Journal of Machine Learning Research 23 (2022) 1-60 Submitted 12/20; Revised 2/22; Published 4/22

Low-rank Tensor Learning with Nonconvex Overlapped Nuclear Norm Regularization

Quanming Yao QYAOAA@TSINGHUA.EDU.CN
Department of Electronic Engineering, Tsinghua University

Yaqing Wang[*] WANGYAQING01@BAIDU.COM
Baidu Research, Baidu Inc.

Bo Han BHANML@COMP.HKBU.EDU.HK
Department of Computer Science, Hong Kong Baptist University

James T. Kwok JAMESK@CSE.UST.HK
Department of Computer Science and Engineering, Hong Kong University of Science and Technology

Editor: Prateek Jain

Abstract

Nonconvex regularization has been popularly used in low-rank matrix learning. However, extending it for low-rank tensor learning is still computationally expensive. To address this problem, we develop an efficient solver for use with a nonconvex extension of the overlapped nuclear norm regularizer. Based on the proximal average algorithm, the proposed algorithm can avoid expensive tensor folding/unfolding operations. A special "sparse plus low-rank" structure is maintained throughout the iterations, and allows fast computation of the individual proximal steps. Empirical convergence is further improved with the use of adaptive momentum. We provide convergence guarantees to critical points on smooth losses and also on objectives satisfying the Kurdyka-Łojasiewicz condition. While the optimization problem is nonconvex and nonsmooth, we show that its critical points still have good statistical performance on the tensor completion problem. Experiments on various synthetic and real-world data sets show that the proposed algorithm is efficient in both time and space and more accurate than the existing state-of-the-art.

Keywords: Low-rank tensor, Proximal algorithm, Proximal average algorithm, Nonconvex regularization, Overlapped nuclear norm.

1. Introduction

Tensors can be seen as high-order matrices and are widely used for describing multilinear relationships in the data. They have been popularly applied in areas such as computer vision, data mining and machine learning (Kolda and Bader, 2009; Zhao et al., 2016; Song et al., 2017; Papalexakis et al., 2017; Hong et al., 2020; Janzamin et al., 2020). For example, color images (Liu et al., 2013), hyper-spectral images (Signoretto et al., 2011; He et al., 2019), and knowledge graphs (Nickel et al., 2015; Lacroix et al., 2018) can be naturally represented as third-order tensors, while color videos can be seen as 4-order tensors (Candès et al., 2011; Bengua et al., 2017). In YouTube, users can follow each other and belong to the same subscribed channels. By treating channel as the third dimension, the users' co-subscription network can also be represented as a third-order tensor (Lei et al., 2009).

[*] Corresponding Author.

In many applications, only a few entries in the tensor are observed. For example, each YouTube user usually only interacts with a few other users (Lei et al., 2009; Davis et al., 2011), and in knowledge graphs, we can only have a few labeled edges describing the relations between entities (Nickel et al., 2015; Lacroix et al., 2018). Tensor completion, which aims at filling in this partially observed tensor, has attracted a lot of recent interest (Rendle and Schmidt-Thieme, 2010; Signoretto et al., 2011; Bahadori et al., 2014; Cichocki et al., 2015).

In the related task of matrix completion, the underlying matrix is often assumed to be low-rank (Candès and Recht, 2009), as its rows/columns share similar characteristics. The nuclear norm, which is the tightest convex envelope of the matrix rank (Boyd and Vandenberghe, 2009), is popularly used as its surrogate in low-rank matrix completion (Cai et al., 2010; Mazumder et al., 2010). In tensor completion, the low-rank assumption also captures relatedness in the different tensor dimensions (Tomioka et al., 2010; Acar et al., 2011; Song et al., 2017; Hong et al., 2020). However, tensors are more complicated than matrices. Indeed, even the computation of tensor rank is NP-hard (Hillar and Lim, 2013). In recent years, many convex relaxations based on the matrix nuclear norm have been proposed for tensors. Examples include the tensor trace norm (Cheng et al., 2016), overlapped nuclear norm (Tomioka et al., 2010; Gandy et al., 2011), and latent nuclear norm (Tomioka et al., 2010). Among these convex relaxations, the overlapped nuclear norm is the most popular as it (i) can be evaluated exactly by performing SVD on the unfolded matrices (Cheng et al., 2016), (ii) has better low-rank approximation (Tomioka et al., 2010), and (iii) can lead to exact recovery (Tomioka et al., 2011; Tomioka and Suzuki, 2013; Mu et al., 2014).

The (overlapped) nuclear norm equally penalizes all singular values. Intuitively, larger singular values are more informative and should be less penalized (Mazumder et al., 2010; Lu et al., 2016b; Yao et al., 2019b). To alleviate this problem in low-rank matrix learning, various adaptive nonconvex regularizers have been recently introduced. Examples include the capped-ℓ_1 norm (Zhang, 2010b), log-sum-penalty (LSP) (Candès et al., 2008), truncated nuclear norm (TNN) (Hu et al., 2013), smoothed-capped-absolute-deviation (SCAD) (Fan and Li, 2001) and minimax concave penalty (MCP) (Zhang, 2010a). All these assign smaller penalties to the larger singular values. This leads to better recovery performance in many applications such as image recovery (Lu et al., 2016b; Gu et al., 2017) and collaborative filtering (Yao et al., 2019b), and lower statistical errors of various matrix completion and regression problems (Gui et al., 2016; Mazumder et al., 2020).

Motivated by the success of adaptive nonconvex regularizers in low-rank matrix learning, there are recent works that apply nonconvex regularization in learning low-rank tensors. For example, the TNN regularizer is used with the overlapped nuclear norm regularizer on video processing (Xue et al., 2018) and traffic data processing (Chen et al., 2020). In this paper, we propose a general nonconvex variant of the overlapped nuclear norm regularizer for low-rank tensor completion. Unlike the standard convex tensor completion problem, the resulting optimization problem is nonconvex and more difficult to solve. Previous algorithms in (Xue et al., 2018; Chen et al., 2020) are computationally expensive, and have neither convergence results nor statistical guarantees.

To solve this issue, based on the proximal average algorithm (Bauschke et al., 2008), we develop an efficient solver with much smaller time and space complexities. The keys to its success are on (i) avoiding expensive tensor folding/unfolding, (ii) maintaining a "sparse plus low-rank" structure on the iterates, and (iii) incorporating the adaptive momentum (Li and Lin, 2015; Li et al., 2017; Yao et al., 2017). Convergence guarantees to critical points are provided under the usual smoothness assumption for the loss and further Kurdyka-Łojasiewicz (Attouch et al., 2013) condition on the whole learning objective.

Besides, we study its statistical guarantees, and show that critical points of the proposed objective can have small statistical errors under the restricted strong convexity condition (Agarwal et al., 2010). Informally, for tensor completion with unknown noise, we show that the recovery error can be bounded as $\|\mathcal{X}^* - \tilde{\mathcal{X}}\|_F \le \mathcal{O}(\lambda\kappa_0 \sum_{i=1}^{M} \sqrt{k_i})$ (see Theorem 16), where $\mathcal{O}$ omits constant terms, $\mathcal{X}^*$ (resp. $\tilde{\mathcal{X}}$) is the ground-truth (resp. recovered) tensor, M is the tensor order and k_i is the rank of unfolding matrix on the ith mode. When Gaussian additive noise is assumed, we show that the recovery error also depends linearly with the noise level σ (see Corollary 17) and $\sqrt{\log I^{\pi} / \|\boldsymbol{\Omega}\|_1}$ where I^{π} is the tensor size and $\|\boldsymbol{\Omega}\|_1$ is the number of observed elements (see Corollary 18).

We further extend it for use with Laplacian regularizer as in spatial-temporal analysis and nonsmooth losses as in robust tensor completion. Experiments on a variety of synthetic and real-world data sets (including images, videos, hyper-spectral images, social networks, knowledge graphs and spatial-temporal climate observation records) show that the proposed algorithm is more efficient and has much better empirical performance than other low-rank tensor regularization and decomposition methods.

Difference with the Conference Version

A preliminary conference version of this work (Yao et al., 2019a) was published in ICML-2019. The main differences with this conference version are as follows.

1). Only third-order tensor and square loss are considered in (Yao et al., 2019a), while the proposed algorithm here, which is enabled by Proposition 4, can work on tensors with arbitrary orders. The difficulties of extending to higher order tensors are also discussed after Proposition 4.

2). Statistical guarantee of the proposed model for the tensor completion problem is added in Section 3.5, which shows that tensors that are not too spiky can be recovered. We also show how the recovery performance can depend on noise level, tensor ranks, and the number of observations.

3). In Section 4, we enable the proposed method work with robust loss function (which is nonconvex and nonsmooth) and Laplacian regularizer. These enable the proposed algorithm to be applied to a wider range of application scenarios such as knowledge graph completion, spatial-temporal analysis and robust video recovery.

4). Extensive experiments are added. Specifically, quality of the obtained critical points is studied in Section 5.1.3; application to knowledge graphs in Section 5.3, application to robust video completion in Section 5.4, and application to spatial-temporal data in Section 5.5.

Notation

Vectors are denoted by lowercase boldface, matrices by uppercase boldface, and tensors by Euler.

For a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ (without loss of generality, we assume that $m \ge n$), $\sigma_i(\boldsymbol{A})$ denotes its ith singular, its nuclear norm is $\|\boldsymbol{A}\|_* = \sum_i \sigma_i$; $\|\boldsymbol{A}\|_\infty$ returns its maximum singular.

For tensors, we follow the notation in (Kolda and Bader, 2009). For a M-order tensor $\mathcal{X} \in \mathbb{R}^{I^1 \times \cdots \times I^M}$ (without loss of generality, we assume $I^1 \ge \cdots \ge I^M$), its $(i_1, \dots, i_M)$th entry is $\mathcal{X}_{i_1 \dots i_M}$. One can *unfold* $\mathcal{X}$ along its dth mode to obtain the matrix $\boldsymbol{X}_{\langle d \rangle} \in \mathbb{R}^{I^d \times (\frac{I^\pi}{I^d})}$ with $I^\pi = \prod_{i=1}^{M} I^i$, whose (i_d, j) entry is $\mathcal{X}_{i_1 \dots i_M}$ with $j = 1 + \sum_{l=1, l \neq d}^{M} (i_l - 1) \prod_{m=1, m \neq d}^{l-1} I^m$. One can also *fold* a matrix $\boldsymbol{X}$ back to a tensor $\mathcal{X} = \boldsymbol{X}^{\langle d \rangle}$, with $\mathcal{X}_{i_1 \dots i_M} = \boldsymbol{X}_{i_d j}$, and j as defined above. A slice in a tensor $\mathcal{X}$ is a matrix $\boldsymbol{x}$ obtained by fixing all but two $\mathcal{X}$'s indices. The inner product of two M-order tensors $\mathcal{X}$ and $\mathcal{Y}$ is $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I^1} \dots \sum_{i_M=1}^{I^M} \mathcal{X}_{i_1 \dots i_M} \mathcal{Y}_{i_1 \dots i_M}$, the Frobenius norm is $\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$, $\|\mathcal{X}\|_{\max}$ returns the value of the element in $\mathcal{X}$ with the maximum absolute value.

For a proper and lower-semi-continuous function f, ∂f denotes its Frechet subdifferential (Attouch et al., 2013).

Finally, $P_{\mathbf{\Omega}}(\cdot)$ is the observer operator, i.e., given a binary tensor $\mathbf{\Omega} \in \{0,1\}^{I_1 \times \dots \times I_M}$ and an arbitrary tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_M}$, $[P_{\mathbf{\Omega}}(\mathcal{X})]_{i_1 \dots i_M} = \mathcal{X}_{i_1 \dots i_M}$ if $\mathbf{\Omega}_{i_1 \dots i_M} = 1$ and $[P_{\mathbf{\Omega}}(\mathcal{X})]_{i_1 \dots i_M} = 0$ otherwise.

2. Related Works

2.1 Low-Rank Matrix Learning

Learning of a low-rank matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ can be formulated as the following optimization problem:

$$\min_{\boldsymbol{X}} f(\boldsymbol{X}) + \lambda r(\boldsymbol{X}), \tag{1}$$

where r is a low-rank regularizer, $\lambda \geq 0$ is a hyperparameter, and f is a loss function that is ρ-Lipschitz smooth[1]. Existing methods for low-rank matrix learning generally fall into three types: (i) nuclear norm minimization; (ii) nonconvex regularization; and (iii) matrix factorization.

2.1.1 NUCLEAR NORM MINIMIZATION

A common choice for r is the nuclear norm regularizer. Using the proximal algorithm (Parikh and Boyd, 2013) on (1), the iterate at iteration t is given by $\boldsymbol{X}_{t+1} = \text{prox}_{\frac{\lambda}{\tau}\|\cdot\|_*}(\boldsymbol{Z}_t)$, where

$$\boldsymbol{Z}_t = \boldsymbol{X}_t - \frac{1}{\tau} \nabla f(\boldsymbol{X}_t). \tag{2}$$

Here, $\tau > \rho$ controls the stepsize ($1/\tau$), and

$$\text{prox}_{\frac{\lambda}{\tau}\|\cdot\|_*}(\boldsymbol{Z}) = \arg\min_{\boldsymbol{X}} \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{Z}\|_F^2 + \frac{\lambda}{\tau} \|\boldsymbol{X}\|_* \tag{3}$$

is the proximal step. The following Lemma shows that $\text{prox}_{\frac{\lambda}{\tau}\|\cdot\|_*}(\boldsymbol{Z})$ can be obtained by shrinking the singular values of $\boldsymbol{Z}$, which encourages $\boldsymbol{X}_t$ to be low-rank.

Lemma 1 *(Cai et al., 2010)* $\text{prox}_{\lambda\|\cdot\|_*}(\boldsymbol{Z}) = \boldsymbol{U}(\boldsymbol{\Sigma} - \lambda \boldsymbol{I})_+ \boldsymbol{V}^\top$*, where* $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ *is the SVD of* $\boldsymbol{Z}$*, and* $[(\boldsymbol{X})_+]_{ij} = \max(\boldsymbol{X}_{ij}, 0)$.

A special class of low-rank matrix learning problems is matrix completion, which attempts to find a low-rank matrix that agrees with the observations in data $\boldsymbol{O}$:

$$\min_{\boldsymbol{X}} \frac{1}{2} \|P_{\mathbf{\Omega}}(\boldsymbol{X} - \boldsymbol{O})\|_F^2 + \lambda \|\boldsymbol{X}\|_*. \tag{4}$$

Here, positions of the observed elements in $\boldsymbol{O}$ are indicated by 1's in the binary matrix $\mathbf{\Omega}$. Setting $f(\boldsymbol{X}) = \frac{1}{2}\|P_{\mathbf{\Omega}}(\boldsymbol{X} - \boldsymbol{O})\|_F^2$ in (1), $\boldsymbol{Z}_t$ in (2) becomes:

$$\boldsymbol{Z}_t = \boldsymbol{X}_t - \frac{1}{\tau} P_{\mathbf{\Omega}}(\boldsymbol{X}_t - \boldsymbol{O}). \tag{5}$$

[1] In other words, $\|\nabla f(\boldsymbol{X}) - \nabla f(\boldsymbol{Y})\|_F \leq \rho \|\boldsymbol{X} - \boldsymbol{Y}\|_F$ for any $\boldsymbol{X}, \boldsymbol{Y}$.

Note that $\boldsymbol{X}_t$ is low-rank and $\frac{1}{\tau} P_{\boldsymbol{\Omega}}\left(\boldsymbol{X}_t - \boldsymbol{O}\right)$ is sparse. $\boldsymbol{Z}_t$ thus has a "sparse plus low-rank" structure. This allows the SVD computation in Lemma 1 to be much more efficient (Mazumder et al., 2010). Specifically, on using the power method to compute $\boldsymbol{Z}_t$'s SVD, most effort is spent on multiplications of the forms $\boldsymbol{Z}_t \boldsymbol{b}$ and $\boldsymbol{a}^\top \boldsymbol{Z}_t$ (where $\boldsymbol{a} \in \mathbb{R}^n$ and $\boldsymbol{b} \in \mathbb{R}^m$). Let $\boldsymbol{X}_t$ in (5) be low-rank factorized as $\boldsymbol{U}_t \boldsymbol{V}_t^\top$, where $\boldsymbol{U}_t \in \mathbb{R}^{m \times k_t}$ and $\boldsymbol{V}_t \in \mathbb{R}^{n \times k_t}$ with rank k_t. Computing

$$\boldsymbol{Z}_t \boldsymbol{b} = \boldsymbol{U}_t \left(\boldsymbol{V}_t^\top \boldsymbol{b}\right) - \frac{1}{\tau} P_{\boldsymbol{\Omega}}\left(\boldsymbol{Y}_t - \boldsymbol{O}\right) \boldsymbol{b} \tag{6}$$

takes $O((m+n)k_t + \|\boldsymbol{\Omega}\|_1)$ time. Usually, $k_t \ll n$ and $\|\boldsymbol{\Omega}\|_1 \ll mn$. Thus, this is much faster than directly multiplying $\boldsymbol{Z}_t$ and $\boldsymbol{b}$, which takes $O(mn)$ time. The same holds for computing $\boldsymbol{a}^\top \boldsymbol{Z}_t$. The proximal step in (3) takes a total of $O((m+n)k_t k_{t+1} + \|\boldsymbol{\Omega}\|_1 k_{t+1})$ time, while a direct computation without utilizing the "sparse plus low-rank" structure takes $O(mnk_{t+1})$ time. Besides, as only $P_{\boldsymbol{\Omega}}\left(\boldsymbol{X}_t\right)$ and the factorized form of $\boldsymbol{X}_t$ need to be kept, the space complexity is reduced from $O(mn)$ to $O((m+n)k_t + \|\boldsymbol{\Omega}\|_1)$.

2.1.2 NONCONVEX LOW-RANK REGULARIZER

Instead of using a convex r in (1), the following nonconvex regularizer has been commonly used (Gui et al., 2016; Lu et al., 2016b; Gu et al., 2017; Yao et al., 2019b):

$$\phi(\boldsymbol{X}) = \sum\nolimits_{i=1}^{n} \kappa(\sigma_i(\boldsymbol{X})), \tag{7}$$

where κ is nonconvex and possibly nonsmooth. We assume the following on κ.

Assumption 1 *$\kappa(\alpha)$ is a concave and non-decreasing function on $\alpha \geq 0$, with $\kappa(0) = 0$ and $\lim_{\alpha \to 0^+} \kappa'(\alpha) = \kappa_0$ for a positive constant κ_0.*

Table 1 shows the κ's corresponding to the popular nonconvex regularizers of capped-ℓ_1 penalty (Zhang, 2010b), log-sum-penalty (LSP) (Candès et al., 2008), truncated nuclear norm (TNN) (Hu et al., 2013), smoothed-capped-absolute-deviation (SCAD) (Fan and Li, 2001), and minimax concave penalty (MCP) (Zhang, 2010a). These nonconvex regularizers have similar statistical guarantees (Gui et al., 2016), and perform empirically better than the convex nuclear norm regularizer (Lu et al., 2016b; Yao et al., 2019b). The proximal algorithm can also be used, and converges to a critical point (Attouch et al., 2013). Analogous to Lemma 1, the underlying proximal step

$$\text{prox}_{\frac{\lambda}{\tau}\phi}(\boldsymbol{Z}) = \arg\min\nolimits_{\boldsymbol{X}} \frac{1}{2} \left\|\boldsymbol{X} - \boldsymbol{Z}\right\|_F^2 + \frac{\lambda}{\tau}\phi(\boldsymbol{X}) \tag{8}$$

can be obtained as follows.

Lemma 2 (Lu et al., 2016b) *$\text{prox}_{\lambda\phi}(\boldsymbol{Z}) = \boldsymbol{U}\, \text{Diag}\left(y_1, \dots, y_n\right) \boldsymbol{V}^\top$, where $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is the SVD of $\boldsymbol{Z}$, and $y_i = \arg\min_{y \geq 0} \frac{1}{2}(y - \sigma_i(\boldsymbol{Z}))^2 + \lambda\kappa(y)$.*

2.1.3 MATRIX FACTORIZATION

Note that the aforementioned regularizers require access to individual singular values. As computing the singular values of a $m \times n$ matrix (with $m \geq n$) via SVD takes $O(mn^2)$ time, this can be costly for a large matrix. Even when rank-k truncated SVD is used, the computation cost is still

Table 1: Common examples of $\kappa(\sigma_i(\boldsymbol{X}))$. Here, θ is a constant. For the capped-ℓ_1, LSP and MCP, $\theta > 0$; for SCAD, $\theta > 2$; and for TNN, θ is a positive integer.

	$\kappa(\sigma_i(\boldsymbol{X}))$
nuclear norm (Candès and Recht, 2009)	$\sigma_i(\boldsymbol{X})$
capped-ℓ_1 (Zhang, 2010b)	$\min(\sigma_i(\boldsymbol{X}), \theta)$
LSP (Candès et al., 2008)	$\log(\frac{\sigma_i(\boldsymbol{X})}{\theta} + 1)$
TNN (Hu et al., 2013)	$\begin{cases} \sigma_i(\boldsymbol{X}) & \text{if } i > \theta \\ 0 & \text{otherwise} \end{cases}$
SCAD (Fan and Li, 2001)	$\begin{cases} \sigma_i(\boldsymbol{X}) & \text{if } \sigma_i(\boldsymbol{X}) \leq 1 \\ \frac{(2\theta\sigma_i(\boldsymbol{X}) - \sigma_i(\boldsymbol{X})^2 - 1)}{2(\theta-1)} & \text{if } 1 < \sigma_i(\boldsymbol{X}) \leq \theta \\ \frac{(\theta+1)^2}{2} & \text{otherwise} \end{cases}$
MCP (Zhang, 2010a)	$\begin{cases} \sigma_i(\boldsymbol{X}) - \frac{\alpha^2}{2\theta} & \text{if } \sigma_i(\boldsymbol{X}) \leq \theta \\ \frac{\theta^2}{2} & \text{otherwise} \end{cases}$

$O(mnk)$. To reduce the computational burden, factored low-rank regularizers are proposed (Srebro et al., 2005; Mazumder et al., 2010; Wang et al., 2021). Specifically, equation (1) is rewritten into a factored form as

$$\min\nolimits_{\boldsymbol{W},\boldsymbol{H}} f(\boldsymbol{W}\boldsymbol{H}^\top) + \lambda \cdot h(\boldsymbol{W}, \boldsymbol{H}), \tag{9}$$

where $\boldsymbol{X}$ is factorized as $\boldsymbol{W}\boldsymbol{H}^\top$ with $\boldsymbol{W} \in \mathbb{R}^{m \times k}$ and $\boldsymbol{H} \in \mathbb{R}^{n \times k}$, h is a regularizer on $\boldsymbol{W}$ and $\boldsymbol{H}$, and $\lambda \geq 0$ is a hyperparameter. When $\lambda = 0$, this reduces to matrix factorization (Vandereycken, 2013; Boumal and Absil, 2015; Tu et al., 2016; Wang et al., 2017). After factorization, gradient descent or alternative minimization are usually used for optimization. When certain conditions (such as proper initialization, restricted strong convexity (RSC) (Negahban and Wainwright, 2012), or restricted isometry property (RIP) (Candès and Tao, 2005)) are met, statistical guarantees can be obtained (Zheng and Lafferty, 2015; Tu et al., 2016; Wang et al., 2017).

