le}(I_{k_1}, I_{k_2})\).

V. CONVERGENCE ANALYSIS

In this section, the convergence of PALM is established under some mild conditions, which is mainly based on the framework in [40].

Theorem 9: Suppose that \(\rho_1 = \gamma_1\mu\) with \(\gamma_1 > 1\). Let the sequence \{(\(Z, M, W, \Lambda\))\} be generated by Algorithm 1. Then,

(a) any accumulation point of the sequence \{(\(Z^k, M^k, W^k, \Lambda^k\))\} is a critical point of (27).

(b) if \(H_1\) is KL functions, the sequence \{(\(Z^k, M^k, W^k, \Lambda^k\))\} converges to a critical point of (27).

Proof: First, in the solution process, only the non-convex function \(H_1(M)\) is included in the iteration of \(M\), and the updates of other elements are solved by convex functions. The variables solved by the convex function are strictly descending, hence we get the following inequality:

\[
H_1(W^{k+1}) + H_2(W^{k+1}) + \frac{\rho}{2} \|W^{k+1} - W^k\|_F^2 \\
\leq H_1(W^k) + H_2(W^k) + \frac{\rho}{2} \|W^k - W^k\|_F^2,
\]

(52)

\[
H_2(\Lambda^{k+1}) + \frac{\rho}{2} \|\Lambda^{k+1} - \Lambda^k\|_F^2 \leq H_2(\Lambda^k),
\]

(53)

\[
H_3(\Xi^{k+1}) + \frac{\rho}{2} \|\Xi^{k+1} - \Xi^k\|_F^2 \leq H_3(\Xi^k),
\]

(54)

By the definition of \(H_3(M, \Xi)\) in (30), the gradients of \(H_3\) with respect to \(M\) and \(\Xi\), respectively, are

\[
\nabla M H_3(M, \Xi) = \mu(M - Z - \frac{Q}{\mu}),
\]

\[
\nabla \Xi H_3(M, \Xi) = \mu(\Xi - M + \frac{Q}{\mu}).
\]

(55)

(56)

For any fixed \(\Xi\), we obtain that

\[
\|\nabla M H_3(M_1, \Xi) - \nabla M H_3(M_2, \Xi)\|_F = \|\mu(M_1 - Z - \frac{Q}{\mu}) - \mu(M_2 - Z - \frac{Q}{\mu})\|_F
\]

\[
= \mu\|M_1 - M_2\|_F,
\]

(57)

and for any fixed \(M\), we get that

\[
\|\nabla \Xi H_3(M, \Xi_1) - \nabla \Xi H_3(M, \Xi_2)\|_F = \|\mu(\Xi_1 - M + \frac{Q}{\mu}) - \mu(\Xi_2 - M + \frac{Q}{\mu})\|_F
\]

\[
= \mu\|\Xi_1 - \Xi_2\|_F.
\]

(58)

[57] and [58] imply that the gradient of \(H_3(M, \Xi)\) is Lipschitz continuous block-wise. Note that \(H_3(M, \Xi)\) is twice continuously differentiable, which brings that \(\nabla^2 H_3(M, \Xi)\) is Lipschitz continuous on bounded subsets of \(\mathbb{R}^1 \times \mathbb{R}^1 \times \cdots \times \mathbb{R}^N \times \mathbb{R}^1 \times \mathbb{R}^1 \times \cdots \times \mathbb{R}^N\). So

\[
H_3(M^{k+1}) + H_1(M^{k+1}) + \frac{(\gamma_1 - 1)\mu}{2} \|M^{k+1} - M^k\|_F^2
\]

\[
\leq H_3(M^k) + H_1(M^k).
\]

(59)

From (59), we get that

\[
\begin{align*}
LAG(\Xi^{k+1}, M^{k+1}, W^{k+1}, \Lambda^{k+1}) &+ \frac{(\gamma_1 - 1)\mu}{2} \|M^{k+1} - M^k\|_F^2 + \frac{\rho}{2} \|W^{k+1} - W^k\|_F^2 \\
&= \|\Xi^{k+1} - \Xi^k\|_F^2 + \|Z^{k+1} - Z^k\|_F^2
\end{align*}
\]

\[
\leq LAG(\Xi^k, M^k, W^k, \Lambda^k),
\]

(60)

\[
\lim_{k \to +\infty} \|M^{k+1} - M^k\|_F = \lim_{k \to +\infty} \|W^{k+1} - W^k\|_F = \lim_{k \to +\infty} \|\Lambda^{k+1} - \Lambda^k\|_F = \lim_{k \to +\infty} \|Z^{k+1} - Z^k\|_F = 0.
\]