Note that in Table 1, the nuclear norm is the only regularizer $r(\boldsymbol{X})$ that has an equivalent factored form $h(\boldsymbol{W}, \boldsymbol{H})$. For a matrix $\boldsymbol{X}$ with ground-truth rank no larger than k, it has been shown that the nuclear norm can be rewritten in a factored form as (Srebro et al., 2005)

$$\|\boldsymbol{X}\|_* = \min\nolimits_{\boldsymbol{X}=\boldsymbol{W}\boldsymbol{H}^\top} \frac{1}{2}\left(\|\boldsymbol{W}\|_F^2 + \|\boldsymbol{H}\|_F^2\right).$$

However, the other nonconvex regularizers need to penalize individual singular values, and so cannot be written in factored form.

2.2 Low-Rank Tensor Learning

A M-order tensor $\mathcal{X}$ has rank one if it can be written as the outer product of M vectors, i.e., $\mathcal{X} = \boldsymbol{x}^1 \circ \boldsymbol{x}^2 \circ \cdots \circ \boldsymbol{x}^M$ where $\circ$ denotes the outer product (i.e., $\mathcal{X}_{i_1,\dots,i_M} = \boldsymbol{x}^1_{i_1} \cdot \boldsymbol{x}^2_{i_2} \cdot \cdots \cdot \boldsymbol{x}^M_{i_M}$). In general, the rank of a tensor $\mathcal{X}$ is the smallest number of rank-one tensors that generate $\mathcal{X}$ as their sum (Kolda and Bader, 2009).

To impose a low-rank structure on tensors, factorization methods (such as the Tucker / CP (Kolda and Bader, 2009; Hong et al., 2020), tensor-train (Oseledets, 2011) and tensor ring (Zhao et al., 2016) decompositions) have been used for low-rank tensor learning. These methods assume that the tensor can be decomposed into low-rank factor matrices (Kolda and Bader, 2009), which are then learned by alternating least squares or coordinate descent (Acar et al., 2011; Xu et al., 2013; Balazevic et al., 2019). Kressner et al. (2014) proposed to utilize the Riemannian structure on the manifold of tensors with fixed multilinear rank, and then perform nonlinear conjugate gradient descent. It can be speeded up by preconditioning (Kasai and Mishra, 2016). However, these models suffer from the problem of local minimum, and have no theoretical guarantee on the convergence rate. Moreover, its per-iteration cost depends on the product of all the mode ranks, and so can be expensive. Thus, they may lead to worse approximations and inferior performance (Tomioka et al., 2011; Liu et al., 2013; Guo et al., 2017).

Due to the above limitations, the nuclear norm, which has been commonly used in low-rank matrix learning, has been extended to the learning of low-rank tensors (Tomioka et al., 2010; Signoretto et al., 2011; Gu et al., 2014; Yuan and Zhang, 2016; Zhang and Aeron, 2017). The most commonly used low-rank tensor regularizer is the following (convex) overlapped nuclear norm:

Definition 3 *(Tomioka et al., 2010) The overlapped nuclear norm of a M-order tensor $\mathcal{X}$ is $\|\mathcal{X}\|_{overlap}$ $= \sum_{i=1}^{M} \lambda_i \left\|\mathcal{X}_{\langle i \rangle}\right\|_*$, where $\{\lambda_i \geq 0\}$ are hyperparameters.*

Note that the nuclear norm is a convex envelop of the matrix rank (Candès and Recht, 2009). Similarly, $\|\mathcal{X}\|_{\text{overlap}}$ is a convex envelop of the tensor rank (Tomioka et al., 2010, 2011). Empirically, $\|\mathcal{X}\|_{\text{overlap}}$ has better performance than the other nuclear norm variants in many tensor applications such as image inpainting (Liu et al., 2013) and multi-relational link prediction (Guo et al., 2017). On the theoretical side, let $\mathcal{X}^*$ be the ground-truth tensor, and $\mathcal{X}$ be the tensor obtained by solving the overlapped nuclear norm regularized problem. The statistical error between $\mathcal{X}^*$ and $\mathcal{X}$ has been established in tensor decomposition (Tomioka et al., 2011) and robust tensor decomposition problems (Gu et al., 2014). Specifically, under the restricted strong convexity (RSC) condition (Negahban and Wainwright, 2012), $\|\mathcal{X}^* - \mathcal{X}\|_F$ can be bounded by $O(\sigma \sum_{i=1}^{M} \sqrt{k_i})$, where σ is the noise level and k_i is the rank of $\mathcal{X}^*_{\langle i \rangle}$. Furthermore, we can see that when $\sigma = 0$ (no noise), exactly recovery can be guaranteed.

2.3 Proximal Average (PA) Algorithm

Let $\mathcal{H}$ be a Hilbert space of $\mathcal{X}$, which can be a scalar/vector/matrix/tensor variable. Consider the following optimization problem:

$$\min\nolimits_{\mathcal{X} \in \mathcal{H}} F(\mathcal{X}) = f(\mathcal{X}) + \sum\nolimits_{i=1}^{K} \lambda_i \, g_i(\mathcal{X}), \tag{10}$$

where f is the loss and each g_i is a regularizer with hyper-parameter $\{\lambda_i\}$. Often, $g(\mathcal{X}) = \sum_{i=1}^{K} \lambda_i \; g_i(\mathcal{X})$ is complicated, and its proximal step does not have a simple solution. Hence, the proximal algorithm cannot be efficiently used. However, it is possible that the proximal step for each individual g_i can be easy obtained. For example, let $g_1(\boldsymbol{X}) = \|\boldsymbol{X}\|_1$ and $g_2(\boldsymbol{X}) = \|\boldsymbol{X}\|_*$. The closed-form solution on the proximal step for g_1 (resp. g_2) is given by the soft-thresholding operator (Efron et al., 2004) (resp. singular value thresholding operator (Cai et al., 2010)). However, the closed-form solution does not exist for the proximal step with $g_1 + g_2$.

In this case, the proximal average (PA) algorithm (Bauschke et al., 2008) can be used instead. The PA algorithm generates $\mathcal{X}_t$'s as

$$\mathcal{X}_t = \sum\nolimits_{i=1}^{K} \mathcal{Y}_t^i, \tag{11}$$

$$\mathcal{Z}_t = \mathcal{X}_t - \frac{1}{\tau}\nabla f(\mathcal{X}_t), \tag{12}$$

$$\mathcal{Y}_{t+1}^i = \text{prox}_{\frac{\lambda_i g_i}{\tau}}(\mathcal{Z}_t), \quad i = 1, \dots, K. \tag{13}$$

As each individual proximal step in (13) is easy, the PA algorithm can be significantly faster than the proximal algorithm (Yu, 2013; Zhong and Kwok, 2014; Yu et al., 2015; Shen et al., 2017). When both f and g are convex, the PA algorithm converges to an optimal solution of (10) with a proper choice of stepsize τ (Yu, 2013; Shen et al., 2017). Recently, the PA algorithm has also been extended to nonconvex f and g_i's (Zhong and Kwok, 2014; Yu et al., 2015). Moreover, τ can be adaptively changed to obtain an empirically faster convergence (Shen et al., 2017).

3. Proposed Algorithm

Analogous to the low-rank matrix completion problem in (1), we consider the following low-rank tensor completion problem with a nonconvex extension of the overlapped nuclear norm:

$$\min_{\mathcal{X}} \sum\nolimits_{\mathbf{\Omega}_{i_1 \dots i_M}=1} \ell\left(\mathcal{X}_{i_1 \dots i_M}, \mathcal{O}_{i_1 \dots i_M}\right) + \sum\nolimits_{i=1}^{D} \lambda_i \, \phi(\mathcal{X}_{\langle i \rangle}). \tag{14}$$

Here, the observed elements are in $\mathcal{O}_{i_1 \dots i_M}$, $\mathcal{X}$ is the tensor to be recovered, $\ell(\cdot, \cdot)$ is a smooth loss function, and ϕ is a nonconvex regularizer in the form in (7). Unlike the overlapped nuclear norm in Definition 3, here we only sum over $D \le M$ modes. This is useful when some modes are already small (e.g., the number of bands in color images), and so do not need to be low-rank regularized. When $D = M$ and $\kappa(\alpha) = \alpha$ in (7), problem (14) reduces to (convex) overlapped nuclear norm regularization. When $D = 1$ and ℓ is the square loss, (14) reduces to the matrix completion problem:

$$\min\nolimits_{\boldsymbol{X} \in \mathbb{R}^{I^1 \times (\frac{I^\pi}{I^1})}} \frac{1}{2} \left\| P_{\mathbf{\Omega}} \left(\boldsymbol{X} - \mathcal{O}_{\langle 1 \rangle} \right) \right\|_F^2 + \lambda_1 \phi(\boldsymbol{X}),$$

which can be solved by the proximal algorithm as in (Lu et al., 2016b; Yao et al., 2019b). In the sequel, we only consider $D > 1$.

3.1 Issues with Existing Solvers

First, consider the case where κ in (7) is convex. While D may not be equal to M, it can be easily shown that existing optimization solvers in (Tomioka et al., 2010; Boyd et al., 2011; Liu et al., 2013) can still be used. However, when κ is nonconvex, the fast low-rank tensor completion (FaLRTC) solver (Liu et al., 2013) cannot be applied, as the dual of (14) cannot be derived. Tomioka et al. (2010) used the alternating direction of multiple multipliers (ADMM) (Boyd et al., 2011) solver for the overlapped nuclear norm. Recently, it is used in (Chen et al., 2020) to solve a special case of (14), in which ϕ is the truncated nuclear norm (TNN) regularizer (see Table 1). Specifically, (14) is first reformulated as

$$\min\nolimits_{\mathcal{X}} \sum\nolimits_{\mathbf{\Omega}_{i_1 \dots i_M}=1} \ell\left(\mathcal{X}_{i_1 \dots i_M}, \mathcal{O}_{i_1 \dots i_M}\right) + \sum\nolimits_{i=1}^{D} \lambda_i \, \phi(\boldsymbol{X}_i) \text{ s.t. } \boldsymbol{X}_i = \mathcal{X}_{\langle i \rangle}, \; i = 1, \dots, D.$$

Using ADMM, it then generates iterates as

$$\mathcal{X}_{t+1} = \arg\min\nolimits_{\mathcal{X}} \sum_{\boldsymbol{\Omega}_{i_1\dots i_M}=1} \ell\left(\mathcal{X}_{i_1\dots i_M}, \mathcal{O}_{i_1\dots i_M}\right) + \frac{\zeta}{2}\sum\nolimits_{i=1}^{D} \left\|(\boldsymbol{X}_i)_t - \mathcal{X}_{\langle i\rangle} + \frac{1}{\zeta}(\boldsymbol{M}_i)_t\right\|_F^2, \tag{15}$$

$$(\boldsymbol{X}_i)_{t+1} = \text{prox}_{\frac{\lambda_i}{\zeta}}((\boldsymbol{X}_i)_{t+1} + \frac{1}{\zeta}(\boldsymbol{M}_i)_t), \quad i = 1, \dots, D, \tag{16}$$

$$(\boldsymbol{M}_i)_{t+1} = (\boldsymbol{M}_i)_t + \frac{1}{\zeta}\left((\boldsymbol{X}_i)_{t+1} - (\mathcal{X}_{\langle i\rangle})_{t+1}\right), \quad i = 1, \dots, D, \tag{17}$$

where $\boldsymbol{M}_i$'s are the dual variables, and $\zeta > 0$. The proximal step in (16) can be computed from Lemma 2. Convergence of this ADMM procedure is guaranteed in (Hong et al., 2016; Wang et al., 2019). However, it does not utilize the sparsity induced by $\boldsymbol{\Omega}$. Moreover, as the tensor $\mathcal{X}$ needs to be folded and unfolded repeatedly, the iterative procedure is expensive, taking $O(I^{\pi})$ space and $O(I^{\pi}\sum_{i=1}^{D} I^i)$ time per iteration.

On the other hand, the proximal algorithm (Section 2.1) cannot be easily used, as the proximal step for $\sum_{i=1}^{D}\lambda_i\phi(\mathcal{X}_{\langle i\rangle})$ is not simple in general.

3.2 Structure-aware Proximal Average Iterations

Note that ϕ in (7) admits a difference-of-convex decomposition (Hartman, 1959; Le Thi and Tao, 2005), i.e., ϕ can be decomposed as $\phi = \phi_1 - \phi_2$ where ϕ_1 and ϕ_2 are convex (Yao et al., 2019b). The proximal average (PA) algorithm (Section 2.3) has been recently extended for nonconvex f and g_i's, where each g_i admits a difference-of-convex decomposition (Zhong and Kwok, 2014). Hence, as (14) is in the form in (10), one can generate the PA iterates as:

$$\mathcal{X}_t = \sum\nolimits_{i=1}^{D} \mathcal{Y}_t^i, \tag{18}$$

$$\mathcal{Z}_t = \mathcal{X}_t - \frac{1}{\tau}\varpi(\mathcal{X}_t), \tag{19}$$

$$\mathcal{Y}_{t+1}^i = \left[\text{prox}_{\frac{\lambda_i\phi}{\tau}}([\mathcal{Z}_t]_{\langle i\rangle})\right]^{\langle i\rangle}, \quad i = 1, \dots, D. \tag{20}$$

where $\xi(\mathcal{X}_t)$ is a sparse tensor with

$$\left[\varpi(\mathcal{X}_t)\right]_{i_1\dots i_M} = \begin{cases} 0 & (i_1, \dots, i_M) \notin \boldsymbol{\Omega} \\ \ell'\left([\mathcal{X}_t]_{i_1\dots i_M}, \mathcal{O}_{i_1\dots i_M}\right) & (i_1, \dots, i_M) \in \boldsymbol{\Omega} \end{cases}. \tag{21}$$

In (20), each individual proximal step can be computed using Lemma 2. However, tensor folding and unfolding are still required. A direct application of the PA algorithm is as expensive as using ADMM (see Table 2).

In the following, we show that by utilizing the "sparse plus low-rank" structure, the PA iterations can be computed efficiently without tensor folding/unfolding. In the earlier conference version (Yao et al., 2019a), we only considered the case $M = 3$. Here, we extend this to $M \geq 3$ by noting that the coordinate format of sparse tensors can naturally handle tensors with arbitrary orders (Section 3.2.1) and the proximal step can be performed without tensor folding/unfolding (Section 3.2.2).

3.2.1 EFFICIENT COMPUTATIONS OF $\mathcal{X}_t$ AND $\mathcal{Z}_t$ IN (18), (19)

First, rewrite (20) as $\mathcal{Y}^i_{t+1} = [\boldsymbol{Y}^i_{t+1}]^{\langle i \rangle}$, where $\boldsymbol{Y}^i_{t+1} = \text{prox}_{\frac{\lambda_i \phi}{\tau}}(\boldsymbol{Z}^i_t)$ and $\boldsymbol{Z}^i_t = [\mathcal{Z}_t]_{\langle i \rangle}$. Recall that $\boldsymbol{Y}^i_t$ obtained from the proximal step is low-rank. Let its rank be k^i_t. Hence, $\boldsymbol{Y}^i_t$ can be represented as $\boldsymbol{U}^i_t(\boldsymbol{V}^i_t)^\top$, where $\boldsymbol{U}^i_t \in \mathbb{R}^{I^i \times k^i_t}$ and $\boldsymbol{V}^i_t \in \mathbb{R}^{(\frac{I^\pi}{I^i}) \times k^i_t}$. In each PA iteration, we avoid constructing the dense $\mathcal{Y}^i_t$ by storing the above low-rank factorized form of $\boldsymbol{Y}^i_t$ instead. Similarly, we also avoid constructing $\mathcal{X}_t$ in (18) by storing it implicitly as

$$\mathcal{X}_t = \sum\nolimits_{i=1}^{D} \big(\boldsymbol{U}^i_t(\boldsymbol{V}^i_t)^\top\big)^{\langle i \rangle}. \tag{22}$$

$\mathcal{Z}_t$ in (19) can then be rewritten as

$$\mathcal{Z}_t = \sum\nolimits_{i=1}^{D} \Big(\boldsymbol{U}^i_t(\boldsymbol{V}^i_t)^\top\Big)^{\langle i \rangle} - \frac{1}{\tau}\xi(\mathcal{X}_t). \tag{23}$$

The sparse tensor $\xi(\mathcal{X}_t)$ in (21) can be constructed efficiently by using the coordinate format[2] (Bader and Kolda, 2007). Using (22), each $[\xi(\mathcal{X}_t)]_{i_1 \dots i_M}$ can be computed by finding the corresponding rows in $\{\boldsymbol{U}^i_t, \boldsymbol{V}^i_t\}$ as shown in Algorithm 1. This takes $O(\sum_{i=1}^{D} k^i_t)$ time.

Algorithm 1 Computing the pth element v_p with index $(i^1_p \dots i^M_p)$ in $\xi(\mathcal{X}_t)$.

```
Require: factorizations $\{\boldsymbol{U}^1_t(\boldsymbol{V}^1_t)^\top, \dots, \boldsymbol{U}^D_t(\boldsymbol{V}^D_t)^\top\}$ of $\boldsymbol{Y}^1_t, \dots, \boldsymbol{Y}^D_t$, and observed elements in $P_{\boldsymbol{\Omega}}(\mathcal{O})$;
$v_p \leftarrow 0$;
for $d = 1, \dots, D$ do
    $\boldsymbol{u}^\top \leftarrow i^d_p$th row of $\boldsymbol{U}^d_t$;
    $\boldsymbol{v}^\top \leftarrow (\sum_{k \neq d}^{M} i^k_p I^\pi + i^d_p)$th row of $\boldsymbol{V}^d_t$;
    $v_p \leftarrow v_p + \boldsymbol{u}^\top \boldsymbol{v}$;
end for
$v_p \leftarrow \ell'(v_p, \mathcal{O}_{i^1_p \dots i^M_p})$;
return $v_p$.
```

3.2.2 EFFICIENT COMPUTATION OF $\mathcal{Y}^i_{t+1}$ IN (20)

Recall that the proximal step in (20) requires SVD, which involves matrix multiplications in the form $\boldsymbol{a}^\top(\mathcal{Z}_t)_{\langle i \rangle}$ (where $\boldsymbol{a} \in \mathbb{R}^{I^i}$) and $(\mathcal{Z}_t)_{\langle i \rangle}\boldsymbol{b}$ (where $\boldsymbol{b} \in \mathbb{R}^{\frac{I^\pi}{I^i}}$). Using the "sparse plus low-rank" structure in (23), these can be computed as

$$\boldsymbol{a}^\top(\mathcal{Z}_t)_{\langle i \rangle} = \big(\boldsymbol{a}^\top \boldsymbol{U}^i_t\big)(\boldsymbol{V}^i_t)^\top + \sum\nolimits_{j \neq i} \boldsymbol{a}^\top \big[(\boldsymbol{U}^j_t(\boldsymbol{V}^j_t)^\top)^{\langle j \rangle}\big]_{\langle i \rangle} - \frac{1}{\tau}\boldsymbol{a}^\top[\xi(\mathcal{X}_t)]_{\langle i \rangle}, \tag{24}$$

and

$$(\mathcal{Z}_t)_{\langle i \rangle}\boldsymbol{b} = \boldsymbol{U}^i_t\big[(\boldsymbol{V}^i_t)^\top \boldsymbol{b}\big] + \sum\nolimits_{j \neq i} \big[(\boldsymbol{U}^j_t(\boldsymbol{V}^j_t)^\top)^{\langle j \rangle}\big]_{\langle i \rangle}\boldsymbol{b} - \frac{1}{\tau}[\xi(\mathcal{X}_t)]_{\langle i \rangle}\boldsymbol{b}. \tag{25}$$

The first terms in (24) and (25) can be easily computed in $O((\frac{I^\pi}{I^i} + I^i)k^i_t)$ space and time. The last terms ($\boldsymbol{a}^\top\ [\xi(\mathcal{X}_t)]_{\langle i \rangle}$ and $[\xi(\mathcal{X}_t)]_{\langle i \rangle}\ \boldsymbol{b}$) are sparse, and can be computed in $O(\|\boldsymbol{\Omega}\|_1)$ space and time

[2] For a sparse M-order tensor, its pth nonzero element is represented in the coordinate format as $(i^1_p, \dots, i^M_p, v_p)$, where $i^1_p, \dots, i^M_p$ are indices on each mode and v_p is the value.

by using sparse tensor packages such as the Tensor Toolbox (Bader and Kolda, 2007). However, a direct computation of the $\boldsymbol{a}^\top[(\boldsymbol{U}_t^j(\boldsymbol{V}_t^j)^\top)^{\langle j\rangle}]_{\langle i\rangle}$ and $[(\boldsymbol{U}_t^j(\boldsymbol{V}_t^j)^\top)^{\langle j\rangle}]_{\langle i\rangle}\boldsymbol{b}$ terms involves tensor folding/unfolding, and is expensive. By examining how elements are ordered by folding/unfolding, the following shows that these multiplications can indeed be computed efficiently without explicit folding/unfolding.

Proposition 4 *Let $\boldsymbol{U} \in \mathbb{R}^{I^j \times k}$, $\boldsymbol{V} \in \mathbb{R}^{(\frac{I^\pi}{I^j})\times k}$, and $\boldsymbol{u}_p$ (resp.$\boldsymbol{v}_p$) be the pth column of $\boldsymbol{U}$ (resp.$\boldsymbol{V}$). For any $i \neq j$, $\boldsymbol{a} \in \mathbb{R}^{I^i}$ and $\boldsymbol{b} \in \mathbb{R}^{\frac{I^\pi}{I^i}}$, we have*

$$\boldsymbol{a}^\top\left[(\boldsymbol{U}\boldsymbol{V}^\top)^{\langle j\rangle}\right]_{\langle i\rangle} = \sum\nolimits_{p=1}^{k} \boldsymbol{u}_p^\top \otimes \left[\boldsymbol{a}^\top \mathit{mat}(\boldsymbol{v}_p; I^i, \bar{I}^{ij})\right], \tag{26}$$

$$\left[(\boldsymbol{U}\boldsymbol{V}^\top)^{\langle j\rangle}\right]_{\langle i\rangle}\boldsymbol{b} = \sum\nolimits_{p=1}^{k} \mathit{mat}\big(\boldsymbol{v}_p; I^i, \bar{I}^{ij}\big)\mathit{mat}\big(\boldsymbol{b}; \bar{I}^{ij}, I^j\big)\boldsymbol{u}_p, \tag{27}$$

where $\otimes$ is the Kronecker product, $\bar{I}^{ij} = I^\pi/(I^i I^j)$, and mat$(\boldsymbol{x}; a, b)$ reshapes a vector $\boldsymbol{x}$ of length ab into a matrix of size $a \times b$.

In the earlier conference version (Yao et al., 2019a), Proposition 3.2 there (not the proposed algorithm) limits the usage to $M = 3$. Without Proposition 4, the algorithm can suffer from expensive computation cost, and thus has no efficiency advantage over the simple use of the PA algorithm. Specifically, when mapping the vector $\boldsymbol{v}_p$ back to a matrix, we do not need to take a special treatment on the size of matrix. The reason is that, $\boldsymbol{v}_p$ has $I_i I_j$ elements and we just need to map it back to a matrix of size $I_i \times I_j$. Thus, we do not have parameters for *mat* operation in the conference version. However, when $M > 3$, $\boldsymbol{v}_p$ has I^π/I^i elements, we need to check whether ideas used in the conference version can be done in a similar way. As a result, we have two more parameters for the *mat* operation here, which customize reshaping matrix to a proper size.

Remark 5 *For a second-order tensor (i.e., matrix case with $M = 2$), Proposition 4 becomes*

$$\boldsymbol{a}^\top\left[(\boldsymbol{U}\boldsymbol{V}^\top)^{\langle 1\rangle}\right]_{\langle 2\rangle} = \sum\nolimits_{p=1}^{k} (\boldsymbol{a}^\top\boldsymbol{v}_p)\boldsymbol{u}_p^\top \quad \text{and} \quad \left[(\boldsymbol{U}\boldsymbol{V}^\top)^{\langle 1\rangle}\right]_{\langle 2\rangle}\boldsymbol{b} = \sum\nolimits_{p=1}^{k} \boldsymbol{v}_p(\boldsymbol{b}^\top\boldsymbol{u}_p).$$

With the usual square loss (i.e., $\sum_{\boldsymbol{\Omega}_{i_1\dots i_M}=1} \ell\left(\mathcal{X}_{i_1\dots i_M}, \mathcal{O}_{i_1\dots i_M}\right)$ in (14) equals $\frac{1}{2}\left\|P_{\boldsymbol{\Omega}}\left(\mathcal{X} - \mathcal{O}\right)\right\|_F^2$), (25) then reduces to (6) when $D = 1$. When $D = 2$, $\sum_{i=1}^{D} \lambda_i\, \phi(\mathcal{X}_{\langle i\rangle})$ in (14) becomes $\lambda_1\phi(\boldsymbol{X}) + \lambda_2\phi(\boldsymbol{X}^\top) = (\lambda_1 + \lambda_2)\phi(\boldsymbol{X})$, and is the same as the corresponding regularizer when $D = 1$. Hence, the reduction from (25) to (6) still holds.

3.2.3 Time and Space Complexities

A direct computation of $\boldsymbol{a}^\top[(\boldsymbol{U}_t^j(\boldsymbol{V}_t^j)^\top)^{\langle j\rangle}]_{\langle i\rangle}$ takes $O(k_t^i I^\pi)$ time and $O(I^\pi)$ space. By using the computation in Proposition 4, these are reduced to $O((\frac{1}{I^i} + \frac{1}{I^j})k_t^i I^\pi)$ time and $O((\frac{1}{I^j} + \frac{1}{I^i})k_t^i I^\pi)$ space. This is also the case for $[(\boldsymbol{U}_t^j(\boldsymbol{V}_t^j)^\top)^{\langle j\rangle}]_{\langle i\rangle}\boldsymbol{b}$. Details are in the following.

	operation	time	space
reshaping	$\text{mat}(\boldsymbol{v}_p; I^j, \bar{I}^{ij})$	$O(\frac{I^\pi}{I^i})$	$O(\frac{I^\pi}{I^i})$
multiplication	$\boldsymbol{a}^\top(\cdot)$	$O(\frac{I^\pi}{I^i})$	$O(\frac{I^\pi}{I^i})$
Kronecker product	$\boldsymbol{u}_p^\top \otimes (\cdot)$	$O(\frac{I^\pi}{I^j})$	$O(\frac{I^\pi}{I^j})$
summation	$\sum_{p=1}^{k_t^i}(\cdot)$	$O(\frac{k_t^i I^\pi}{I^j})$	$O((\frac{1}{I^j} + \frac{1}{I^i})k_t^i I^\pi)$
total for (26)		$O((\frac{1}{I^j} + \frac{k_t^i}{I^j})I^\pi)$	$O((\frac{1}{I^j} + \frac{1}{I^i})k_t^i I^\pi)$

	operation	time	space
reshaping	$\text{mat}\left(\boldsymbol{b}; \bar{I}^{ij}, I^i\right)$	$O(\frac{I^\pi}{I^j})$	$O(\frac{I^\pi}{I^j})$
multiplication	$(\cdot)\boldsymbol{u}_p$	$O(\frac{I^\pi}{I^i})$	$O(\frac{I^\pi}{I^j})$
reshaping	$\text{mat}\left(\boldsymbol{v}_p; I^j, \bar{I}^{ij}\right)$	$O(\frac{I^\pi}{I^i})$	$O(\frac{I^\pi}{I^i})$
multiplication	$\text{mat}\left(\boldsymbol{v}_p; I^j, \bar{I}^{ij}\right)(\cdot)$	$O(\frac{I^\pi}{I^j})$	$O(\frac{I^\pi}{I^i})$
summation	$\sum_{p=1}^{k_t^i}(\cdot)$	$O(k_t^i I^i)$	$O(k_t^i I^i)$
total for (27)		$O((\frac{1}{I^j} + \frac{k_t^i}{I^j})I^\pi)$	$O((\frac{1}{I^j} + \frac{1}{I^i})k_t^i I^\pi)$

Combining the above, and noting that we have to keep the factorized form $\boldsymbol{U}_t^i(\boldsymbol{V}_t^i)^\top$ of $\boldsymbol{Y}_t^i$, computing all the proximal steps in (20) takes

$$O\big(\sum\nolimits_{i=1}^{D}\sum\nolimits_{j\neq i}(\frac{1}{I^i} + \frac{1}{I^j})k_t^i I^\pi + \|\boldsymbol{\Omega}\|_1\big) \tag{28}$$

space and

$$O\big(\sum\nolimits_{i=1}^{D}\sum\nolimits_{j\neq i}(\frac{1}{I^i} + \frac{1}{I^j})k_t^i k_{t+1}^i I^\pi + \|\boldsymbol{\Omega}\|_1(k_t^i + k_{t+1}^i)\big) \tag{29}$$

time. Empirically, as will be seen in the experimental results in Section 5.1.2, k_t^i, $k_{t+1}^i \ll I^i$. Hence, (28) and (29) are much smaller than the complexities with a direct usage of PA and ADMM in Section 3.1 (Table 2).

Table 2: Comparison of the proposed NORT with PA and ADMM for (14) in Section 3.1.

algorithm	complexity		adaptive
	time per iteration	space	momentum
PA (Zhong and Kwok, 2014)	$O(I^\pi\sum_{i=1}^{D} I^i)$	$O(I^\pi)$	×
ADMM (Chen et al., 2020)	$O(I^\pi\sum_{i=1}^{D} I^i)$	$O(I^\pi)$	×
NORT (Algorithm 2)	see (29)	see (28)	√

3.3 Use of Adaptive Momentum

The PA algorithm uses only first-order information, and can be slow to converge empirically (Parikh and Boyd, 2013). To address this problem, we adopt adaptive momentum, which uses historical iterates to speed up convergence. This has been popularly used in stochastic gradient descent (Duchi et al., 2011; Kingma and Ba, 2014), proximal algorithms (Li and Lin, 2015; Yao et al., 2017; Li et al., 2017), cubic regularization (Wang et al., 2020), and zero-order black-box optimization (Chen et al., 2019). Here, we adopt the adaptive scheme in (Li et al., 2017).

The resultant procedure, which will be called NOnconvex Regularized Tensor (NORT), is shown in Algorithm 2. When the extrapolation step $\bar{\mathcal{X}}_t$ achieves a lower function value (step 4), the momentum γ_t is increased to further exploit the opportunity of acceleration; otherwise, γ_t is decayed (step 7). When step 5 is performed, $\mathcal{V}_t = \mathcal{X}_t + \gamma_t(\mathcal{X}_t - \mathcal{X}_{t-1})$. $\mathcal{Z}_t$ in step 9 becomes

$$\mathcal{Z}_t = (1+\gamma_t)\sum\nolimits_{i=1}^{D} \left(\boldsymbol{U}_t^i(\boldsymbol{V}_t^i)^\top\right)^{\langle i\rangle} - \gamma_t \sum\nolimits_{i=1}^{D} \left(\boldsymbol{U}_{t-1}^i(\boldsymbol{V}_{t-1}^i)^\top\right)^{\langle i\rangle} - \frac{1}{\tau}\xi(\mathcal{V}_t), \tag{30}$$

which still has the "sparse plus low-rank" structure. When step 7 is performed, $\mathcal{V}_t = \mathcal{X}_t$, and obviously the resultant $\mathcal{Z}_t$ is "sparse plus low-rank". Thus, the more efficient reformulations in Proposition 4 can be applied in computing the proximal steps at step 11. Note that the rank of $\boldsymbol{X}_{t+1}^i$ in step 11 is determined implicitly by the proximal step. As $\mathcal{X}_t$ and $\mathcal{Z}_t$ are implicitly represented in factorized forms, $\mathcal{V}_t$ and $\bar{\mathcal{X}}_t$ (in step 3) do not need to be explicitly constructed. As a result, the resultant time and space complexities are the same as those in Section 3.2.3.

Algorithm 2 NOnconvex Regularized Tensor (NORT) Algorithm.

```
 1: **Initialize** $\tau > \rho + D\kappa_0$, $\gamma_1, p \in (0,1]$, $\mathcal{X}_0 = \mathcal{X}_1 = 0$ and $t = 1$;
 2: **while** not converged **do**
 3:     $\bar{\mathcal{X}}_t \leftarrow \mathcal{X}_t + \gamma_t(\mathcal{X}_t - \mathcal{X}_{t-1})$;
 4:     **if** $F(\bar{\mathcal{X}}_t) \le F(\mathcal{X}_t)$ **then**
 5:         $\mathcal{V}_t \leftarrow \bar{\mathcal{X}}_t$ and $\gamma_{t+1} \leftarrow \min(\frac{\gamma_t}{p}, 1)$;
 6:     **else**
 7:         $\mathcal{V}_t \leftarrow \mathcal{X}_t$ and $\gamma_{t+1} \leftarrow p\gamma_t$;
 8:     **end if**
 9:     $\mathcal{Z}_t \leftarrow \mathcal{V}_t - \frac{1}{\tau}\xi(\mathcal{V}_t)$; // compute $\xi(\mathcal{V}_t)$ using Algorithm 1
10:     **for** $i = 1, \dots, D$ **do**
11:         $\boldsymbol{X}_{t+1}^i \leftarrow \text{prox}_{\frac{\lambda_i \phi}{\tau}}((\mathcal{Z}_t)_{\langle i\rangle})$; // keep as $\boldsymbol{U}_t^i(\boldsymbol{V}_t^i)^\top$;
12:     **end for**
        // *implicitly construct* $\mathcal{X}_{t+1} \leftarrow \sum_{i=1}^{D}\left(\boldsymbol{U}_{t+1}^i(\boldsymbol{V}_{t+1}^i)^\top\right)^{\langle i\rangle}$;
13:     $t = t + 1$
14: **end while**
15: **return** $\mathcal{X}_t$.
```

3.4 Convergence Properties

In this section, we analyze the convergence properties of the proposed algorithm. As can be seen from (14), we have $f(\mathcal{X}) = \sum_{\boldsymbol{\Omega}_{i_1 \dots i_M}=1} \ell\left(\mathcal{X}_{i_1 \dots i_M}, \mathcal{O}_{i_1 \dots i_M}\right)$ here. Moreover, throughout this section, we assume that the loss f is (Lipschitz-)smooth.

Note that existing proofs for PA algorithm (Yu, 2013; Zhong and Kwok, 2014; Yu et al., 2015) cannot be directly used, as adaptive momentum has not been used with the PA algorithm on nonconvex problems (see Table 2), and also that they do not involve tensor folding/unfolding operations. Our proof strategy will still follow the three main steps in proving convergence of PA:

1. Show that the proximal average step with g_i's in (13) corresponds to a regularizer;
2. Show that this regularizer, when combined with the loss f in (10), serves as a good approximation of the original objective F.
3. Show that the proposed algorithm finds critical points of this approximate optimization problem.

First, the following Proposition shows that the average step in (18) and proximal steps in (20) together correspond to a new regularizer $\bar{g}_\tau$.

Proposition 6 *For any $\tau > 0$, $\sum_{i=1}^{D}[\text{prox}_{\frac{1}{\tau}\lambda_i\phi}([\mathcal{Z}]_{\langle i\rangle})]^{\langle i\rangle} = \text{prox}_{\frac{1}{\tau}\bar{g}_\tau}(\mathcal{Z})$, where*

$$\bar{g}_\tau(\mathcal{X}) = \tau\big[\min_{\{\boldsymbol{X}_d\}:\sum_{d=1}^{D}\boldsymbol{X}_d^{\langle d\rangle}=\mathcal{X}}\sum\nolimits_{d=1}^{D}\big(\frac{1}{2}\|\boldsymbol{X}_d\|_F^2 + \frac{\lambda_d}{\tau}\phi(\boldsymbol{X}_d)\big) - \frac{D}{2}\|\mathcal{X}\|_F^2\big].$$

Analogous to (10), let the objective corresponding to regularizer $\bar{g}_\tau$ be

$$F_\tau(\mathcal{X}) = f(\mathcal{X}) + \bar{g}_\tau(\mathcal{X}). \tag{31}$$

The following bounds the difference between the optimal values ($F^{\min}$ and $F_\tau^{\min}$, respectively) of the objectives F in (10) and F_τ. It thus shows that F_τ serves as an approximation to F, which is controlled by τ.

Proposition 7 $0 \le F^{\min} - F_\tau^{\min} \le \frac{\kappa_0^2}{2\tau}\sum_{i=1}^{D}\lambda_i^2$*, where κ_0 is defined in Assumption 1.*

Before showing the convergence of the proposed algorithm, the following Proposition first shows the condition of being critical points of $F_\tau(\mathcal{X})$.

Proposition 8 *If there exists $\tau > 0$ such that $\tilde{\mathcal{X}} = \text{prox}_{\frac{\bar{g}_\tau}{\tau}}(\tilde{\mathcal{X}} - \nabla f(\tilde{\mathcal{X}})/\tau)$, then $\tilde{\mathcal{X}}$ is a critical point of $F_\tau(\tilde{\mathcal{X}})$.*

Finally, we show how convergence to critical points can be ensured by the proposed algorithm under smooth assumption of loss f (Section 3.4.1) and Kurdyka-Łojasiewicz condition for the approximated objective F_τ (Section 3.4.2).

3.4.1 WITH SMOOTHNESS ASSUMPTION ON LOSS f

The following shows that Algorithm 2 converges to a critical point (Theorem 9).

Theorem 9 *The sequence $\{\mathcal{X}_t\}$ generated from Algorithm 2 has at least one limit point, and all limits points are critical points of $F_\tau(\mathcal{X})$.*

Proof [Sketch, details are in Appendix B.5.] *The main idea is as follows. First, we show that (i) if step 5 is performed, $F_\tau(\mathcal{X}_{t+1}) \le F_\tau(\mathcal{X}_t) - \frac{\eta}{2}\left\|\mathcal{X}_{t+1} - \bar{\mathcal{X}}_t\right\|_F^2$; (ii) if step 7 is performed, we have $F_\tau(\mathcal{X}_{t+1}) \le F_\tau(\mathcal{X}_t) - \frac{\eta}{2}\|\mathcal{X}_{t+1} - \mathcal{X}_t\|_F^2$. Combining the above two conditions, we obtain*

$$\frac{2}{\eta}(F_\tau(\mathcal{X}_1) - F_\tau(\mathcal{X}_{T+1})) \ge \sum\nolimits_{j\in\chi_1(T)}\left\|\mathcal{X}_{t+1} - \bar{\mathcal{X}}_t\right\|_F^2 + \sum\nolimits_{j\in\chi_2(T)}\|\mathcal{X}_{t+1} - \mathcal{X}_t\|_F^2,$$

where $\chi_1(T)$ and $\chi_2(T)$ are partitions of $\{1, \dots, T\}$ such that when $j \in \chi_1(T)$ step 5 is performed, and when $j \in \chi_2(T)$ step 7 is performed. Finally, when $T \rightarrow \infty$, we discuss three cases: (i) $\chi_1(\infty)$ is finite, $\chi_2(\infty)$ is infinite; (ii) $\chi_1(\infty)$ is infinite, $\chi_2(\infty)$ is finite; and (iii) both $\chi_1(\infty)$ and $\chi_2(\infty)$ are infinite. Let $\tilde{\mathcal{X}}$ be a limit point of $\{\mathcal{X}_t\}$, and $\{\mathcal{X}_{j_t}\}$ be a subsequence that converges to $\tilde{\mathcal{X}}$. In all three cases, we show that

$$\lim_{j_t \rightarrow \infty} \|\mathcal{X}_{j_t+1} - \mathcal{X}_{j_t}\|_F^2 = \|\text{prox}_{\frac{\bar{g}_\tau}{\tau}}(\tilde{\mathcal{X}} - \frac{1}{\tau}\nabla f(\tilde{\mathcal{X}})) - \tilde{\mathcal{X}}\|_F^2 = 0.$$

Thus, we must have $\tilde{\mathcal{X}}$ is also a critical point based on Proposition 8. It is easy to see that we have not made any specifications on the limit points. Thus, all limit points are also critical points. ■

Recall that $\mathcal{X}_{t+1}$ is generated from $\mathcal{V}_t$ in steps 9-12 and $\mathcal{X}_{t+1} = \mathcal{V}_t$ indicates convergence to a critical point (Proposition 8). Thus, we can measure convergence of Algorithm 2 by $\|\mathcal{X}_{t+1} - \mathcal{V}_t\|_F$. Corollary 10 shows that a rate of $O(1/T)$ can be obtained, which is also the best possible rate for first-order methods on general nonconvex problems (Nesterov, 2013; Ghadimi and Lan, 2016).

Corollary 10 $\min_{t=1,\dots,T} \frac{1}{2} \|\mathcal{X}_{t+1} - \mathcal{V}_t\|_F^2 \le \frac{1}{\eta T} \left[F_\tau(\mathcal{X}_1) - F_\tau^{\min}\right]$*, where $\eta = \tau - \rho - DL$.*

Remark 11 *A larger τ leads to a better approximation to the original problem F (Proposition 7). However, it also make the stepsize $1/\tau$ smaller (step 11 in Algorithm 2) and thus slower convergence (Corollary 10).*

3.4.2 With Kurdyka-Łojasiewicz Condition on Approximated Objective F_τ

In Section 3.4.1, we showed the convergence results when f is smooth and g is of the form in (7). In this section, we consider using the Kurdyka-Łojasiewicz (KL) condition (Attouch et al., 2013; Bolte et al., 2014) on F_τ, which has been popularly used in nonconvex optimization, particularly in gradient descent (Attouch et al., 2013) and proximal gradient algorithms (Bolte et al., 2014; Li and Lin, 2015; Li et al., 2017). For example, the class of semi-algebraic functions satisfy the KL condition. More examples can be found in (Bolte et al., 2010, 2014).

Definition 12 *A function h: $\mathbb{R}^n \rightarrow (-\infty, \infty]$ has the* uniformized KL property *if for every compact set $\mathcal{S} \in dom(h)$ on which h is a constant, there exist ϵ, $c > 0$ such that for all $\boldsymbol{u} \in \mathcal{S}$ and all $\bar{\boldsymbol{u}} \in \{\boldsymbol{u} : \min_{\boldsymbol{v} \in \mathcal{S}} \|\boldsymbol{u} - \boldsymbol{v}\|_2 \le \epsilon\} \cap \{\boldsymbol{u} : f(\bar{\boldsymbol{u}}) < f(\boldsymbol{u}) < f(\bar{\boldsymbol{u}}) + c\}$, one has $\psi'\left(f(\boldsymbol{u}) - f(\bar{\boldsymbol{u}})\right) \min_{\boldsymbol{v} \in \partial f(\boldsymbol{u})} \|\boldsymbol{v}\|_2 > 1$, where $\psi(\alpha) = \frac{C\alpha^x}{x}$ for some $C > 0$, $\alpha \in [0, c)$ and $x \in (0, 1]$.*

Since the KL property (Attouch et al., 2013; Bolte et al., 2014) does not require h to be smooth or convex, it thus allows convergence analysis under the nonconvex and nonsmooth setting. However, such a property cannot replace the smoothness assumption in Section 3.4.1, as there are example functions which are smooth but fail to meet the KL condition (Section 4.3 of (Bolte et al., 2010)).

The following Theorem extends Algorithm 2 to be used with the uniformized KL property.

Theorem 13 *Assume that F_τ in (31) has the uniformized KL property, and let $r_t = F_\tau(\mathcal{X}_t) - F_\tau^{\min}$. For a sufficiently large t_0,*
a) If x in Definition 12 equals 1, then $r_t = 0$ for all $t \ge t_0$;

b) If $x \in [\frac{1}{2}, 1)$, $r_t \le (\frac{d_1 C^2}{1+d_1 C^2})^{t-t_0} r_{t_0}$ *where* $d_1 = 2(\tau+\rho)^2/\eta$;

c) If $x \in (0, \frac{1}{2})$, $r_t \le (\frac{C}{(t-t_0)d_2(1-2x)})^{1/(1-2x)}$ *where* $d_2 = \min\big(\frac{1}{2d_1 C}, \frac{C}{1-2x}(2^{\frac{2x-1}{2x-2}} - 1)r_{t_0}\big)$.