(61)

(a) Assume that there exists a subsequence \{(\(Z^{j_k}, M^{j_k}, W^{j_k}, \Lambda^{j_k}\))\} such that \{(\(Z^{j_k}, M^{j_k}, W^{j_k}, \Lambda^{j_k}\))\} converges to \((Z^*, M^*, W^*, \Lambda^*)\) as \(j \to +\infty\). By (61), we have that \{(\(Z^{j_k}, M^{j_k}, W^{j_k}, \Lambda^{j_k}\))\} also converges to \((Z^*, M^*, W^*, \Lambda^*)\) as \(j \to +\infty\). Moreover, the optimality conditions of (31) gives that

\[
0 = \nabla H_1(W^{j_k+1}) + \nabla H_2(W^{j_k+1}) + \rho k(W^{j_k+1} - W^{j_k}),
\]

\[
0 \in \partial H_1(M^{j_k+1}) - \mu(Z - M^{j_k} + \frac{Q}{\mu}) + \rho k(M^{j_k+1} - M^{j_k}),
\]
0 = \nabla H_2(\tilde{\lambda}^{k+1}_j) + \rho^k(\tilde{\lambda}^{k+1}_j - \tilde{\lambda}_j),

0 = \nabla H_3(\tilde{Z}^{k+1}_j) + \rho^k(\tilde{Z}^{k+1}_j - Z^{k}_j),

When j \to +\infty, by (45), we get that

0 = \nabla H_1(W^*) + \nabla H_2(W^*),

0 = \partial H_1(\mathcal{M}^*) - \mu(\mathcal{Z} - \mathcal{M}^* + \frac{Q}{\mu}) = \partial H_1(\mathcal{M}^*) + \nabla H_3(\mathcal{M}^*),

0 = \nabla H_2(\tilde{\lambda}^*),

0 = \nabla H_3(\tilde{Z}^*),

Therefore, we obtain that

(0, 0, 0, 0) \in \partial \text{LAG}(\tilde{Z}^*, \mathcal{M}^*, \mathcal{W}^*, \tilde{\lambda}^*)

which implies that (\tilde{Z}^*, \mathcal{M}^*, \mathcal{W}^*, \tilde{\lambda}^*) is a critical point of \text{LAG}(\mathcal{Z}, \mathcal{M}, \mathcal{W}, \tilde{\lambda}).

(b) By the definition of \text{LAG}(\mathcal{Z}, \mathcal{M}, \mathcal{W}, \tilde{\lambda}), we have that

\text{LAG}(\mathcal{Z}, \mathcal{M}, \mathcal{W}, \tilde{\lambda}) \rightarrow +\infty \text{ as } \|\|\mathcal{Z}, \mathcal{M}, \mathcal{W}, \tilde{\lambda}\|\|_F \rightarrow +\infty.

Suppose that (\tilde{Z}^k, \tilde{M}^k, \tilde{W}^k, \tilde{\lambda}^k) is unbounded, i.e.,

\|\|\tilde{Z}^k, \tilde{M}^k, \tilde{W}^k, \tilde{\lambda}^k\|\|_F \rightarrow +\infty,

derive that \text{LAG}(\tilde{Z}^k, \tilde{M}^k, \tilde{W}^k, \tilde{\lambda}^k) \rightarrow +\infty. However, it follows from (60) that \text{LAG}(\tilde{Z}^k, \tilde{M}^k, \tilde{W}^k, \tilde{\lambda}^k) is upper bounded. Therefore, the sequence \{(\tilde{Z}^k, \tilde{M}^k, \tilde{W}^k, \tilde{\lambda}^k)\} is bounded.

From (39), Logarithmic function is KL function. Thus, \text{H}_1 also KL function. Notice that \text{H}_2 and \text{H}_3 are KL functions, we have that \text{LAG}(\tilde{Z}^k, \tilde{M}^k, \tilde{W}^k, \tilde{\lambda}^k) is also a KL function (40). Then by (40), we obtain that the sequence \{(\tilde{Z}^k, \tilde{M}^k, \tilde{W}^k, \tilde{\lambda}^k)\} converges to a critical point of (27).

Theorem 10: Suppose that \rho_1 = \gamma_1 \mu with \gamma_1 > 1. Let the sequence \{(L, G, W, \tilde{\lambda}, \mathcal{E}, N)\} be generated by Algorithm 2.