Proof [Sketch, details are in Appendix B.7.] *The proof idea generally follows that for (Bolte et al., 2014) with a special treatment for* $\mathcal{V}_t$ *here. First, we show*

$$\lim_{t\to\infty} \min\nolimits_{\mathcal{U}_t \in \partial F_\tau(\mathcal{X}_t)} \|\mathcal{U}_t\|_F \le \lim_{t\to\infty} (\tau+\rho) \|\mathcal{X}_{t+1} - \mathcal{V}_t\|_F = 0.$$

Next, using the KL condition, we have

$$1 \le \psi'\left(F_\tau(\mathcal{X}_{t+1}) - F_\tau^{\min}\right)(\tau+\rho)\|\mathcal{X}_{t+1} - \mathcal{V}_t\|_F.$$

Then, let $r_t = F_\tau(\mathcal{X}_t) - F_\tau^{\min}$. *From its definition, we have*

$$r_t - r_{t+1} \ge F_\tau(\mathcal{V}_t) - F_\tau(\mathcal{X}_{t+1}).$$

Combining the above three inequalities, we obtain

$$1 \le \frac{2(\tau+\rho)^2}{\eta}\left[\psi'(r_{t+1})\right]^2 (r_t - r_{t+1}).$$

Since $\phi(\alpha) = \frac{C\alpha^x}{x}$, *then* $\phi'(\alpha) = C\alpha^{x-1}$. *The above inequality becomes* $1 \le d_1 C^2 r_{t+1}^{2x-2}(r_t - r_{t+1})$, *where* $d_1 = \frac{2(\tau+\rho)^2}{\eta}$. *It is shown in (Bolte et al., 2014; Li and Lin, 2015; Li et al., 2017) that for the sequence* $\{r_t\}$ *satisfying the above inequality, we have convergence to zero with the different rates stated in the Theorem.* ■

In Corollary 10 and Theorem 13, the convergence rates do not depend on p, and thus do not demonstrate the effect of momentum. Empirically, the proposed algorithm does have faster convergence when momentum is used, and will be shown in Section 5. This also agrees with previous studies in (Duchi et al., 2011; Kingma and Ba, 2014; Li and Lin, 2015; Li et al., 2017; Yao et al., 2017).

3.5 Statistical Guarantees

Existing statistical analysis on nonconvex regularization has been studied in the context of sparse and low-rank matrix learning. For example, the SCAD (Fan and Li, 2001), MCP (Zhang, 2010a) and capped-ℓ_1 (Zhang, 2010b) penalties have shown to be better than the convex ℓ_1-regularizer on sparse learning problems; and SCAD, MCP and LSP have shown to be better than the convex nuclear norm in matrix completion (Gui et al., 2016). However, these results cannot be extended to the tensor completion problem here as the nonconvex overlapped nuclear norm regularizer in (10) is not separable. Statistical analysis on tensor completion has been studied with CP and Tucker decompositions (Mu et al., 2014), tensor ring decomposition (Huang et al., 2020), convex overlapped nuclear norm (Tomioka et al., 2011), and tensor nuclear norm (Yuan and Zhang, 2016; Cheng et al., 2016). They show that tensor completion is possible under the incoherence condition when the number of observations is sufficiently large. In comparison, in this section, we will (i) use the restricted strong convexity condition (Agarwal et al., 2010; Negahban et al., 2012)) instead of the incoherence condition, and (ii) study nonconvex overlapped nuclear norm regularization.

3.5.1 CONTROLLING THE SPIKINESS AND RANK

In the following, we assume that elements in $\mathbf{\Omega}$ are drawn i.i.d. from the uniform distribution. However, when the sample size $\|\mathbf{\Omega}\|_1 \ll I^\pi$, tensor completion is not always possible. Take the special case of matrix completion as an example. If $\mathcal{X}$ is an almost-zero matrix with only one element being 1, it cannot be recovered unless the nonzero element is observed. However, when $\mathcal{X}$ gets larger, there is a vanishing probability of observing the nonzero element, and so $P_{\mathbf{\Omega}}(\mathcal{X}) = \mathbf{0}$ with high probability (Candès and Recht, 2009; Negahban and Wainwright, 2012).

To exclude tensors that are too "spiky" and allow tensor completion, we introduce

$$m_{\text{spike}}(\mathcal{X}) = \sqrt{I^\pi} \, \|\mathcal{X}\|_{\max} / \|\mathcal{X}\|_F , \tag{32}$$

which is an extension of the measure $\sqrt{I^1 I^2} \, \|\boldsymbol{X}\|_{\max} / \|\boldsymbol{X}\|_F$ in (Negahban and Wainwright, 2012; Gu et al., 2014) for matrices. Note that $m_{\text{spike}}(\mathcal{X})$ is invariant to the scale of $\mathcal{X}$ and $1 \le m_{\text{spike}}(\mathcal{X}) \le \sqrt{I^\pi}$. Moreover, $m_{\text{spike}}(\mathcal{X}) = 1$ when all elements in $\mathcal{X}$ have the same value (least spiky); and $m_{\text{spike}}(\mathcal{X}) = \sqrt{I^\pi}$ when $\mathcal{X}$ has only one nonzero element (spikiest). Similarly, to measure how close is $\mathcal{X}$ to low-rank, we use

$$m_{\text{rank}}(\mathcal{X}) = \sum\nolimits_{i=1}^{D} \alpha_i \left\|\mathcal{X}_{\langle i \rangle}\right\|_* / \|\mathcal{X}\|_F , \tag{33}$$

where $\alpha_i = \lambda_i / \sum_{d=1}^{D} \lambda_d$'s are pre-defined constants depending on the penalty strength. This is also extended from the measure $\|\boldsymbol{X}\|_* / \|\boldsymbol{X}\|_F$ in (Negahban and Wainwright, 2012; Gu et al., 2014) on matrices. Note that $m_{\text{rank}}(\mathcal{X}) \le \sum_{i=1}^{D} \alpha_i \sqrt{\text{rank}(\mathcal{X}_{\langle i \rangle})}$, with equality holds when all nonzero singular values of $\mathcal{X}_i$'s are the same. The target tensor $\mathcal{X}$ should thus have small $m_{\text{spike}}(\mathcal{X})$ and $m_{\text{rank}}(\mathcal{X})$. In (14), assume for simplicity that $D = M$ and $\lambda_i = \lambda$ for $i = 1, \dots, M$. We then have the following constrained version of (14):

$$\min\nolimits_{\mathcal{X}} \frac{1}{2} \|P_{\mathbf{\Omega}}(\mathcal{X} - \mathcal{O})\|_F^2 + \lambda r(\mathcal{X}) \quad \text{s.t.} \quad \|\mathcal{X}\|_{\max} \le C, \tag{34}$$

where $r(\mathcal{X}) = \sum_{i=1}^{D} \phi(\mathcal{X}_{\langle i \rangle})$ encourages $\mathcal{X}$ to be low-rank (i.e., small m_{rank}), and the constraint on $\|\mathcal{X}\|_{\max}$ avoids $\mathcal{X}$ to be spiky (i.e., small m_{spike}).

3.5.2 RESTRICTED STRONG CONVEXITY (RSC)

Following (Tomioka et al., 2011; Negahban and Wainwright, 2012; Loh and Wainwright, 2015; Zhu et al., 2018), we introduce the restricted strong convexity (RSC) condition.

Definition 14 *(Restricted strong convexity (RSC) condition (Agarwal et al., 2010)) Let Δ be an arbitrary M-order tensor. It satisfies the RSC condition if there exist constants $\alpha_1, \alpha_2 > 0$ and $\tau_1, \tau_2 \ge 0$ such that*

$$\|P_{\mathbf{\Omega}}(\Delta)\|_F^2 \ge \begin{cases} \alpha_1 \|\Delta\|_F^2 - \tau_1 \frac{\log I^\pi}{\|\mathbf{\Omega}\|_1} \left(\sum_{i=1}^{M} \left\|\Delta_{\langle i \rangle}\right\|_* \right)^2 & \textit{if } \|\Delta\|_F \le 1 \\ \alpha_2 \|\Delta\|_F^2 - \tau_2 \sqrt{\frac{\log I^\pi}{\|\mathbf{\Omega}\|_1}} \left(\sum_{i=1}^{M} \left\|\Delta_{\langle i \rangle}\right\|_* \right) & \textit{otherwise} \end{cases} . \tag{35}$$

Let $d_i = \frac{1}{2}(I_i + \frac{I^\pi}{I_i})$ for $i = 1, \dots, M$. Consider the set of tensors parameterized by $n, \gamma \ge 0$:

$$\tilde{\mathcal{C}}(n, \gamma) = \left\{ \mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_M}, \mathcal{X} \ne 0 \mid m_{\text{spike}}(\mathcal{X}) \cdot m_{\text{rank}}(\mathcal{X}) \le \frac{1}{\gamma L} \min_{i=1,\dots,M} \sqrt{\frac{n}{d_i \log d_i}} \right\},$$

where L is a positive constant. The following Lemma shows that the RSC condition holds when the low-rank tensor is not too spiky. If the RSC condition does not hold, the tensor can be too hard to be recovered.

Lemma 15 *There exists* $c_0, c_1, c_2, c_3 \geq 0$ *such that* $\forall \Delta \in \tilde{\mathcal{C}}(\|\mathbf{\Omega}\|_1, c_0)$*, where* $\|\mathbf{\Omega}\|_1 > c_3 \max_{i=1,\dots,M}(d_i \log d_i)$*, we have*

$$\frac{\|P_{\mathbf{\Omega}}(\Delta)\|_F}{\|\mathbf{\Omega}\|_1} \geq \frac{1}{8}\|\Delta\|_F \left\{1 - \frac{128L \cdot m_{\textit{spike}}(\Delta)}{\sqrt{\|\mathbf{\Omega}\|_1}}\right\}, \tag{36}$$

with a high probability of at least $1 - \max_{i=1,\dots,M} c_1 \exp(-c_2 d_i \log d_i)$.

Another condition commonly used in low-rank matrix/tensor learning is incoherence (Candès and Recht, 2009; Mu et al., 2014; Yuan and Zhang, 2016), which prevents information of the row/column spaces of the matrix/tensor from being too concentrated in a few rows/columns. However, as discussed in (Negahban and Wainwright, 2012), the RSC condition is less restrictive than the incoherence condition, and can better describe "spikiness" (details are in Appendix A). Thus, we adopt the RSC instead of the incoherence condition here.

3.5.3 MAIN RESULTS

Let $\mathcal{X}^* \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ be the ground-truth tensor, and $\tilde{\mathcal{X}}$ be an estimate of $\mathcal{X}^*$ obtained as a critical point of (34). The following bounds the distance between $\mathcal{X}^*$ and $\tilde{\mathcal{X}}$.

Theorem 16 *Assume that* κ *is differentiable, and the RSC condition holds with* $3\kappa_0 M/4 < \alpha_1$. *Assume that there exists positive constant* $R > 0$ *such that* $\sum_{i=1}^{M} \left\|\mathcal{X}_{\langle i\rangle}\right\|_* \leq R$*, and* λ *satisfies*

$$\frac{4}{\kappa_0}\max\left(\max_i \left\|[P_{\mathbf{\Omega}}(\mathcal{X}^* - \mathcal{O})]_{\langle i\rangle}\right\|_\infty, \alpha_2\sqrt{\log I^\pi / \|\mathbf{\Omega}\|_1}\right) \leq \lambda \leq \frac{\alpha_2}{4R\kappa_0}, \tag{37}$$

where $\|\mathbf{\Omega}\|_1 \geq \max\left(\tau_1^2, \tau_2^2\right) \frac{16R^2 \log(I^\pi)}{\alpha_2^2}$. *Then,*

$$\|\mathcal{X}^* - \tilde{\mathcal{X}}\|_F \leq \frac{\lambda\kappa_0 c_v}{a_v} \sum_{i=1}^{M} \sqrt{k_i}, \tag{38}$$

where $a_v = \alpha_1 - \frac{3M\kappa_0}{4}$, $c_v = 1 - \frac{1}{2M}$*, and* k_i *is the rank of* $\mathcal{X}^*_{\langle i\rangle}$.

Proof [Sketch, details are in Appendix B.9.2.] *The general idea of this proof is inspired from (Loh and Wainwright, 2015).*[3] *There are three main steps:*

- *Let* $\tilde{\mathcal{V}} = \tilde{\mathcal{X}} - \mathcal{X}^*$. *We prove by contradiction that* $\|\tilde{\mathcal{V}}\|_F \leq 1$. *Thus, we only need to consider the first condition in* (35).
- *Let* $h_i(\mathcal{X}) = \phi(\mathcal{X}_{\langle i\rangle})$. *From Assumption 1, we have that* $h_i(\mathcal{X}) + \frac{\mu}{2}\|\mathcal{X}\|_F^2$ *is convex. Using this together with the first condition in* (35)*, we obtain*

$$\left(\alpha_1 - \frac{\mu M}{2}\right)\|\tilde{\mathcal{V}}\|_F^2 \leq \lambda \sum_{i=1}^{M}\left(h_i(\mathcal{X}^*) - h_i(\tilde{\mathcal{X}})\right) + \frac{\lambda\kappa_0}{2}\sum_{i=1}^{M}\|\tilde{\mathcal{V}}_{\langle i\rangle}\|_*.$$

[3] Note, however, that Loh and Wainwright (2015) use different mathematical tools as they consider sparse vectors with separable dimensions, while we consider overlapped tensor regularization with coupled singular values.

- *Using the above inequality and properties of h_i, we obtain*

$$a_v\|\tilde{\mathcal{V}}\|_F^2 \le \lambda \sum\nolimits_{i=1}^{M} b_v h_i(\mathcal{X}^*) - c_v h_i(\tilde{\mathcal{X}}),$$

where $a_v = \alpha_1 - \frac{3M}{4}\kappa_0$, $b_v = 1 + \frac{1}{2M}$ and $c_v = 1 - \frac{1}{2M}$. Finally, using Lemma 30 in Appendix B.9.1 on the above inequality, we have $\|\tilde{\mathcal{V}}\|_F \le \frac{\lambda\kappa_0 c_v}{a_v}\sum_{i=1}^{M}\sqrt{k_i}$. ■

Since $\|\mathcal{X}\|_{\max} \le C$ in (34), we have $\sum_{i=1}^{M}\|\mathcal{X}_{\langle i\rangle}\|_* \le \sum_{i=1}^{M}\sqrt{k_i(I_i + \frac{I^\pi}{I_i})}C$ (as $\|\boldsymbol{X}\|_* \le \sqrt{k}\,\|\boldsymbol{X}\|_F \le \sqrt{mnk}\,\|\boldsymbol{X}\|_{\max}$ for a rank-k matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$). Thus, in Theorem 16, we can take $R = \sum_{i=1}^{M}\sqrt{k_i(I_i + I^\pi/I_i)}C$, which is finite and cannot be arbitrarily large. While we do not have access to the ground-truth $\mathcal{X}^*$ in practice, Theorem 16 shows that the critical point $\tilde{\mathcal{X}}$ can be bounded by a finite distance from $\mathcal{X}^*$, which means that an arbitrary critical point may not be bad. From (38), we can also see that the error $\|\mathcal{X}^* - \tilde{\mathcal{X}}\|_F$ increases with the tensor order M and rank k_i. This is reasonable as tensors with higher orders or larger ranks are usually harder to estimate. Besides, recall that κ_0 in Assumption 1 reflects how nonconvex the function $\kappa(\alpha)$ is; while α_1 in Definition 14 measures strong convexity. Thus, these two quantities play opposing roles in (38). Specifically, a larger α_1 leads to a larger a_v, and subsequently smaller $\|\mathcal{X}^* - \tilde{\mathcal{X}}\|_F$; whereas a larger κ_0 leads to a larger $\frac{\lambda\kappa_0 c_v}{a_v}$, and subsequently larger $\|\mathcal{X}^* - \tilde{\mathcal{X}}\|_F$.

Finally, note that the range for λ is (37) can be empty, which means there can be no λ to ensure Theorem 16. To understand when this can happen, consider the two extreme cases:

C1. There is no noise in the observations, i.e., $P_{\boldsymbol{\Omega}}(\mathcal{X}^* - \mathcal{O}) = 0$: In this case, (37) reduces to

$$\frac{4\alpha_2}{\kappa_0}\sqrt{\log I^\pi/\|\boldsymbol{\Omega}\|_1} \le \lambda \le \frac{\alpha_2}{4R\kappa_0}.$$

Thus, such a λ may not exist when the number of observations $\|\boldsymbol{\Omega}\|_1$ is too small.

C2. All elements are observed: we then have $\|P_{\boldsymbol{\Omega}}(\Delta)\|_F = \|\Delta\|_F$, and so $\alpha_1 = \alpha_2 = 1$ and $\tau_1 = \tau_2 = 0$ in Definition 14. Besides, the noise is not too small, which means $\sqrt{\log I^\pi/\|\boldsymbol{\Omega}\|_1} \le \max_i \|[\mathcal{X}^* - \mathcal{O}]_{\langle i\rangle}\|_\infty$. Then, (37) reduces to

$$\frac{4}{\kappa_0}\max_i \left\|[\mathcal{X}^* - \mathcal{O}]_{\langle i\rangle}\right\|_\infty \le \lambda \le \frac{1}{4R\kappa_0}.$$

Thus, such a λ may not exist when the noise is too high.

Overall, when λ does not exist, it is likely that the tensor completion problem is too hard to have good recovery performance.

On the other hand, there are cases that λ always exists. For example, when $\mathcal{O} = \mathcal{X}^* = 0$, we have $R = 0$. The requirement on λ is then $\frac{4\alpha_2}{\kappa_0}\sqrt{\log I^\pi/\|\boldsymbol{\Omega}\|_1} \le \lambda \le +\infty$, and such a λ always exists.

3.5.4 DEPENDENCIES ON NOISE LEVEL AND NUMBER OF OBSERVATIONS

In this section, we demonstrate how the noise level affects (38). We assume that the observations are contaminated by additive Gaussian noise, i.e.,

$$\mathcal{O}_{i_1\dots i_M} = \begin{cases} \mathcal{X}^*_{i_1\dots i_M} + \xi_{i_1\dots i_M} & \text{if} \quad \boldsymbol{\Omega}_{i_1\dots i_M} = 1 \\ 0 & \text{otherwise} \end{cases}, \tag{39}$$

where $\xi_{i_1 \dots i_M}$ is a random variable following the normal distribution $\mathcal{N}(0, \sigma^2)$. The effects of the noise level σ and number of observations in $\mathbf{\Omega}$ are shown in Corollaries 17 and 18, respectively, which can be derived from Theorem 16.

Corollary 17 *Let $\mathcal{E} = \mathcal{O} - \mathcal{X}^*$ and $\lambda = b_1 \max_i \| [P_{\mathbf{\Omega}}(\mathcal{E})]_{\langle i \rangle} \|_\infty$. When $\|\mathbf{\Omega}\|_1$ is sufficiently large and $b_1 \in [\frac{4}{\kappa_0}, \frac{\alpha_2}{4R\kappa_0 \max_i \|[P_{\mathbf{\Omega}}(\mathcal{E})]_{\langle i \rangle}\|_\infty}]$ (to ensure λ satisfies* (37)*), then $\mathbb{E}[\|\mathcal{X}^* - \tilde{\mathcal{X}}\|_F] \le \sigma \frac{\kappa_0 c_v \sqrt{I^\pi}}{a_v} \sum_{i=1}^M \sqrt{k_i}$.*

Corollary 17 shows that the recovery error decreases as the noise level σ gets smaller, and we can expect an exact recovery when $\sigma = 0$, which is empirically verified in Section 5.1.4. When $\kappa(\alpha) = \alpha$, $r(\mathcal{X})$ becomes the convex overlapping nuclear norm. In this case, Theorem 2 in (Tomioka et al., 2011) shows that the recovery error can be bounded as $\left\|\mathcal{X}^* - \tilde{\mathcal{X}}\right\|_F \le O(\sigma \sum_{i=1}^M \sqrt{k_i})$. Thus, Corollary 17 can be seen as an extension of Theorem 2 in (Tomioka et al., 2011) to the nonconvex case.

Corollary 18 *Let $\lambda = b_3 \sqrt{\log I^\pi / \|\mathbf{\Omega}\|_1}$. Suppose that the noise level σ is sufficiently small and $b_3 \in \left[4, \frac{1}{(4R\sqrt{\log I^\pi / \|\mathbf{\Omega}\|_1})}\right]$ (to ensure λ satisfies* (37)*). Then, $\left\|\mathcal{X}^* - \tilde{\mathcal{X}}\right\|_F \le \frac{b_3 \kappa_0 c_v}{a_v} \sqrt{\frac{\log I^\pi}{\|\mathbf{\Omega}\|_1}} \sum_{i=1}^M \sqrt{k_i}$.*

Corollary 18 shows that the recovery error decays as $\sqrt{\|\mathbf{\Omega}\|_1}$ gets larger. Such a dependency on the number of observed elements is the same as in matrix completion problems with nonconvex regularization (Gui et al., 2016). Corollary 18 can be seen as an extension of Corollary 3.6 in (Gui et al., 2016) to the tensor case.

4. Extensions

In this section, we show how the proposed NORT algorithm in Section 3 can be extended for robust tensor completion (Section 4.1) and tensor completion with graph Laplacian regularization (Section 4.2).

4.1 Robust Tensor Completion

In tensor completion applications such as video recovery and shadow removal, the observed data often have outliers (Candès et al., 2011; Lu et al., 2016a). Instead of using the square loss, more robust losses like the ℓ_1 loss (Candès et al., 2011; Lu et al., 2013; Gu et al., 2014) and capped-ℓ_1 loss (Jiang et al., 2015), are preferred.

In the following, we assume that the loss is of the form $\ell(a) = \kappa_\ell(|a|)$, where κ_ℓ is smooth and satisfies Assumption 1. The optimization problem then becomes

$$\min\nolimits_{\mathcal{X}} F_\ell(\mathcal{X}) = \sum\nolimits_{\mathbf{\Omega}_{i_1 \dots i_M} = 1} \kappa_\ell \left(|\mathcal{X}_{i_1 \dots i_M} - \mathcal{O}_{i_1 \dots i_M}| \right) + \sum\nolimits_{i=1}^D \lambda_i \phi(\mathcal{X}_{\langle i \rangle}). \tag{40}$$

Since $\kappa_\ell(|a|)$ is non-differentiable at $a = 0$, Algorithm 2 cannot be directly used. Motivated by smoothing the ℓ_1 loss with the Huber loss (Huber, 1964) and the difference-of-convex decomposition of κ_ℓ (Le Thi and Tao, 2005; Yao and Kwok, 2018), we propose to smoothly approximate $\kappa_\ell(|a|)$ by

$$\tilde{\kappa}_\ell(|a|; \delta) = \kappa_0 \cdot \tilde{\ell}(|a|; \delta) + \Big(\kappa_\ell(|a|) - \kappa_0 \cdot |a| \Big), \tag{41}$$

where κ_0 is in Assumption 1, δ is a smoothing parameter, and $\tilde{\ell}$ is the Huber loss (Huber, 1964):

$$\tilde{\ell}(a;\delta) = \begin{cases} |a| & |a| \geq \delta \\ \frac{1}{2\delta}a^2 + \frac{1}{2}\delta & \text{otherwise} \end{cases}.$$

The following Proposition shows that $\tilde{\kappa}_\ell$ is smooth, and a small δ ensures that it is a close approximation to κ_ℓ.

Proposition 19 *$\tilde{\kappa}_\ell(|a|;\delta)$ is differentiable and $\lim_{\delta \to 0} \tilde{\kappa}_\ell(|a|;\delta) = \kappa_\ell(|a|)$.*

Problem (40) is then transformed to

$$\min_{\mathcal{X}} \sum\nolimits_{\mathbf{\Omega}_{i_1 \dots i_M}=1} \tilde{\kappa}_\ell(|\mathcal{X}_{i_1 \dots i_M} - \mathcal{O}_{i_1 \dots i_M}|;\delta) + \sum\nolimits_{i=1}^{D} \lambda_i \phi(\mathcal{X}_{\langle i \rangle}). \tag{42}$$

In Algorithm 3, we gradually reduce the smoothing factor in step 3, and use Algorithm 2 to solve the smoothed problem (42) in each iteration.

Algorithm 3 Smoothing NORT for (40).

```
1: Initialize $\delta_0 \in (0,1)$ and $s = 1$;
2: while not converged do
3:     transform to problem (42) with $\tilde{\kappa}_\ell$ using $\delta = (\delta_0)^s$;
4:     obtain $\mathcal{X}_s$ by solving the smoothed objective with Algorithm 2;
5:     $s = s + 1$;
6: end while
7: return $\mathcal{X}_s$.
```

Convergence of Algorithm 3 is ensured in Theorem 20. However, the statistical guarantee in Section 3.5 does not hold as the robust loss is not smooth.

Theorem 20 *The sequence $\{\mathcal{X}_s\}$ generated from Algorithm 3 has at least one limit point, and all limits points are critical points of $F_{\ell\tau}(\mathcal{X}) = \sum_{\mathbf{\Omega}_{i_1 \dots i_M}=1} \kappa_\ell\left(|\mathcal{X}_{i_1 \dots i_M} - \mathcal{O}_{i_1 \dots i_M}|\right) + \bar{g}_\tau(\mathcal{X})$.*

4.2 Tensor Completion with Graph Laplacian Regularization

The graph Laplacian regularizer is often used in tensor completion (Narita et al., 2012; Song et al., 2017). For example, in Section 5.5, we will consider an application in spatial-temporal analysis (Bahadori et al., 2014), namely, climate prediction based on meteorological records. The spatial-temporal data is represented by a 3-order tensor $\mathcal{O} \in \mathbb{R}^{I^1 \times I^2 \times I^3}$, where I^1 is the number of locations, I^2 is the number of time stamps, and I^3 is the number of variables corresponding to climate observations (such as temperature and precipitation). Usually, observations are only available at a few stations, and slices in $\mathcal{O}$ corresponding to the unobserved locations are missing. Learning these entries can then be formulated as a tensor completion problem. To allow generalization to the unobserved locations, correlations among locations have to be leveraged. This can be achieved by using the graph Laplacian regularizer (Belkin et al., 2006) on a graph G with nodes being the locations (Bahadori et al., 2014). Let the affinity matrix of G be $\boldsymbol{A} \in \mathbb{R}^{m \times m}$, and the corresponding graph Laplacian matrix be $\boldsymbol{G} = \boldsymbol{D} - \boldsymbol{A}$, where $D_{ii} = \sum_j A_{ij}$. As the spatial locations are stored along the tensor's first dimension, the graph Laplacian regularizer is defined as $h(\mathcal{X}_{\langle 1 \rangle}) = \text{Tr}(\mathcal{X}_{\langle 1 \rangle}^\top \boldsymbol{G} \mathcal{X}_{\langle 1 \rangle})$, which encourages nearby stations to have similar observations. When $\boldsymbol{G} = \boldsymbol{I}$, it reduces to the commonly used Frobenius-norm regularizer $\|\mathcal{X}\|_F^2$ (Hsieh et al., 2015). With regularizer $h(\mathcal{X}_{\langle 1\rangle})$, problem (14) is then extended to:

$$\min\nolimits_{\mathcal{X}} \sum\nolimits_{\mathbf{\Omega}_{i_1\dots i_M}=1} \ell\left(\mathcal{X}_{i_1\dots i_M}, \mathcal{O}_{i_1\dots i_M}\right) + \sum\nolimits_{i=1}^{D} \lambda_i \, \phi(\mathcal{X}_{\langle i\rangle}) + \mu \, h(\mathcal{X}_{\langle 1\rangle}), \tag{43}$$

where μ is a hyperparameter.

Using the PA algorithm, it can be easily seen that the updates in (18)-(20) for $\mathcal{X}_t$ and $\mathcal{Y}_t$ remain the same, but that for $\mathcal{Z}_t$ becomes

$$\mathcal{Z}_t = \mathcal{X}_t - \frac{1}{\tau}\xi(\mathcal{X}_t) + \mu \nabla \text{Tr}(\mathcal{X}_{\langle 1\rangle}^\top \boldsymbol{G} \mathcal{X}_{\langle 1\rangle}).$$

To maintain efficiency of NORT, the key is to exploit the low-rank structures. Using (22), $\mathcal{Z}_t$ can be written as

$$\mathcal{Z}_t \;\; = \;\; \sum\nolimits_{i=1}^{D} (\boldsymbol{U}_t^i (\boldsymbol{V}_t^i)^\top)^{\langle i\rangle} - \frac{1}{\tau}\xi(\mathcal{X}_t) - \mu [\boldsymbol{G}\mathcal{X}_{\langle 1\rangle}]^{\langle 1\rangle}. \tag{44}$$

$\boldsymbol{G}\mathcal{X}_{\langle 1\rangle}$ can also be rewritten in low-rank form as

$$\boldsymbol{G}\mathcal{X}_{\langle 1\rangle} = (\boldsymbol{G}\boldsymbol{U}_t^1)(\boldsymbol{V}_t^1)^\top + \boldsymbol{G}\sum\nolimits_{j\neq 1} \left[(\boldsymbol{U}_t^j (\boldsymbol{V}_t^j)^\top)^{\langle j\rangle}\right]_{\langle 1\rangle}.$$

For matrix multiplications of the forms $\boldsymbol{a}^\top(\mathcal{Z}_t)_{\langle i\rangle}$ and $(\mathcal{Z}_t)_{\langle i\rangle}\boldsymbol{b}$ involved in the SVD of the proximal step, we have

$$\boldsymbol{a}^\top(\mathcal{Z}_t)_{\langle i\rangle} = (\boldsymbol{a}^\top(\boldsymbol{I}-\mu\boldsymbol{G})\boldsymbol{U}_t^i)(\boldsymbol{V}_t^i)^\top + \sum\nolimits_{j\neq i} \boldsymbol{a}^\top(\boldsymbol{I}-\mu\boldsymbol{G})\left[(\boldsymbol{U}_t^j(\boldsymbol{V}_t^j)^\top)^{\langle j\rangle}\right]_{\langle i\rangle} - \frac{1}{\tau}\boldsymbol{a}^\top[\xi(\mathcal{X}_t)]_{\langle i\rangle}, \tag{45}$$

and

$$(\mathcal{Z}_t)_{\langle i\rangle}\boldsymbol{b} = (\boldsymbol{I}-\mu\boldsymbol{G})\boldsymbol{U}_t^i\left[(\boldsymbol{V}_t^i)^\top\boldsymbol{b}\right] + (\boldsymbol{I}-\mu\boldsymbol{G})\sum\nolimits_{j\neq i}\left[(\boldsymbol{U}_t^j(\boldsymbol{V}_t^j)^\top)^{\langle j\rangle}\right]_{\langle i\rangle}\boldsymbol{b} - \frac{1}{\tau}[\xi(\mathcal{X}_t)]_{\langle i\rangle}\boldsymbol{b}. \tag{46}$$

Thus, one can still leverage the efficient computational procedures in Proposition 4 to compute $\hat{\boldsymbol{a}}^\top[(\boldsymbol{U}_t^j(\boldsymbol{V}_t^j)^\top)^{\langle j\rangle}]_{\langle i\rangle}$, where $\hat{\boldsymbol{a}}^\top = \boldsymbol{a}^\top(\boldsymbol{I}-\mu\boldsymbol{G})$ in (45), and $[(\boldsymbol{U}_t^j(\boldsymbol{V}_t^j)^\top)^{\langle j\rangle}]_{\langle i\rangle}\boldsymbol{b}$ in (46).

By taking $f(\mathcal{X}) = \sum_{\mathbf{\Omega}_{i_1\dots i_M}=1} \ell\left(\mathcal{X}_{i_1\dots i_M}, \mathcal{O}_{i_1\dots i_M}\right) + \mu \, h(\mathcal{X}_{\langle 1\rangle})$, it is easy to see that the statistical analysis in Section 3.4 and convergence analysis in Section 3.5 still hold.

5. Experiments

In this section, experiments are performed on both synthetic (Section 5.1) and real-world data sets (Sections 5.2-5.5), using a PC with Intel-i9 CPU and 32GB memory. To reduce statistical variation, all results are averaged over five repetitions.

5.1 Synthetic Data

We follow the setup in (Song et al., 2017). First, we generate a 3-order tensor (i.e., $M = 3$) $\bar{\mathcal{O}} = \sum_{i=1}^{r_g} s_i(\boldsymbol{a}_i \circ \boldsymbol{b}_i \circ \boldsymbol{c}_i)$, where $\boldsymbol{a}_i \in \mathbb{R}^{I^1}$, $\boldsymbol{b}_i \in \mathbb{R}^{I^2}$ and $\boldsymbol{c}_i \in \mathbb{R}^{I^3}$, $\circ$ denotes the outer product (i.e., $[\boldsymbol{a} \circ \boldsymbol{b} \circ \boldsymbol{c}]_{ijk} = a_i b_j c_k$). r_g denotes the ground-truth rank and is set to 5, with all k_i's equal to $r_g = 5$. All elements in $\boldsymbol{a}_i$'s, $\boldsymbol{b}_i$'s, $\boldsymbol{c}_i$'s and s_i's are sampled independently from the standard normal distribution. Each element of $\bar{\mathcal{O}}$ is then corrupted by noise from $\mathcal{N}(0, 0.01^2)$ to form $\mathcal{O}$. A total of $\|\boldsymbol{\Omega}\|_1 = \frac{I^3}{r_g} \sum_{i=1}^{3} I^i \log(I^\pi)$ random elements are observed from $\mathcal{O}$. We use 50% of them for training, and the remaining 50% for validation. Testing is evaluated on the unobserved elements in $\bar{\mathcal{O}}$.

We use the square loss and three nonconvex penalties: capped-ℓ_1 (Zhang, 2010a), LSP (Candès et al., 2008) and TNN (Hu et al., 2013). The following methods are compared:

- PA-APG (Yu, 2013), which solves the convex overlapped nuclear norm minimization problem;
- GDPAN (Zhong and Kwok, 2014), which directly applies the PA algorithm to (14) as described in (18)-(20);
- LRTC (Chen et al., 2020), which uses ADMM (Boyd et al., 2011) on (14) as described in (15)-(17); and
- The proposed NORT algorithm (Algorithm 2), and its slower variant without adaptive momentum (denoted "sNORT"). Recall from Corollary 10 that τ has to be larger than $\rho + D\kappa_0$. However, a large τ leads to slow convergence (Remark 11). Hence, we set $\tau = 1.01(\rho + D\kappa_0)$. Moreover, as in (Li et al., 2017), we set $\gamma_1 = 0.1$ and $p = 0.5$ in Algorithm 2.

All algorithms are implemented in Matlab, with sparse tensor and matrix operations performed via Mex files in C. All hypeprparamters (including the λ_i's in (14) and hyperparameter in the baselines) are tuned by grid search using the validation set. We early stop training if the relative change of objective in consecutive iterations is smaller than 10^{-4} or reaching the maximum of 2000 iterations.

5.1.1 Recovery Performance Comparison

In this experiment, we set $I^1 = I^2 = I^3 = \hat{c}$, where $\hat{c} = 200$ and 400. Following (Lu et al., 2016b; Yao et al., 2017, 2019b), performance is evaluated by the (i) root-mean-square-error on the unobserved elements: RMSE $= \left\|P_{\bar{\boldsymbol{\Omega}}}(\mathcal{X} - \bar{\mathcal{O}})\right\|_F / \left\|\bar{\boldsymbol{\Omega}}\right\|_1^{0.5}$, where $\mathcal{X}$ is the low-rank tensor recovered, and $\bar{\boldsymbol{\Omega}}$ contains the unobserved elements in $\bar{\mathcal{O}}$; (ii) CPU time; and (iii) space, which is measured as the memory used by MATLAB when running each algorithm.

Results on RMSE and space are shown in Table 3. We can see that the nonconvex regularizers (capped-ℓ_1, LSP and TNN, with methods GDPAN, LRTC, sNORT and NORT) all yield almost the same RMSE, which is much lower than that of using the convex nuclear norm regularizer in PA-APG. As for the space required, sNORT and NORT take orders of magnitude smaller space than the others. LRTC takes the largest space due to the use of multiple auxiliary and dual variables. Convergence of the optimization objective is shown in Figure 1. As can be seen, NORT is the fastest, followed by sNORT and GDPAN, while LRTC is the slowest. These demonstrate the benefits of avoiding repeated tensor folding/unfolding operations and faster convergence of the proximal average algorithm.

5.1.2 Ranks during Iteration

Unlike factorization methods which explicitly constrain the iterate's rank, in NORT (Algorithm 2), this is only implicitly controlled by the nonconvex regularizer. As shown in Table 2, having a large rank during the iteration may affect the efficiency of NORT. Figure 2 shows the ranks of $(\mathcal{Z}_t)_{\langle i \rangle}$ and

Table 3: Testing RMSE and space required for the synthetic data.

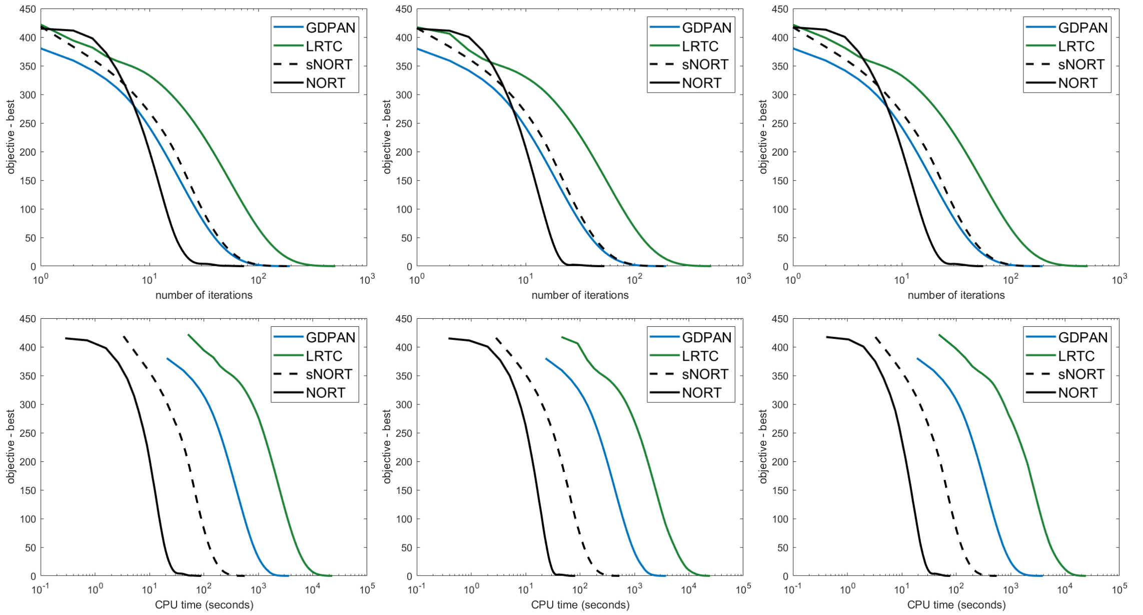
		$\hat{c} = 200$ (sparsity:4.77%)		$\hat{c} = 400$ (sparsity:2.70%)	
		RMSE	space (MB)	RMSE	space (MB)
convex	PA-APG	0.0110±0.0007	600.8±70.4	0.0098±0.0001	4804.5±598.2
nonconvex (capped-ℓ_1)	GDPAN	**0.0010±0.0001**	423.1±11.4	**0.0006±0.0001**	3243.3±489.6
	LRTC	**0.0010±0.0001**	698.9±21.5	**0.0006±0.0001**	5870.6±514.0
	sNORT	**0.0010±0.0001**	**10.1±0.1**	**0.0006±0.0001**	**44.6±0.3**
	NORT	**0.0009±0.0001**	14.4±0.1	**0.0006±0.0001**	66.3±0.6
nonconvex (LSP)	GDPAN	**0.0010±0.0001**	426.9±9.7	**0.0006±0.0001**	3009.3±376.2
	LRTC	**0.0010±0.0001**	714.0±24.1	**0.0006±0.0001**	5867.7±529.1
	sNORT	**0.0010±0.0001**	10.8±0.1	**0.0006±0.0001**	**44.6±0.2**
	NORT	**0.0010±0.0001**	14.0±0.1	**0.0006±0.0001**	62.1±0.5
nonconvex (TNN)	GDPAN	**0.0010±0.0001**	427.3±10.1	**0.0006±0.0001**	3009.2±412.2
	LRTC	**0.0010±0.0001**	759.0±24.3	**0.0006±0.0001**	5865.5±519.3
	sNORT	**0.0010±0.0001**	**10.2±0.1**	**0.0006±0.0001**	**44.7±0.2**
	NORT	**0.0010±0.0001**	14.4±0.2	**0.0006±0.0001**	63.1±0.6

$\boldsymbol{X}^i_{t+1}$ at step 11 of Algorithm 2. As can be seen, the ranks of the iterates remain small compared with the tensor size ($\hat{c} = 400$). Moreover, the ranks of $\boldsymbol{X}^1_{t+1}$, $\boldsymbol{X}^2_{t+1}$, and $\boldsymbol{X}^3_{t+1}$ all converge to the true rank (i.e., 5) of the ground-truth tensor.

5.1.3 QUALITY OF CRITICAL POINTS

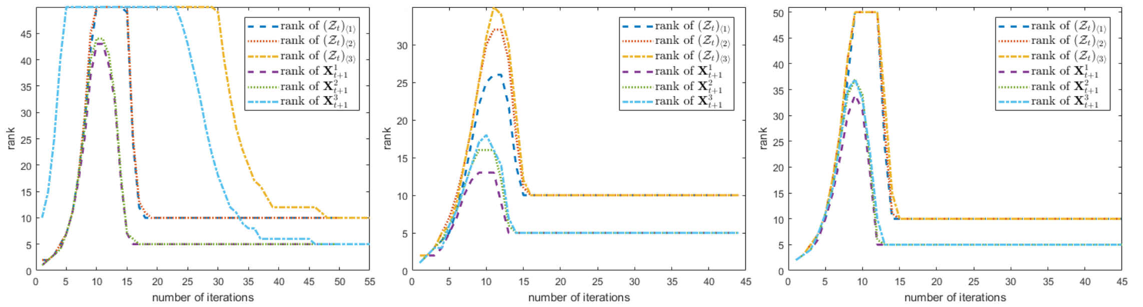
In this experiment, we empirically validate the statistical performance of critical points analysed in Theorem 16. Note that $\mathcal{X}_0$ and $\mathcal{X}_1$ are initialized as the zero tensor in Algorithm 2, and $\mathcal{X}_t$ is implicitly stored by a summation of D factorized matrices in (22). We randomly generate $\mathcal{X}_0 = \mathcal{X}_1 = \sum_{i=1}^{D} \left(\boldsymbol{u}^i(\boldsymbol{v}^i)^\top\right)^{\langle i \rangle}$, where elements in $\boldsymbol{u}^i$'s and $\boldsymbol{v}^i$'s follow $\mathcal{N}(0, 1)$. The statistical error is measured as the RMSE between $\mathcal{X}_t$ during iterating of NORT (Algorithm 2) and the underlying ground-truth $\mathcal{X}^*$ (i.e., $\|\mathcal{X}_t - \mathcal{X}^*\|_F^2$), while the optimization error is measured as the RMSE between iterate $\mathcal{X}_t$ and the globally optimal solution $\dot{\mathcal{X}}$ of (14) (i.e., $\|\mathcal{X}_t - \dot{\mathcal{X}}\|_F^2$). We use the same experimental setup as in Section 5.1.1. As the exact $\dot{\mathcal{X}}$ is not known, it is approximated by the $\tilde{\mathcal{X}}$ which obtains the lowest training objective value over 20 repetitions.

Figure 3 shows the statistical error versus optimization error obtained by NORT with the (smooth) LSP regularizer and (nonsmooth) capped-ℓ_1 regularizer. While both the statistical and optimization errors decrease with more iterations, the statistical error is generally larger than the optimization error since we may not have exact recovery when noise is present. Moreover, the optimization errors for different runs terminate at different values, indicating that NORT indeed converges to different local solutions. However, all these have similar statistical errors, which validates Theorem 16. Finally, while the capped-ℓ_1 regularizer does not satisfy Assumption 1 (which is required by Theorem 16), Figure 3(b) still shows a similar pattern as Figure 3(a). This helps explain the good



(a) capped-ℓ_1. (b) LSP. (c) TNN.

Figure 1: Convergence of the objective versus number of iterations (top) and CPU time (bottom) on the synthetic data (with $\hat{c} = 400$).



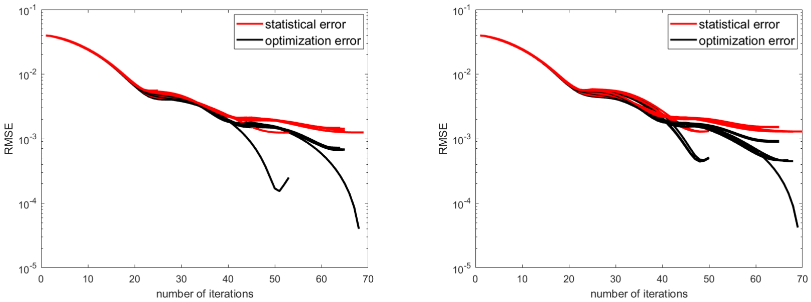
(a) capped-ℓ_1. (b) LSP. (c) TNN.

Figure 2: Ranks of $\{(\mathcal{Z}_t)_{\langle i \rangle}, \boldsymbol{X}^i_{t+1}\}_{i=1,2,3}$ versus number of iterations on synthetic data (with $\hat{c} = 400$).

empirical performance obtained by the capped-ℓ_1 regularizer (Jiang et al., 2015; Lu et al., 2016b; Yao et al., 2019b).