Then, (a) any accumulation point of the sequence \{(L, G, W, \tilde{\lambda}, \mathcal{E}, N)\} is a critical point of (38).

(b) if \text{H}_1 is KL functions and coercive, the sequence \{(L, G, W, \tilde{\lambda}, \mathcal{E}, N)\} converges to a critical point of (38).

Proof: Compared with LRTC, the TRPCA algorithm has two more variables, \mathcal{E} and \mathcal{N}, but its solutions are all convex functions. Therefore, the convergence proof of the TRPCA algorithm is similar to that of the LRTC algorithm, and will not be repeated here.

VI. EXPERIMENTS

We evaluate the performance of the proposed EMLCP-based LRTC and TRPCA methods. All methods are tested on real-world data. We employ the peak signal-to-noise ratio (PSNR) value, the structural similarity (SSIM) value (46), the feature similarity (FSIM) value (47), and error relative globale adimensionnelle de synthèse (ERGAS) value (48) to measure the quality of the recovered results. The PSNR, SSIM and FSIM value are the bigger the better, and the ERGAS value is the smaller the better. For simplicity, EMLCP-based LRTC and EMLCP-based TRPCA are denoted as EMLCP. All tests are implemented on the Windows 10 platform and MATLAB (R2019a) with an Intel Core i7-10875H 2.30 GHz and 32 GB of RAM.

A. Low-rank tensor completion

In this section, we test three kinds of real-world data: MSI, MRI and CV. The method for sampling the data is purely random sampling. The comparative LRTC methods are as follows: HalRRTC (49), and LRTCTV-I (50) represent state-of-the-art for the Tucker-decomposition-based methods; TNN (27), PSTNN (51), FTNN (52), WSTNN (28), and nonconvex WSTNN (53) represent state-of-the-art for the t-SVD-based methods; and minmax concave plus penalty-based TC method (McpTC) (54). Since the TNN, PSTNN, and FTNN methods are only applicable to three-order tensors, in all four-order tensor tests, we first reshape the four-order tensor into three-order tensors and then test the performances of these methods. It is not difficult to find that the NWSTNN method in the comparison method adopts the non-convex relaxation of the Logarithmic function, and the results obtained by comparing with such method are consistent with our theory property.

1) MSI completion: We test 32 MSIs in the dataset CAVE\footnote{http://www.cs.columbia.edu/CAVE/databases/multispectral/} All testing data are of size 256 × 256 × 31. In Fig4 we randomly select three from 32 MSIs, and brings the different sampling rate and different band visual results. The individual MSI names and their corresponding bands are written in the caption of Fig4. As can be seen from Fig4, the visual effect of the EMLCP method is superior to the NWSTNN method under all sample rate, which is consistent with our theory. To further highlight the superiority of our method, the average quantitative results of 32 MSIs are listed in Table I. The results show that the PSNR value of our algorithms is 0.4dB higher than that of the suboptimal method when the sampling rate is 20%, and even reaches 0.8dB when the sampling rate is 5%. More experimental results are available in the Appendix K.

2) MRI completion: We test the performance of the proposed method and the comparative method on MRI\footnote{http://brainweb.bic.mni.mcgill.ca/brainweb/selection_normal.html} data with the size of 181 × 217 × 181. First, we demonstrate the visual effect recovered by MRI data at sampling rates of 5%, 10% and 20% in Fig5. Our method is clearly superior to the comparative methods. Then, we list the average quantitative results of frontal slices of MRI restored by all methods at different sampling rates in Table II. Obviously, the PSNR value of our method is at average 0.3dB higher than that of the suboptimal method, and the values of SSIM, FSIM and ERGAS are significantly better than that of the suboptimal method.

3) CV completion: We test nine CV\footnote{http://trace.eas.asu.edu/yuv/} respectively named news, akiyo, hall, highway, foreman, container, coastguard, susie, carphone) of size 144 × 176 × 3 × 50. Firstly, we list the average quantitative results of 9 CVs in Table III. At this time, the suboptimal method is the NWSTNN method. The PSNR value of our method is average 0.4dB higher than it at three sampling rates. Furthermore, we demonstrate the visual results of 9 CVs in our experiment in Fig6 in which the number of frames and sampling rate corresponding to each CV are described in the caption of Fig6. It is not hard to see from the picture that the recovery of our method on the vision
TABLE I
THE AVERAGE PSNR, SSIM, FSIM AND ERGAS VALUES FOR 32 MSIs TESTED BY OBSERVED AND THE NINE UTILIZED LRTC METHODS.