5.1.4 Effects of Noise Level and Number of Observations

In this section, we show the effects of noise level σ and number of observed elements $\|\boldsymbol{\Omega}\|_1$ on the testing RMSE and training time. We use the same experimental setup as in Section 5.1.1. Since PA-APG is much worse (see Table 3) while LRTC and sNORT are slower than NORT (see Figure 1), we only use GDPAN as comparison baseline.

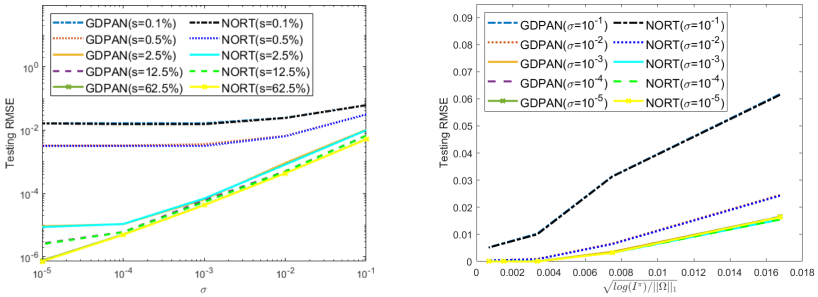


(a) LSP.

(b) capped-ℓ_1.

Figure 3: Statistical error (red) and optimization error (black) versus the number of NORT iterations (with $\hat{c} = 400$) from 20 runs of NORT (with different random seeds).

Figure 4(a) shows the testing RMSE with σ at different $\|\boldsymbol{\Omega}\|_1$'s (here, we plot $s = \|\boldsymbol{\Omega}\|_1 / I^\pi$). As can be seen, the curves show a linear dependency on σ when $\|\boldsymbol{\Omega}\|_1$ is sufficiently large, which agrees with Corollary 17. Figure 4(b) shows the testing RMSE versus $\sqrt{\log I^\pi / \|\boldsymbol{\Omega}\|_1}$ at different σ's. As can be seen, there is a linear dependency when the noise level σ is small, which agrees with Corollary 18. Finally, note that NORT and GDPAN obtain very similar testing RMSEs as both solve the same objective (but with different algorithms).

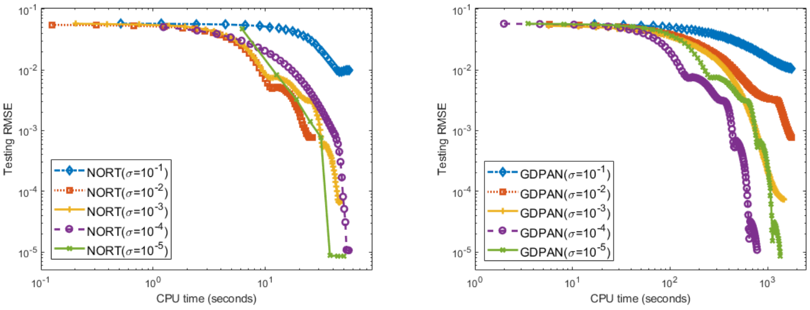


(a) Different noise levels.

(b) Different numbers of observations.

Figure 4: Effect of noise level and number of observations on the testing RMSE on the synthetic data (with $\hat{c} = 400$). Note that NORT and GDPAN obtain similar performance and their curves overlap with each other.

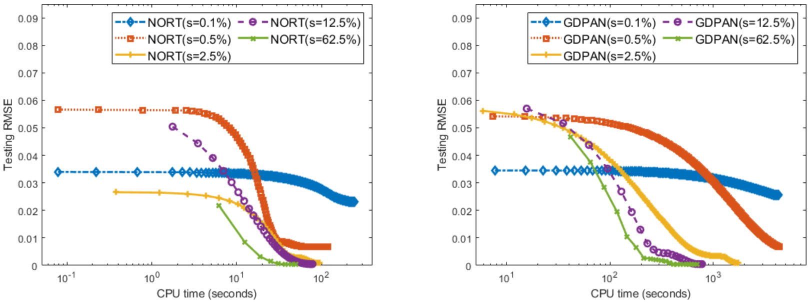
Figure 5 shows the effects of noise level on the convergence of testing RMSE versus (training) CPU time. As can be seen, testing RMSEs generally terminates at a higher level when the noise level gets larger, and NORT is much faster than GDPAN under all noise level.



(a) NORT. (b) GDPAN.

Figure 5: Effects of the noise level on the convergence on synthetic data (with $\hat{c} = 400$, $s = 2.5\%$).

Figure 6 shows the effects of numbers of observations on the convergence of testing RMSE versus (training) CPU time. First, we can see that NORT is much faster than GDPAN under various numbers of observations. Then, when s gets smaller and the tensor completion problem is more ill-posed, more iterations are needed by both NORT and GDPAN, which makes them take more time to converge.



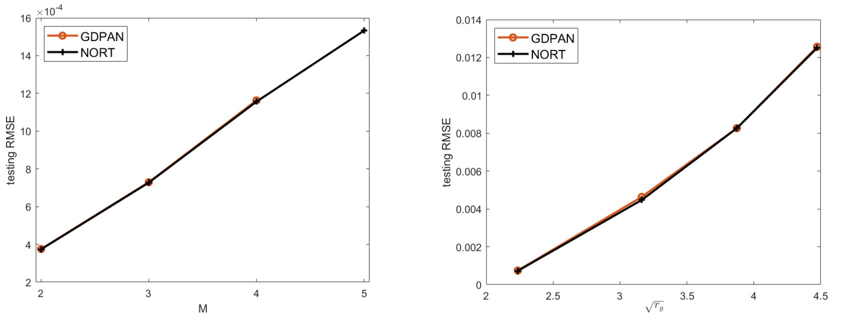
(a) NORT. (b) GDPAN.

Figure 6: Effect of the number of observations on the convergence on synthetic data (with $\hat{c} = 400$, $\sigma = 10^{-2}$).

5.1.5 EFFECTS OF TENSOR ORDER AND RANK

In this experiment, we use a similar experimental setup as in Section 5.1.1, except that the tensor order M is varied from 2 to 5. As high-order tensors have large memory requirements, while we always set $I^1 = I^2 = I^3 = \hat{c} = 400$, we set $I^4 = 5$ when $M = 4$ and $I^4 = I^5 = 5$ when $M = 5$. Figure 7(a) shows the testing RMSE versus M. As can be seen, the error grows almost linearly, which agrees with Theorem 16. Moreover, note that at $M = 5$, GDPAN runs out of memory because it needs to maintain dense tensors in each iteration.

Figure 7(b) shows the testing RMSE w.r.t. $\sqrt{r_g}$ (where r_g is the ground-truth tensor rank). As can be seen, the error grows linearly w.r.t. $\sqrt{r_g}$, which again agrees with Theorem 16.

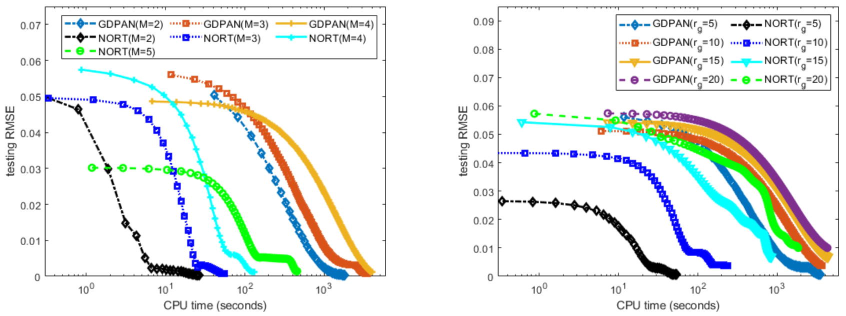


(a) Effect of tensor order M ($r_g = 5$).

(b) Effect of ground-truth rank r_g at $M = 3$ (the corresponding r_g values are 5, 10, 15 and 20).

Figure 7: Effects of tensor order and ground-truth rank on the testing RMSE on the synthetic data (with $\hat{c} = 400$). Note that NORT and GDPAN obtain similar performance and their curves overlap with each other.

Figure 8(a) shows the convergence of testing RMSE versus (training) CPU time at different tensor orders. As can be seen, while both GDPAN and NORT need more time to converge for higher-order tensors, NORT is consistently faster than GDPAN. Figure 8(b) shows the convergence of testing RMSE at different ground-truth ranks. As can be seen, while NORT is still faster than GDPAN at different ground-truth tensor ranks (r_g), the relative speedup gets smaller when r_g gets larger. This is because NORT needs to construct sparse tensors (e.g., Algorithm 1) before using them for multiplications, and empirically, the handling of sparse tensors requires more time on memory addressing as the rank increases (Bader and Kolda, 2007).



(a) Effect of tensor order M ($r_g = 5$).

(b) Effect of ground-truth rank ($M\!=\!3$).

Figure 8: Effects of tensor order and ground-truth rank on the convergence on the synthetic data (with $\hat{c} = 400$). GDPAN runs out of memory when $M = 5$.

5.2 Tensor Completion Applications

In this section, we use the square loss. As different nonconvex regularizers have similar performance, we will only use LSP in the sequel. The proposed NORT algorithm is compared with:[4]

[4] We used our own implementations of LRTC, PA-APG and GDPAN as their codes are not publicly available.

Table 4: Algorithms compared on the real-world data sets.

	algorithm	model	basic solver
convex	ADMM (Tomioka et al., 2010)	overlapped nuclear norm	ADMM
	FaLRTC (Liu et al., 2013)		accelerated proximal algorithm on dual problem
	PA-APG (Yu, 2013)		accelerated PA algorithm
	FFW (Guo et al., 2017)	latent nuclear norm	efficient Frank-Wolfe algorithm
	TR-MM (Nimishakavi et al., 2018)	squared latent nuclear norm	Riemannian optimization on dual problem
	TenNN (Zhang and Aeron, 2017)	tensor-SVD	ADMM
factorization	RP (Kasai and Mishra, 2016)	Turker decomposition	Riemannian preconditioning
	TMac (Xu et al., 2013)	multiple matrices factorization	alternative minimization
	CP-WOPT (Hong et al., 2020)	CP decomposition	nonlinear conjugate gradient
	TMac-TT (Bengua et al., 2017)	tensor-train decomposition	alternative minimization
	TRLRF (Yuan et al., 2019)	tensor-ring decomposition	ADMM
non-convex	GDPAN (Zhong and Kwok, 2014)	nonconvex overlapped nuclear norm regularization	nonconvex PA algorithm
	LRTC (Chen et al., 2020)		ADMM
	NORT (Algorithm 2)		proposed algorithm

(i) algorithms for various convex regularizers including: ADMM (Boyd et al., 2011)[5], PA-APG (Yu, 2013), FaLRTC (Liu et al., 2013)[6], FFW (Guo et al., 2017)[7], TR-MM (Nimishakavi et al., 2018)[8], and TenNN (Zhang and Aeron, 2017)[9];

[5] https://web.stanford.edu/~boyd/papers/admm/

[6] https://github.com/andrewssobral/mctc4bmi/tree/master/algs_tc/LRTC

[7] https://github.com/quanmingyao/FFWTensor

[8] https://github.com/madhavcsa/Low-Rank-Tensor-Completion

[9] http://www.ece.tufts.edu/~shuchin/software.html

(ii) factorization-based algorithms including: RP (Kasai and Mishra, 2016)[10], TMac (Xu et al., 2013)[11], CP-WOPT (Hong et al., 2020)[12], TMac-TT (Bengua et al., 2017)[13], and TRLRF (Yuan et al., 2019)[14];

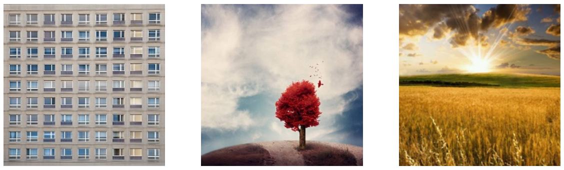
(iii) algorithms that can handle nonconvex regularizers including GDPAN (Zhong and Kwok, 2014) and LRTC (Chen et al., 2020).

More details are in Table 4. We do not compare with (i) sNORT, as it has already been shown to be slower than NORT; (ii) iterative hard thresholding (Rauhut et al., 2017), as its code is not publicly available and the more recent TMac-TT solves the same problem; (iii) the method in (Bahadori et al., 2014), as it can only deal with cokriging and forecasting problems.

Unless otherwise specified, performance is evaluated by (i) root-mean-squared-error on the unobserved elements: RMSE $= \|P_{\mathbf{\Omega}^\perp}(\mathcal{X} - \mathcal{O})\|_F \,/\, \left\|\mathbf{\Omega}^\perp\right\|_1^{0.5}$, where $\mathcal{X}$ is the low-rank tensor recovered, and $\mathbf{\Omega}^\perp$ contains the unobserved elements in $\mathcal{O}$; and (ii) CPU time.

5.2.1 Color Images

We use the *Windows*, *Tree* and *Rice* images from (Hu et al., 2013), which are resized to $1000 \times 1000 \times 3$ (Figure 9). Each pixel is normalized to $[0, 1]$. We randomly sample 5% of the pixels for training, which are then corrupted by Gaussian noise $\mathcal{N}(0, 0.01^2)$; and another 5% clean pixels are used for validation. The remaining unseen clean pixels are used for testing. Hyperparameters of the various methods are tuned by using the validation set.



(a) *Windows*. (b) *Tree*. (c) *Rice*.

Figure 9: Color images used in experiments. All are of size $1000 \times 1000 \times 3$.

Table 5 shows the RMSE results. As can be seen, the best convex methods (PA-APG and FaLRTC) are based on the overlapped nuclear norm. This agrees with our motivation to build a nonconvex regularizer based on the overlapped nuclear norm. GDPAN, LRTC and NORT have similar RMSEs, which are lower than those by convex regularization and the factorization approach. Convergence of the testing RMSE is shown in Figure 10. As can be seen, while ADMM solves the same convex model as PA-APG and FaLRTC, it has slower convergence. FFW, RP and TR-MM are very fast but their testing RMSEs are higher than that of NORT. By utilizing the “sparse plus low-rank” structure and adaptive momentum, NORT is more efficient than GDPAN and LRTC.

Finally, Table 6 compares NORT with PA-APG and RP, which are the best convex-regularization-based and factorization-based algorithms, respectively, as observed in Table 5. Table 6 shows the

[10] https://bamdevmishra.in/codes/tensorcompletion/

[11] http://www.math.ucla.edu/~wotaoyin/papers/tmac_tensor_recovery.html

[12] https://www.sandia.gov/~tgkolda/TensorToolbox/

[13] https://sites.google.com/site/jbengua/home/projects/efficient-tensor-completion-for-color-image-and-video-recovery-low-rank-tensor-train

[14] https://github.com/yuanlonghao/TRLRF

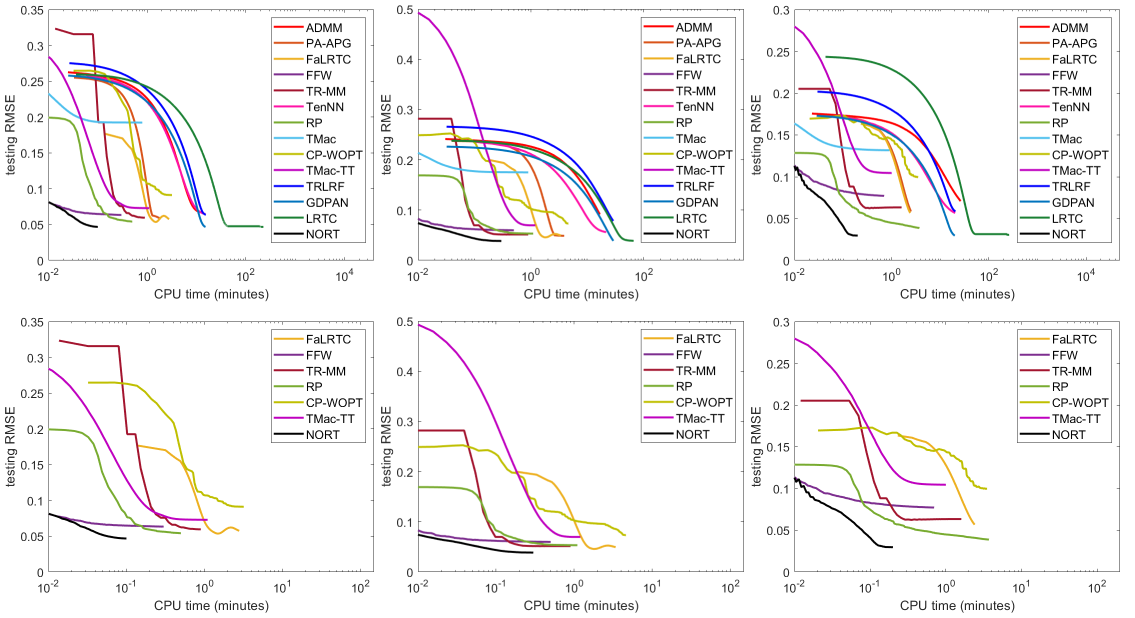
Table 5: Testing RMSEs on color images. For all images 5% of the total pixels, which are corrupted by Gaussian noise $\mathcal{N}(0, 0.01^2)$, are used for training.

dataset		*Rice*	*Tree*	*Windows*
convex	ADMM	0.0680±0.0003	0.0915±0.0005	0.0709±0.0004
	PA-APG	0.0583±0.0016	0.0488±0.0007	0.0585±0.0002
	FaLRTC	0.0576±0.0004	0.0494±0.0011	0.0567±0.0005
	FFW	0.0634±0.0003	0.0599±0.0005	0.0772±0.0004
	TR-MM	0.0596±0.0005	0.0515±0.0011	0.0634±0.0002
	TenNN	0.0647±0.0004	0.0562±0.0004	0.0586±0.0003
factorization	RP	0.0541±0.0011	0.0575±0.0010	0.0388±0.0026
	TMac	0.1923±0.0005	0.1750±0.0006	0.1313±0.0005
	CP-WOPT	0.0912±0.0086	0.0750±0.0060	0.0964±0.0102
	TMac-TT	0.0729±0.0022	0.0665±0.0147	0.1045±0.0107
	TRLRF	0.0640±0.0004	0.0780±0.0048	0.0588±0.0035
nonconvex	GDPAN	**0.0467±0.0002**	0.0394±0.0006	0.0306±0.0007
	LRTC	**0.0468±0.0001**	0.0392±0.0006	0.0304±0.0008
	NORT	**0.0468±0.0001**	**0.0386±0.0009**	**0.0297±0.0007**

testing RMSEs at different noise levels σ's. As can be seen, the testing RMSEs of all methods increase as σ increases. NORT has lower RMSEs at all σ settings. This is because natural images may not be exactly low-rank, and adaptive penalization of the singular values can better preserve the spectrum. A similar observation has also been made for nonconvex regularization on images (Yao et al., 2019b; Lu et al., 2016b). However, when the noise level becomes very high ($\sigma = 0.1$ with pixel values in $[0, 1]$), though NORT is still the best, its testing RMSE is not small.

Table 6: Testing RMSEs on image *Tree* at different noise levels σ. The percentage followed by the marker ↑ indicates the relative increase of testing RMSE compared with NORT.

		$\sigma = 0.001$	$\sigma = 0.01$	$\sigma = 0.1$
(convex)	PA-APG	0.0149 (35.8%↑)	0.0488 (24.6%↑)	0.1749 (18.6% ↑)
(factorization)	RP	0.0139 (26.0%↑)	0.0575 (15.6%↑)	0.1623 (10.1% ↑)
(nonconvex)	NORT	0.0110	0.0386	0.1474



(a) *Rice*. (b) *Tree*. (c) *Windows*.

Figure 10: Testing RMSE versus CPU time on color images. Top: All methods; Bottom: For improved clarity, methods which are too slow or with too poor performance are removed.

5.2.2 REMOTE SENSING DATA

Experiments are performed on three hyper-spectral images (Figure 11): *Cabbage* (1312×432×49), *Scene* (1312×951×49) and *Female* (592×409×148).[15]: The third dimension is for the bands of images.

(a) *Cabbage*. (b) *Scene*. (c) *Female*.

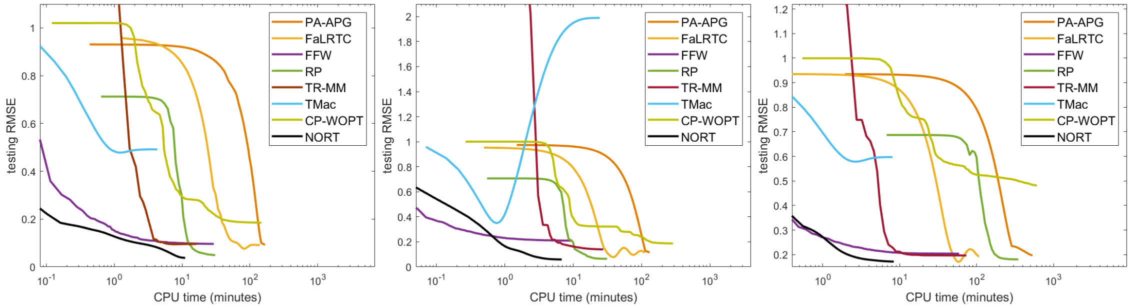
Figure 11: Hyperspectral images used in the experiment.

We use the same setup as in Section 5.2.1, and hyperparameters are tuned on the validation set. ADMM, TenNN, GDPAN, LRTC, TMac-TT and TRLRF are slow and so not compared. Results are shown in Table 7. Again, NORT achieves much lower testing RMSE than convex regularization and factorization approach. Figure 12 shows convergence of the testing RMSE. As can be seen, NORT is the fastest.

[15] *Cabbage* and *Scene* images are from `https://sites.google.com/site/hyperspectralcolorimaging/dataset`, while the *Female* images are downloaded from `http://www.imageval.com/scene-database-4-faces-3-meters/`.

Table 7: Testing RMSEs on remote sensing data.

		Cabbage	*Scene*	*Female*
convex	PA-APG	0.0913±0.0006	0.1965±0.0002	0.1157±0.0003
	FaLRTC	0.0909±0.0002	0.1920±0.0001	0.1133±0.0004
	FFW	0.0962±0.0004	0.2037±0.0002	0.2096±0.0006
	TR-MM	0.0959±0.0001	0.1965±0.0002	0.1397±0.0006
factorization	RP	0.0491±0.0011	0.1804±0.0005	0.0647±0.0003
	TMac	0.4919±0.0059	0.5970±0.0029	1.9897±0.0006
	CP-WOPT	0.1846±0.0514	0.4811±0.0082	0.1868±0.0013
nonconvex	NORT	**0.0376±0.0004**	**0.1714±0.0012**	**0.0592±0.0002**



(a) *Cabbage*.