| Method  | PSNR  | SSIM  | FSIM  | ERGAS  | PSNR  | SSIM  | FSIM  | ERGAS  | PSNR  | SSIM  | FSIM  | ERGAS  |
|---------|-------|-------|-------|--------|-------|-------|-------|--------|-------|-------|-------|--------|
| Observed| 15.438| 0.153 | 0.644 | 845.388| 15.673| 0.194 | 0.646 | 822.788| 16.184| 0.269 | 0.650 | 775.866|
| HaLRTC | 18.112| 0.285 | 0.697 | 689.482| 22.694| 0.527 | 0.786 | 478.325| 32.175| 0.835 | 0.910 | 190.848|
| TNN    | 17.986| 0.247 | 0.685 | 726.893| 28.627| 0.678 | 0.861 | 314.352| 40.170| 0.964 | 0.972 | 59.018 |
| LRTCTV-I| 25.894| 0.800 | 0.835 | 276.620| 30.709| 0.890 | 0.906 | 162.567| 35.486| 0.949 | 0.957 | 94.646 |
| McpTC  | 32.459| 0.875 | 0.909 | 133.472| 35.959| 0.925 | 0.943 | 91.788 | 40.518| 0.964 | 0.972 | 56.083 |
| PSTNN  | 18.713| 0.474 | 0.650 | 574.637| 23.239| 0.683 | 0.783 | 352.012| 34.206| 0.924 | 0.942 | 117.472|
| FTNN   | 32.620| 0.899 | 0.924 | 131.871| 37.182| 0.954 | 0.963 | 78.694  | 43.002| 0.984 | 0.987 | 41.625 |
| WSTNN  | 31.439| 0.806 | 0.911 | 208.988| 40.170| 0.981 | 0.981 | 52.895  | 47.059| 0.995 | 0.995 | 24.914 |
| NWSTNN | 37.417| 0.945 | 0.950 | 71.261 | 43.704| 0.985 | 0.985 | 35.779  | 51.362| 0.997 | 0.997 | 15.572 |
| EMLCP  | 38.298| 0.962 | 0.964 | 64.689 | 44.340| 0.988 | 0.988 | 33.329  | 51.742| 0.997 | 0.997 | 14.779 |

Fig. 1. (a) Original image. (b) Observed image. (c) HaLRTC. (d) TNN. (e) LRTCTV-I. (f) McpTC. (g) PSTNN. (h) FTNN. (i) WSTNN. (j) NWSTNN. (k) EMLCP. SR: top row is 5%, middle row is 10% and last row is 20%. The rows of MSIs are in order: balloons, beads, watercolors. The corresponding bands in each row are: 31, 20, 10.

TABLE II
THE PSNR, SSIM, FSIM AND ERGAS VALUES OUTPUT BY BY OBSERVED AND THE NINE UTILIZED LRTC METHODS FOR MRI.

| Method  | PSNR  | SSIM  | FSIM  | ERGAS  | PSNR  | SSIM  | FSIM  | ERGAS  | PSNR  | SSIM  | FSIM  | ERGAS  |
|---------|-------|-------|-------|--------|-------|-------|-------|--------|-------|-------|-------|--------|
| Observed| 11.399| 0.310 | 0.530 | 1021.071| 11.633| 0.323 | 0.565 | 994.049| 12.149| 0.350 | 0.613 | 936.747|
| HaLRTC | 17.372| 0.301 | 0.638 | 532.927| 20.105| 0.439 | 0.726 | 391.945| 24.451| 0.659 | 0.829 | 235.019|
| TNN    | 22.681| 0.470 | 0.742 | 303.284| 26.064| 0.643 | 0.812 | 205.410| 29.972| 0.798 | 0.882 | 130.791|
| LRTCTV-I| 19.400| 0.598 | 0.702 | 431.241| 22.864| 0.749 | 0.805 | 294.937| 28.236| 0.891 | 0.908 | 155.272|
| McpTC  | 27.931| 0.800 | 0.843 | 154.029| 31.439| 0.844 | 0.888 | 102.744| 35.570| 0.937 | 0.941 | 63.906 |
| PSTNN  | 17.064| 0.243 | 0.639 | 542.819| 22.870| 0.487 | 0.757 | 297.337| 29.083| 0.772 | 0.870 | 145.165|
| FTNN   | 24.673| 0.687 | 0.836 | 234.329| 28.297| 0.820 | 0.896 | 152.733| 32.767| 0.919 | 0.947 | 89.543 |
| WSTNN  | 25.524| 0.708 | 0.825 | 211.315| 29.059| 0.837 | 0.888 | 139.177| 33.497| 0.928 | 0.940 | 82.851 |
| NWSTNN | 30.222| 0.826 | 0.884 | 119.820| 33.293| 0.902 | 0.924 | 83.608  | 36.860| 0.950 | 0.956 | 54.962 |
| EMLCP  | 30.563| 0.850 | 0.893 | 115.395| 33.643| 0.918 | 0.932 | 80.590  | 37.180| 0.959 | 0.962 | 53.344 |

Effect is better. More experimental results are available in the Appendix L.