(b) *Female*.

(c) *Scene*.

Figure 12: Testing RMSE versus CPU time on remote sensing data.

5.2.3 Social Networks

In this experiment, we consider multi-relational link prediction (Guo et al., 2017) as a tensor completion problem. Experiment is performed on the *YouTube* data set[16] (Lei et al., 2009), which contains 15,088 users and five types of user interactions. Thus, it forms a 15088×15088×5 tensor, with a total of 27,257,790 nonzero elements. Besides the full set, we also experiment with a *YouTube* subset obtained by randomly selecting 1,000 users (leading to 12,101 observations). We use 50% of the observations for training, another 25% for validation and the rest for testing. Table 8 shows the testing RMSE, and Figure 13 shows the convergence. As can be seen, NORT achieves smaller RMSE and is also much faster.

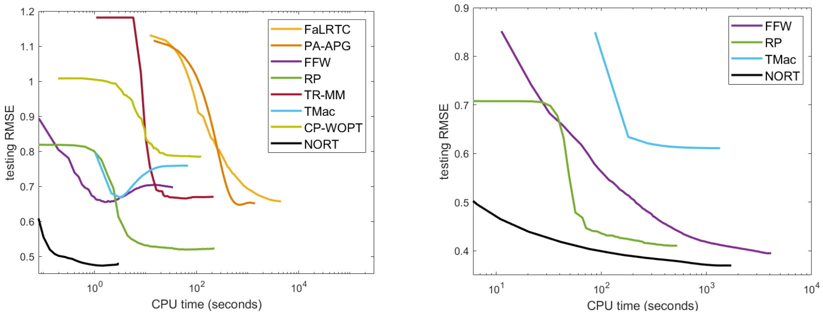
5.3 Link Prediction in Knowledge Graph

Knowledge Graph (KG) (Nickel et al., 2015; Toutanova et al., 2015) is an active research topic in data mining and machine learning. Let $\mathcal{E}$ be the entity set and $\mathcal{R}$ be the relation set. In a KG, nodes are the entities, while edges are relations representing the triplets $\mathcal{S} = \{(h, r, t)\}$, where $h \in \mathcal{E}$ is the *head entity*, $t \in \mathcal{E}$ is the *tail entity*, and $r \in \mathcal{R}$ is the *relation* between h and t.

[16] http://leitang.net/data/youtube-data.tar.gz

Table 8: Testing RMSEs on *YouTube* data sets. FaLRTC, PA-APG, TR-MM and CP-WOPT are slow, and thus not run on the full set.

		subset	full set
convex	FaLRTC	0.657±0.060	—
	PA-APG	0.651±0.047	—
	FFW	0.697±0.054	0.395±0.001
	TR-MM	0.670±0.098	—
factorization	RP	0.522±0.038	0.410±0.001
	TMac	0.795±0.033	0.611±0.007
	CP-WOPT	0.785±0.040	—
nonconvex	NORT	**0.482±0.030**	**0.370±0.001**



(a) Subset.

(b) Full set.

Figure 13: Testing RMSE versus CPU time on *Youtube*.

KGs have many downstream applications, such as link prediction and triplet classification. It is common to store KGs as tensors, and solve the KG learning tasks with tensor methods (Lacroix et al., 2018; Balazevic et al., 2019). Take link prediction as an example. The KG can be seen as a 3-order incomplete tensor $\mathcal{O} = \{\pm 1\} \in \mathbb{R}^{I^1 \times I^2 \times I^3}$, where $I^1 = I^2 = |\mathcal{E}|$ and $I^3 = |\mathcal{R}|$. $\mathcal{O}_{i_1 i_2 i_3} = 1$ when entities i_1 and i_2 have the relation i_3, and -1 otherwise. Let $\mathbf{\Omega}$ be a mask tensor denoting the observed values in $\mathcal{O}$, i.e., $\mathbf{\Omega}_{i_1 i_2 i_3} = 1$ if $\mathcal{O}_{i_1 i_2 i_3}$ is observed and 0 otherwise. The task is to predict elements in $\mathcal{O}$ which are not observed. Since $\mathcal{O}$ is binary, it is common to use the log loss as $\ell(\cdot, \cdot)$ in (14). The objective then becomes:

$$\min_{\mathcal{X}} \sum\nolimits_{(i_1 i_2 i_3) \in \mathbf{\Omega}} \log(1 + \exp(-\mathcal{X}_{i_1 i_2 i_3} \mathcal{O}_{i_1 i_2 i_3})) + \sum\nolimits_{i=1}^{D} \lambda_i \phi(\mathcal{X}_{\langle i \rangle}). \tag{47}$$

In step 9 of Algorithm 2, it is easy to see that

$$[\xi(\mathcal{X}_t)]_{i_1 i_2 i_3} = \begin{cases} \frac{-\mathcal{O}_{i_1 i_2 i_3} \cdot \exp(-\mathcal{X}_{i_1 i_2 i_3} \mathcal{O}_{i_1 i_2 i_3})}{1 + \exp(-\mathcal{X}_{i_1 i_2 i_3} \mathcal{O}_{i_1 i_2 i_3})} & (i_1 i_2 i_3) \in \mathbf{\Omega} \\ 0 & (i_1 i_2 i_3) \notin \mathbf{\Omega} \end{cases}.$$

Experiments are performed on two benchmark data sets: *WN18RR*[17] (Dettmers et al., 2018) and *FB15k-237*[18] (Toutanova et al., 2015), which are subsets of *WN18* and *FB15k* (Bordes et al., 2013), respectively. *WN18* is a subset of WordNet (Miller, 1995), and *FB15k* is a subset of the Freebase database (Bollacker et al., 2008). To avoid test leakage, *WN18RR* and *FB15k-237* do not contain near-duplicate and inverse-duplicate relations. Hence, link prediction on *WN18RR* and *FB15k-237* is harder but more recommended than that on *WN18* and *FB15k* (Dettmers et al., 2018). To form the entity set $\mathcal{E}$, we keep the top 500 (head and tail) entities that appear most frequently in the relations (r's). Relations that do not link to any of these 500 entities are removed, and those remained form the relation set $\mathcal{R}$. Following the public splits on entities in $\mathcal{E}$ and relations in $\mathcal{R}$ (Han et al., 2018), we split the observed triplets in $\mathcal{S}$ into a training set $\mathcal{S}_{\text{train}}$, validation set $\mathcal{S}_{\text{val}}$ and testing set $\mathcal{S}_{\text{test}}$. For each observed triplet $(h, r, t) \in \mathcal{S}_{\text{train}}$, we sample a negative triplet from $\hat{\mathcal{S}}_{(h,r,t)} = \{(\hat{h}, r, t) \notin \mathcal{S} | \hat{h} \in \mathcal{E}\} \cap \{(h, r, \hat{t}) \notin \mathcal{S} | \hat{t} \in \mathcal{E}\}$. We avoid duplicate negative triplets during sampling. We then represent the KG's by tensors $\mathcal{O}$'s of size $500 \times 500 \times 8$ for *WN18RR*, and $500 \times 500 \times 39$ for *FB15k-237* with corresponding mask tensors $\mathbf{\Omega}$'s.

Following (Bordes et al., 2013; Dettmers et al., 2018), performance is evaluated on the testing triplets in $\bar{\mathbf{\Omega}}$ by the following metrics: (i) mean reciprocal ranking: $\text{MRR} = 1/\|\bar{\mathbf{\Omega}}\|_0 \sum_{(i_1 i_2 i_3) \in \bar{\mathbf{\Omega}}} 1/\text{rank}_{i_3}$, where rank_{i_3} is the ranking of score $\mathcal{X}_{i_1 i_2 i_3}$ among $\{\mathcal{X}_{i_1 i_2 j}\}$ with $j = 1, \dots, |\mathcal{R}|$ in descending order; (ii) $\text{Hits@1} = 1/\|\bar{\mathbf{\Omega}}\|_0 \sum_{(i_1 i_2 i_3) \in \bar{\mathbf{\Omega}}} \mathbb{I}(\text{rank}_{i_3} \leq 1)$, where $\mathbb{I}(c)$ is the indicator function which returns 1 if the constraint c is satisfied and 0 otherwise; and (iii) $\text{Hits@3} = 1/\|\bar{\mathbf{\Omega}}\|_0 \sum_{(i_1 i_2 i_3) \in \bar{\mathbf{\Omega}}} \mathbb{I}(\text{rank}_{i_3} \leq 3)$. For these three metrics, the higher the better.

The aforementioned algorithms are designed for the square loss, but not for the log loss in (47). We adapt the gradient-based algorithms including PA-APG, ADMM and CP-WOPT, as we only need to change the gradient calculation for (47). As a further baseline, we implement the classic Tucker decomposition (Tucker, 1966; Kolda and Bader, 2009) to optimize (47). While RP (Kasai and Mishra, 2016) is the state-of-the-art Tucker-type algorithm, it uses Riemannian preconditioning and cannot be easily modified to handle nonsmooth loss.

Results on *WN18RR* and *FB15k-237* are shown in Tables 9 and 10, respectively. As can be seen, NORT again obtains the best ranking results. Figure 14 shows convergence of MRR with CPU time, and NORT is about two orders of magnitude faster than the other methods.

Table 9: Testing performance on the *WN18RR* data set.

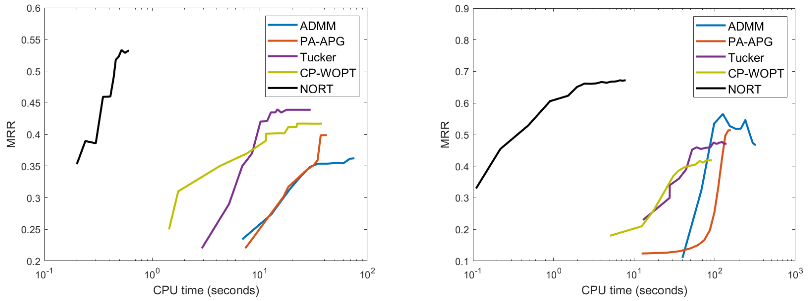
		MRR	Hits@1	Hits@3
convex	ADMM	0.362±0.029	0.156±0.024	0.422±0.038
	PA-APG	0.399±0.017	0.203±0.023	0.500±0.038
factorization	Tucker	0.439±0.013	0.309±0.016	0.438±0.026
	CP-WOPT	0.417±0.018	0.266±0.027	0.453±0.019
nonconvex	NORT	**0.523±0.022**	**0.375±0.033**	**0.578±0.024**

[17] `https://github.com/TimDettmers/ConvE`

[18] `https://www.microsoft.com/en-us/download/details.aspx?id=52312`

Table 10: Testing performance on the *FB15k-237* data set.

		MRR	Hits@1	Hits@3
convex	ADMM	0.466±0.006	0.411±0.006	0.452±0.011
	PA-APG	0.514±0.013	0.463±0.015	0.590±0.016
factorization	Tucker	0.471±0.018	0.355±0.017	0.465±0.015
	CP-WOPT	0.420±0.021	0.373±0.015	0.488±0.014
nonconvex	NORT	**0.677±0.007**	**0.609±0.007**	**0.698±0.011**



(a) *WN18RR*.

(b) *FB15k-237*.

Figure 14: Testing MRR versus CPU time on the *WN18RR* and *FB15k-237* data sets.

5.4 Robust Tensor Completion

In this section, we apply the proposed method on robust video tensor completion. Three videos (*Eagle*[19], *Friends*[20] and *Logo*[21]) from (Indyk et al., 2019) are used. Example frames are shown in Figure 15. For each video, 200 consecutive 360 × 640 frames are downloaded from Youtube, and the pixel values are normalized to $[0, 1]$. Each video can then be represented as a fourth-order tensor $\bar{\mathcal{O}}$ with size 360 × 640 × 3 × 200. Each element of $\bar{\mathcal{O}}$ is normalized to $[0, 1]$. This clean tensor $\bar{\mathcal{O}}$ is corrupted by a noise tensor $\mathcal{N}$ to form $\mathcal{O}$. $\mathcal{N}$ is a sparse random tensor with approximately 1% nonzero elements. Each entry is first drawn uniformly from the interval $[0, 1]$, and then multiplied by 5 times the maximum value of $\bar{\mathcal{O}}$. Hyperparameters are chosen based on performance on the first 100 noisy frames. Denoising performance is measured by the RMSE between the clean tensor $\bar{\mathcal{O}}$ and reconstructed tensor $\mathcal{X}$ on the last 100 frames.

(a) *Eagle*.

(b) *Friends*.

(c) *Logo*.

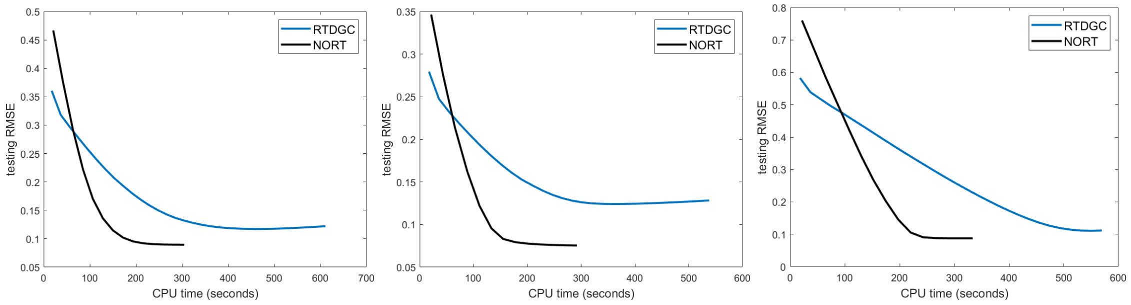
Figure 15: Example image frames in the videos.

[19] http://youtu.be/ufnf_q_3Ofg
[20] http://youtu.be/xmLZsEfXEgE
[21] http://youtu.be/L5HQoFIaT4I

For the robust tensor completion, we take RTDGC (Gu et al., 2014) as the baseline, which adopts the ℓ_1 loss and overlapped nuclear norm in (40) (i.e., $\kappa_\ell(x) = x$ and ϕ is the nuclear norm). As this is non-smooth and non-differentiable, RTDGC uses ADMM (Boyd et al., 2011) for the optimization, which handles the robust loss and low-rank regularizer separately. As discussed in Section 4.1, we use the smoothing NORT (Algorithm 3, with $\delta_0 = 0.9$) to optimize (42), the smoothed version of (40). Table 11 shows the RMSE results. As can be seen, NORT obtains better denoising performance than RTDGC. This again validates the efficacy of nonconvex low-rank learning. Figure 16 shows convergence of the testing RMSE. As shown, NORT leads to a lower RMSE and converges much faster as folding/unfolding are avoided.

Table 11: Testing RMSEs on the videos.

		Eagle	*Friends*	*Logo*
convex	RTDGC	0.122±0.007	0.128±0.005	0.112±0.008
nonconvex	NORT	**0.090±0.003**	**0.075±0.002**	**0.088±0.004**



(a) *Eagle*. (b) *Friends*. (c) *Logo*.

Figure 16: Testing RMSE versus CPU time on the videos.

5.5 Spatial-temporal Data

In this experiment, we predict climate observations for locations that do not have any records. This is formulated as a regularized tensor completion problem in (43). We use the square loss with a graph Laplacian regularizer constructed as in (43).

We use the *CCDS* and *USHCN* data sets from (Bahadori et al., 2014). *CCDS*[22] contains monthly observations of 17 variables (such as carbon dioxide and temperature) in 125 stations from January 1990 to December 2001. *USHCN*[23] contains monthly observations of 4 variables (minimum, maximum, average temperature and total precipitation) in 1218 stations from from January 1919 to November 2019. As discussed in Section 4.2, these records are collectively represented by a 3-order tensor $\mathcal{O} \in \mathbb{R}^{I^1 \times I^2 \times I^3}$, where I^1 is the number of locations, I^2 is the number of recorded time stamps, and I^3 is the number of variables corresponding to climate observations. Consequently, *CCDS* is represented as a 125 × 156 × 17 tensor and *USHCN* is represented as a 1218 × 1211 × 4

[22] `https://viterbi-web.usc.edu/~liu32/data/NA-1990-2002-Monthly.csv`

[23] `http://www.ncdc.noaa.gov/oa/climate/research/ushcn`

tensor. The affinity matrix is denoted $\boldsymbol{A}$, with $\boldsymbol{A}_{ij}$ being the similarity $s(i,j) = \exp(-2b_{ij})$ between locations i and j (b_{ij} is the Haversine distance between i and j). Following (Bahadori et al., 2014), we normalize the data to zero mean and unit variance, then randomly sample 10% of the locations for training, another 10% for validation, and the rest for testing.

Algorithms FaLRTC, FFW, TR-MM, RP and TMac cannot be directly used for this graph Laplacian regularized tensor completion problem, while PA-APG, ADMM, Tucker and CP-WOPT can be adapted by modifying the gradient calculation. Hence we adapt and implement PA-APG, ADMM, Tucker and CP-WOPT as baselines in this section. In addition, we compare with a greedy algorithm (denoted "Greedy")[24] from (Bahadori et al., 2014), which successively adds a rank-1 matrix to approximate the mode-n unfolding with the rank constraint. For the factorization-based algorithms Tucker and CP-WOPT, the graph Laplacian regularizer h takes the corresponding factor matrix rather than $\mathcal{X}_{\langle 1 \rangle}$ as the input. Specifically, recall that Tucker factorizes $\mathcal{X}$ into $[\mathcal{G}; \boldsymbol{B}^1, \boldsymbol{B}^2, \boldsymbol{B}^3]$, where $\mathcal{G} \in \mathbb{R}^{k^1 \times k^2 \times k^3}$, $\boldsymbol{B}^i \in \mathbb{R}^{I^i \times k^i}$, $i = 1, 2, 3$, and k^i's are hyperparameters. When $k^1 = k^2 = k^3$ and $\mathcal{G}$ is superdiagonal, this reduces to the CP-WOPT decomposition. The graph Laplacian regularizer is then constructed as $h(\boldsymbol{B}^1)$ to leverage location proximity. As an additional baseline, we also experiment with a NORT variant that does not use the Laplacian regularizer (denoted "NORT-no-Lap").

Table 12: Testing RMSEs on *CCDS* and *USHCN* data sets.

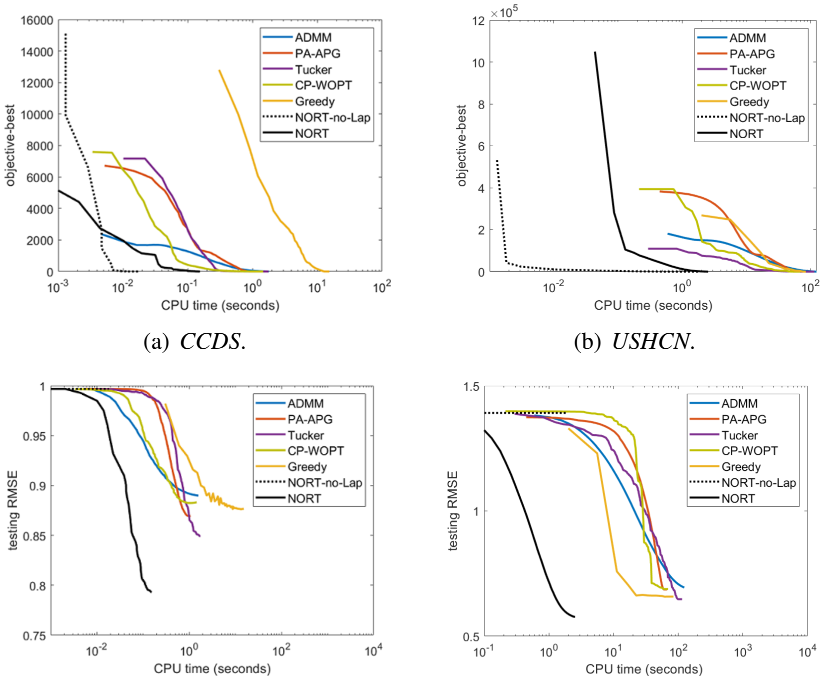
		CCDS	*USHCN*
convex	ADMM	0.890±0.016	0.691±0.005
	PA-APG	0.866±0.014	0.680±0.009
factorization	Tucker	0.856±0.026	0.647±0.006
	CP-WOPT	0.887±0.018	0.688±0.009
rank constraint	Greedy	0.871±0.008	0.658±0.012
nonconvex	NORT-no-Lap	0.997±0.001	1.391±0.001
	NORT	**0.793±0.002**	**0.583±0.012**

Table 12 shows the RMSE results. Again, NORT obtains the lowest testing RMSEs. Moreover, when the Laplacian regularizer is not used, the testing RMSE is much higher, demonstrating that the missing slices cannot be reliably completed. Figure 17 shows the convergence. As can be seen, NORT is orders of magnitude faster than the other algorithms. The gaps on the performance and speed between NORT and the other baselines are more obvious on the larger *USHCN* data set. Further, note from Figures 17(a) and 17(b) that though NORT-no-Lap has converged, it cannot decrease the testing RMSE during learning (Figures 17(c) and 17(d)). This validates the efficacy of the graph Laplacian regularizer.

6. Conclusion

In this paper, we propose a low-rank tensor completion model with nonconvex regularization. An efficient nonconvex proximal average algorithm is developed, which maintains the "sparse plus

[24] This method is denoted "ORTHOGONAL" in (Bahadori et al., 2014) and obtains the best results there.



(a) *CCDS*.

(b) *USHCN*.

(c) *CCDS*.

(d) *USHCN*.

Figure 17: Convergence of the training objective (below) and testing RMSE (top) versus CPU time on the spatial-temporal data.

low-rank" structure throughout the iterations and incorporates adaptive momentum. Convergence to critical points is guaranteed, and the obtained critical points can have small statistical errors. The algorithm is also extended for nonsmooth losses and additional regularization, demonstrating broad applicability of the proposed algorithm. Experiments on a variety of synthetic and real data sets are performed. Results show that the proposed algorithm is more efficient and more accurate than existing state-of-the-art.

In the future, we will extend the proposed algorithm to simultaneous completion of multiple tensors, e.g., collaborative tensor completion (Zheng et al., 2013) and coupled tensor completion (Wimalawarne et al., 2018). Besides, it is also interesting to study how the proposed algorithm can be efficiently parallelized on GPUs and distributed computing environments (Phipps and Kolda, 2019).

Acknowledgement

BH was supported by the RGC Early Career Scheme No.22200720 and NSFC Young Scientists Fund No. 62006202.