B. Tensor robust principal component analysis

In this section, we evaluate the performance of the proposed TRPCA method through HSI mixed noise denoising. The comparative TRPCA methods include the SNN [55], TNN [41], 3DTNN and 3DLogTNN [53] methods.

1) HSI denoising: We test the Pavia University data sets and Washington DC Mall data sets, where Pavia University data size is 200 × 200 × 80 and Washington DC Mall data size is 256 × 256 × 150. We divide the mixed noise into two kinds, one is independent identically distributed Gaussian noise plus independent identically distributed pepper and salt noise, and the other is non i.i.d. Gaussian noise plus i.i.d pepper and salt noise, where σ is pepper and salt noise and ν is Gaussian noise. In Table IV we list the quantitative numerical results of Pavia University and Washington DC Mall Data under 3 combinations of these two kinds of noise respectively. It can be seen that under the influence of the weakest noise, the PSNR value of the obtained results is 0.6 dB higher than that of the suboptimal method 3DLogTNN. Even under the influence
of the most severe noise, the PSNR value of the obtained results is still better than the suboptimal method 3DLogTNN. In Fig.4, we show the visual results of the two kinds of data of the most severe noise, the PSNR value of the obtained results is still better than the suboptimal method 3DLogTNN. In Fig.4, we show the visual results of the two kinds of data of the most severe noise, the PSNR value of the obtained results is still better than the suboptimal method 3DLogTNN. In Fig.4, we show the visual results of the two kinds of data of the most severe noise, the PSNR value of the obtained results is still better than the suboptimal method 3DLogTNN. In Fig.4, we show the visual results of the two kinds of data of the most severe noise, the PSNR value of the obtained results is still better than the suboptimal method 3DLogTNN. In Fig.4, we show the visual results of the two kinds of data of the most severe noise, the PSNR value of the obtained results is still better than the suboptimal method 3DLogTNN. In Fig.4, we show the visual results of the two kinds of data of the most severe noise, the PSNR value of the obtained results is still better than the suboptimal method 3DLogTNN.
Extensive experiments show that our method can achieve good visual and numerical quantitative results. The obtained numerical quantitative results outperform the NWSTNN method using Logarithmic function, which is consistent with our theoretical analysis. In addition, it is worth studying whether the MLCP function can be extended in more applications.

REFERENCES

[1] Y.-M. Huang, H.-Y. Yan, Y.-W. Wen, and X. Yang, “Rank minimization with applications to image noise removal,” Information Sciences, vol. 429, pp. 147–163, 2018.

[2] B. Madathil and S. N. George, “Twist tensor total variation regularized-reweighted nuclear norm based tensor completion for video missing area recovery,” Information Sciences, vol. 423, pp. 376–397, 2018.

[3] Y. Wang, D. Meng, and M. Yuan, “Sparse recovery: from vectors to tensors,” National Science Review, vol. 5, no. 5, pp. 756–767, 2018.

[4] 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, 2020.

[5] S. Li, R. Dian, L. Fang, and J. M. Bioucas-Dias, “Fusing hyperspectral and multispectral images via coupled sparse tensor factorization,” IEEE Transactions on Image Processing, vol. 27, no. 8, pp. 4118–4130, 2018.

[6] X. Fu, W.-K. Ma, J. M. Bioucas-Dias, and T.-H. Chan, “Semiblind hyperspectral unmixing in the presence of spectral library mismatches,” IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 9, pp. 5171–5184, 2016.

[7] J. Xue, Y. Zhao, W. Liao, and J. C.-W. Chan, “Nonlocal low-rank regularized tensor decomposition for hyperspectral image denoising,” IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 7, pp. 5171–5184, 2019.

[8] J. Xue, Y. Zhao, W. Liao, J. C.-W. Chan, and S. G. Kong, “Enhanced sparsity prior model for low-rank tensor completion,” IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 11, pp. 4567–4581, 2020.