Appendix

Appendix A. Comparison with Incoherence Condition

The matrix incoherence condition (Candès and Recht, 2009; Candès et al., 2011; Negahban and Wainwright, 2012) is in form of the singular value decomposition $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \in \mathbb{R}^{m \times n}$, where $\boldsymbol{U} \in \mathbb{R}^{m \times r}$ (resp. $\boldsymbol{V} \in \mathbb{R}^{n \times r}$) contains the left (resp. right) singular vectors and $\boldsymbol{\Sigma} \in \mathbb{R}^{r \times r}$ is the diagonal matrix containing singular values. The purpose of this condition is to enforce that the left and right singular vectors should not be aligned with the standard basis (i.e., vector $\boldsymbol{e}_i$'s with the ith dimension being 1 and others being 0). Typically, this condition is stated as

$$\max_{j=1,\dots,m} \left| [\boldsymbol{U}\boldsymbol{U}^\top]_{jj} \right| \le \mu_0 \frac{r}{m}, \quad \text{and} \quad \max_{j=1,\dots,n} \left| [\boldsymbol{V}\boldsymbol{V}^\top]_{jj} \right| \le \mu_0 \frac{r}{n}, \tag{48}$$

for some constant $\mu_0 > 0$. Note that (48) does not depend on the singular values of $\boldsymbol{X}$. However, this condition can be restrictive in realistic settings, where the underlying matrix is contaminated by noise. In this case, the observed matrix can have small singular values. Therefore, we need to impose conditions related to the singular values, and (32) shows such a dependency. An example matrix satisfying the matrix RSC condition but not the incoherence condition is in Section 3.4.2 of (Negahban and Wainwright, 2012). As a result, the RSC condition, which involves singular values, is less restrictive than the incoherence condition, and can better describe "spikiness".

Appendix B. Proofs

B.1 Proposition 4

Proof For simplicity, we consider the case where $\boldsymbol{U} \in \mathbb{R}^{I^j \times k}$ (resp. $\boldsymbol{V} \in \mathbb{R}^{(\frac{I^\pi}{I^j}) \times k}$) has only one single column $\boldsymbol{u} \in \mathbb{R}^{I^j}$ (resp. $\boldsymbol{v} \in \mathbb{R}^{\frac{I^\pi}{I^j}}$). We need to fold $\boldsymbol{u}\boldsymbol{v}^\top$ along with the jth mode and then unfold it along its ith mode. Let us consider the structure of $\mathcal{X} = (\boldsymbol{u}\boldsymbol{v}^\top)^{\langle j \rangle}$, we can express it as

$$\mathcal{X}_{\langle j \rangle} = \left[\mathbf{u}\boldsymbol{v}_1^\top, ..., \mathbf{u}\boldsymbol{v}^\top_{\frac{I^\pi}{(I^i I^j)}} \right] \in \mathbb{R}^{I^j \times \frac{I^\pi}{I^j}},$$

where $\boldsymbol{v} = [\boldsymbol{v}_1; \dots; \boldsymbol{v}_{\frac{I^\pi}{(I^i I^j)}}]$ with each $\boldsymbol{v}_p \in \mathbb{R}^{I^i}$. When unfolding $\mathcal{X}$ with the ith mode, the unfolding matrix is

$$\left[\boldsymbol{v}_1 \boldsymbol{u}^\top, \dots, \boldsymbol{v}_{\frac{I^\pi}{(I^i I^j)}} \boldsymbol{u}^\top \right] \in \mathbb{R}^{I^i \times \frac{I^\pi}{I^i}}. \tag{49}$$

Thus,

$$\begin{aligned} \boldsymbol{a}^\top \left[\boldsymbol{v}_1 \boldsymbol{u}^\top, \dots, \boldsymbol{v}_{I^3} \boldsymbol{u}^\top \right] &= \left[(\boldsymbol{a}^\top \boldsymbol{v}_1) \boldsymbol{u}^\top, ..., (\boldsymbol{a}^\top \boldsymbol{v}_{I^3}) \boldsymbol{u}^\top \right], \\ &= \left(\boldsymbol{a}^\top \text{mat}\left(\boldsymbol{v}_p; I^i, \bar{I}^{ij} \right) \right) \otimes \boldsymbol{u}^\top. \end{aligned} \tag{50}$$

Similarly, let $\boldsymbol{b} = \big[\boldsymbol{b}_1; \dots; \boldsymbol{b}_{\frac{I^\pi}{I^i I^j}}\big]$, where each $\boldsymbol{b}_p \in \mathbb{R}^{I^j}$. From (49), we have

$$\begin{aligned}
\big[\boldsymbol{v}_1\boldsymbol{u}^\top, \dots, \boldsymbol{v}_{\frac{I^\pi}{(I^iI^j)}}\boldsymbol{u}^\top\big]\boldsymbol{b} &= \textstyle\sum_{j=1}^{\frac{I^\pi}{I^iI^j}} \boldsymbol{v}_i(\boldsymbol{u}^\top\boldsymbol{b}_i), \\
&= \big[\boldsymbol{v}_1; \dots; \boldsymbol{v}_{\frac{I^\pi}{I^iI^j}}\big] \begin{bmatrix} \boldsymbol{u}^\top\boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{u}^\top\boldsymbol{b}_{\frac{I^\pi}{I^iI^j}} \end{bmatrix}, \\
&= \big[\boldsymbol{v}_1; \dots; \boldsymbol{v}_{\frac{I^\pi}{I^iI^j}}\big]\big[\boldsymbol{b}_1; \dots; \boldsymbol{b}_{\frac{I^\pi}{I^iI^j}}\big]^\top \boldsymbol{u}, \\
&= \text{mat}\left(\boldsymbol{v}; I^i, \bar{I}^{ij}\right) \text{mat}\left(\boldsymbol{b}; \bar{I}^{ij}, I^j\right) \boldsymbol{u}. \qquad (51)
\end{aligned}$$

When $\boldsymbol{U}$ (resp. $\boldsymbol{V}$) has k columns, combining with the fact that $\boldsymbol{U}\boldsymbol{V}^\top = \sum_{p=1}^k \boldsymbol{u}_p\boldsymbol{v}_p^\top$ with (50) and (51), we obtain (26) and (27). ■

B.2 Proposition 6

Proof Define $\bar{\lambda}_d = \lambda_d/\tau$, then

$$\begin{aligned}
&\textstyle\sum_{d=1}^D \min_{\boldsymbol{X}_d} \frac{1}{2}\left\|\boldsymbol{X}_d - \mathcal{Z}_{\langle d\rangle}\right\|_F^2 + \bar{\lambda}_d\,\phi(\boldsymbol{X}_d), \\
&= \min_{\{\boldsymbol{X}_d\}} \frac{D}{2}\|\mathcal{Z}\|_F^2 - \big\langle \mathcal{Z}, \textstyle\sum_{d=1}^D \boldsymbol{X}_d^{\langle d\rangle}\big\rangle + \frac{D}{2}\textstyle\sum_{d=1}^D \|\boldsymbol{X}_d\|_F^2 + \textstyle\sum_{d=1}^D \bar{\lambda}_d\phi(\boldsymbol{X}_d), \\
&= \min_{\{\boldsymbol{X}_d\}} \frac{D}{2}\Big\|\mathcal{Z} - \textstyle\sum_{d=1}^D \boldsymbol{X}_d^{\langle d\rangle}\Big\|_F^2 - \frac{D}{2}\Big\|\textstyle\sum_{d=1}^D \boldsymbol{X}_d^{\langle d\rangle}\Big\|_F^2 + \textstyle\sum_{d=1}^D \big[\frac{1}{2}\|\boldsymbol{X}_d\|_F^2 + \bar{\lambda}_d\phi(\boldsymbol{X}_d)\big]. \qquad (52)
\end{aligned}$$

Next, we introduce an extra parameter as $\mathcal{X} = \sum_{d=1}^D \boldsymbol{X}_d^{\langle d\rangle}$, and express (52) as

$$\begin{aligned}
&\min_{\{\boldsymbol{X}_d\}:\mathcal{X}=\sum_{d=1}^D \boldsymbol{X}_d^{\langle d\rangle}} \frac{D}{2}\|\mathcal{Z} - \mathcal{X}\|_F^2 - \frac{D}{2}\Big\|\textstyle\sum_{d=1}^D \boldsymbol{X}_d^{\langle d\rangle}\Big\|_F^2 + \textstyle\sum_{d=1}^D \left[\frac{1}{2}\|\boldsymbol{X}_d\|_F^2 + \bar{\lambda}_d\phi(\boldsymbol{X}_d)\right], \\
&= \min_{\mathcal{X}} \left\{ \frac{D}{2}\|\mathcal{Z} - \mathcal{X}\|_F^2 + \min_{\{\boldsymbol{X}_d\}:\sum_{d=1}^D \boldsymbol{X}_d^{\langle d\rangle} = \mathcal{X}} \textstyle\sum_{d=1}^D \left[\frac{1}{2}\|\boldsymbol{X}_d\|_F^2 + \bar{\lambda}_d\phi(\boldsymbol{X}_d)\right] - \frac{D}{2}\|\mathcal{X}\|_F^2 \right\}. \qquad (53)
\end{aligned}$$

We transform the above equation as

$$\min_{\mathcal{X}} \frac{1}{2}\|\mathcal{Z} - \mathcal{X}\|_F^2 + \frac{1}{\tau}\bar{g}_\tau(\mathcal{X}) = \text{prox}_{\frac{\bar{g}_\tau}{\tau}}(\mathcal{X}),$$

where $\bar{g}_\tau(\mathcal{X})$ is defined as

$$\begin{aligned}
\bar{g}_\tau(\mathcal{X}) = \tau &\left[\min_{\{\boldsymbol{X}_d\}} \textstyle\sum_{d=1}^D \left(\frac{1}{2}\|\boldsymbol{X}_d\|_F^2 + \bar{\lambda}_d\phi(\boldsymbol{X}_d)\right) - \frac{D}{2}\|\mathcal{X}\|_F^2\right], \qquad (54) \\
&\text{s.t. } \textstyle\sum_{d=1}^D \boldsymbol{X}_d^{\langle d\rangle} = \mathcal{X}.
\end{aligned}$$

Thus, there exists $\bar{g}_\tau$ such that $\text{prox}_{\frac{\bar{g}_\tau}{\tau}}(\mathcal{Z}) = \sum_{i=1}^D \left[\text{prox}_{\bar{\lambda}_d\phi}([\mathcal{Z}]_{\langle i\rangle})\right]^{\langle i\rangle}$. ■

B.3 Proposition 7

Let $g(\mathcal{X}) = \sum_{d=1}^{D} \lambda_i \phi(\mathcal{X}_{\langle d \rangle})$. Before proving Proposition 7, we first extend Proposition 2 in (Zhong and Kwok, 2014) in the following auxiliary Lemma.

B.3.1 AUXILIARY LEMMA

Lemma 21 $0 \le g(\mathcal{X}) - \bar{g}_\tau(\mathcal{X}) \le \frac{\kappa_0^2}{2\tau} \sum_{d=1}^{D} \lambda_d^2$.

Proof From the definition of $\bar{g}_\tau$ in (54), if $\mathcal{X} = \boldsymbol{X}_1^{\langle 1 \rangle} = ... = \boldsymbol{X}_D^{\langle D \rangle}$, we have

$$\begin{aligned} \bar{g}_\tau(\mathcal{X}) &\le \tau\big(\textstyle\sum_{d=1}^{D} \big(\frac{1}{2} \left\| \boldsymbol{X}_d^{\langle d \rangle} \right\|_F^2 + \bar{\lambda}_d \phi(\boldsymbol{X}_d) \big) - \frac{D}{2} \left\| \mathcal{X} \right\|_F^2 \big), \\ &= \textstyle\sum_{d=1}^{D} \lambda_d \phi(\boldsymbol{X}_d) = \sum_{d=1}^{D} \lambda_d \phi(\mathcal{X}_{\langle d \rangle}) = g(\mathcal{X}). \end{aligned}$$

Thus, $g(\mathcal{X}) - \bar{g}_\tau(\mathcal{X}) \ge 0$. Next, we prove the "$\le$" part in the Lemma. Note that

$$\begin{aligned} &\sup_{\boldsymbol{X}_d} \lambda_d \phi(\boldsymbol{X}_d) - \tau \min_{\boldsymbol{Y}} \big(\frac{1}{2} \left\| \boldsymbol{Y} - \boldsymbol{X}_d \right\|_F^2 + \bar{\lambda}_d \phi(\boldsymbol{Y}) \big), \\ &= \sup_{\boldsymbol{X}_d, \boldsymbol{Y}} \lambda_d \phi(\boldsymbol{X}_d) - \frac{\tau}{2} \left\| \boldsymbol{Y} - \boldsymbol{X}_d \right\|_F^2 - \lambda_d \phi(\boldsymbol{Y}). \end{aligned} \tag{55}$$

Since ϕ is κ_0-Lipschitz continuous, let $\alpha = \left\| \boldsymbol{Y} - \boldsymbol{X}_d \right\|_F$, we have

$$\begin{aligned} (55) &= \sup_{\boldsymbol{X}_d, \boldsymbol{Y}} \lambda_d \left[\phi(\boldsymbol{X}_d) - \phi(\boldsymbol{X}) \right] - \frac{\tau}{2} \left\| \boldsymbol{Y} - \boldsymbol{X}_d \right\|_F^2, \\ &\le \sup_{\boldsymbol{X}_d, \boldsymbol{Y}} \lambda_d \kappa_0 \left\| \boldsymbol{Y} - \boldsymbol{X}_d \right\|_F - \frac{\tau}{2} \left\| \boldsymbol{Y} - \boldsymbol{X}_d \right\|_F^2, \\ &= \sup_\alpha \big[\lambda_d \kappa_0 \alpha - \frac{\tau}{2} \alpha^2 \big] = \sup_\alpha -\frac{1}{2} \big[\alpha - \frac{\lambda_d \kappa_0}{\tau} \big]^2 + \frac{\lambda_d^2 \kappa_0^2}{2} \le \frac{\lambda_d^2 \kappa_0^2}{2\tau}. \end{aligned} \tag{56}$$

Next, we have

$$g(\mathcal{X}) - \bar{g}_\tau(\mathcal{X}) \le g(\mathcal{X}) - \tau\big(\min_{\mathcal{Y}} \frac{1}{2} \left\| \mathcal{X} - \mathcal{Y} \right\|_F^2 + \frac{1}{\tau} \bar{g}_\tau(\mathcal{Y}) \big), \tag{57}$$

$$= \sum_{d=1}^{D} \lambda_d \phi(\mathcal{X}_{\langle d \rangle}) - \tau \sum_{d=1}^{D} \big(\min_{\{\boldsymbol{Y}_d\}} \frac{1}{2} \left\| \mathcal{X}_{\langle d \rangle} - \boldsymbol{Y}_d \right\|_F^2 + \frac{\lambda_d}{\tau} \phi(\boldsymbol{Y}_d) \big), \tag{58}$$

$$\le \sup_{\mathcal{X}} \sum_{d=1}^{D} \lambda_d \phi(\mathcal{X}_{\langle d \rangle}) - \tau \sum_{d=1}^{D} \big(\min_{\{\boldsymbol{Y}_d\}} \frac{1}{2} \left\| \mathcal{X}_{\langle d \rangle} - \boldsymbol{Y}_d \right\|_F^2 + \frac{\lambda_d}{\tau} \phi(\boldsymbol{Y}_d) \big),$$

$$\le \sum_{d=1}^{D} \frac{\lambda_d^2 \kappa_0^2}{2\tau}. \tag{59}$$

Note that (57) comes from the fact that

$$\min_{\mathcal{Y}} \frac{1}{2} \left\| \mathcal{X} - \mathcal{Y} \right\|_F^2 + \frac{1}{\tau} \bar{g}_\tau(\mathcal{Y}) \le \frac{1}{2} \left\| \mathcal{X} - \mathcal{X} \right\|_F^2 + \frac{1}{\tau} \bar{g}_\tau(\mathcal{X}) = \frac{1}{\tau} \bar{g}_\tau(\mathcal{X}),$$

then (58) is from the definition of $\bar{g}_\tau$ in Proposition 6, and (59) is from (56). ■

B.3.2 PROOF OF PROPOSITION 7

Proof From Lemma 21, we have

$$\min_{\mathcal{X}} F(\mathcal{X}) - \min_{\mathcal{X}} F_\tau(\mathcal{X}) \geq \min_{\mathcal{X}} F(\mathcal{X}) - F_\tau(\mathcal{X}) = g(\mathcal{X}) - \bar{g}_\tau(\mathcal{X}) \geq 0.$$

Let $\mathcal{X}_1 = \arg\min_{\mathcal{X}} F(\mathcal{X})$ and $\mathcal{X}_\tau = \arg\min_{\mathcal{X}} F_\tau(\mathcal{X})$. Then, we have

$$\min_{\mathcal{X}} F(\mathcal{X}) - \min_{\mathcal{X}} F_\tau(\mathcal{X}) = F(\mathcal{X}_1) - F_\tau(\mathcal{X}_\tau) \leq F(\mathcal{X}_\tau) - F_\tau(\mathcal{X}_\tau) = g(\mathcal{X}_\tau) - \bar{g}_\tau(\mathcal{X}_\tau) \leq \frac{\kappa_0^2}{2\tau} \sum_{d=1}^{D} \lambda_d^2.$$

Thus, $0 \leq \min F - \min F_\tau \leq \frac{\kappa_0^2}{2\tau} \sum_{d=1}^{D} \lambda_d^2$. ■

B.4 Proposition 8

Proof The proof of this proposition can also be found in (Zhong and Kwok, 2014), we add one here for the completeness. Recall that

$$\operatorname{prox}_{\frac{\bar{g}_\tau}{\tau}}(\tilde{\mathcal{X}} - \nabla f(\tilde{\mathcal{X}})/\tau) = \arg\min_{\mathcal{X}} \frac{1}{2} \left\| \mathcal{X} - \left(\tilde{\mathcal{X}} - \frac{1}{\tau} \nabla f(\tilde{\mathcal{X}}) \right) \right\|_F^2 + \frac{1}{\tau} \bar{g}_\tau(\mathcal{X}).$$

Let $\mathcal{Z} = \operatorname{prox}_{\frac{\bar{g}_\tau}{\tau}}(\tilde{\mathcal{X}} - \nabla f(\tilde{\mathcal{X}})/\tau)$. Thus,

$$0 \in \mathcal{Z} - \left(\tilde{\mathcal{X}} - \frac{1}{\tau} \nabla f(\tilde{\mathcal{X}}) \right) + \frac{1}{\tau} \partial \bar{g}_\tau(\mathcal{X}).$$

When $\mathcal{Z} = \tilde{\mathcal{X}}$, we have $0 \in \nabla f(\tilde{\mathcal{X}}) + \partial \bar{g}_\tau(\mathcal{X})$. Thus, $\tilde{\mathcal{X}}$ is a critical point of F_τ. ■

B.5 Theorem 9

First, we introduce the following Lemmas, which are basic properties for the proximal step.

Lemma 22 *(Parikh and Boyd, 2013) Let $\tau > \rho + D\kappa_0$ and $\eta = \tau - \rho + D\kappa_0$. Then, $F_\tau(\operatorname{prox}_{\frac{\bar{g}_\tau}{\tau}}(\mathcal{X})) \leq F_\tau(\mathcal{X}) - \frac{\eta}{2} \|\mathcal{X} - \operatorname{prox}_{\frac{\bar{g}_\tau}{\tau}}(\mathcal{X})\|_F^2$.*

Lemma 23 *(Parikh and Boyd, 2013) If $\mathcal{X} = \operatorname{prox}_{\frac{\bar{g}_\tau}{\tau}}(\mathcal{X} - \frac{1}{\tau} \nabla f(\mathcal{X}))$, then $\mathcal{X}$ is a critical point of F_τ.*

Lemma 24 *(Hare and Sagastizábal, 2009) The proximal map $\operatorname{prox}_{\frac{\bar{g}_\tau}{\tau}}(\mathcal{X})$ is continuous.*

Proof (*of Theorem 9*) Recall that $\operatorname{prox}_{\frac{\bar{g}_\tau}{\tau}}(\mathcal{X}) = \sum_{i=1}^{D} \operatorname{prox}_{\frac{\lambda_i \phi}{\tau}}(\mathcal{X}_{\langle i \rangle})$. From Lemma 22,

- If step 7 is performed, we have

$$F_\tau(\mathcal{X}_{t+1}) \leq F_\tau(\mathcal{V}_t) - \frac{\eta}{2} \|\mathcal{X}_{t+1} - \mathcal{V}_t\|_F^2 \leq F_\tau(\mathcal{X}_t) - \frac{\eta}{2} \|\mathcal{X}_{t+1} - \mathcal{X}_t\|_F^2. \quad (60)$$

- If step 5 is performed,

$$
\begin{aligned}
F_\tau(\mathcal{X}_{t+1}) \le F_\tau(\mathcal{V}_t) - \frac{\eta}{2}\left\|\mathcal{X}_{t+1} - \mathcal{V}_t\right\|_F^2 &\le F_\tau(\bar{\mathcal{X}}_t) - \frac{\eta}{2}\left\|\mathcal{X}_{t+1} - \bar{\mathcal{X}}_t\right\|_F^2, \\
&\le F_\tau(\mathcal{X}_t) - \frac{\eta}{2}\left\|\mathcal{X}_{t+1} - \bar{\mathcal{X}}_t\right\|_F^2. \qquad (61)
\end{aligned}
$$

Combining (60) and (61), we have

$$
\frac{2}{\eta}\left(F_\tau(\mathcal{X}_1) - F_\tau(\mathcal{X}_{T+1})\right) \ge \sum\nolimits_{j \in \chi_1(T)} \left\|\mathcal{X}_{t+1} - \bar{\mathcal{X}}_t\right\|_F^2 + \sum\nolimits_{j \in \chi_2(T)} \left\|\mathcal{X}_{t+1} - \mathcal{X}_t\right\|_F^2, \qquad (62)
$$

where $\chi_1(T)$ and $\chi_2(T)$ are a partition of $\{1, ..., T\}$ such that when $j \in \chi_1(T)$ step 5 is performed, and when $j \in \chi_2(T)$ step 7 is performed. As F_τ is bounded from below and $\lim_{\|\mathcal{X}\|_F \to \infty} F_\tau(\mathcal{X}) = \infty$, taking $T = \infty$ in (62), we have

$$
\sum\nolimits_{j \in \chi_1(\infty)} \left\|\mathcal{X}_{t+1} - \mathcal{Y}_t\right\|_F^2 + \sum\nolimits_{j \in \chi_2(\infty)} \left\|\mathcal{X}_{t+1} - \mathcal{X}_t\right\|_F^2 = c,
$$

where $c \le \frac{2}{\eta}\left[F_\tau(\mathcal{X}_1) - F_\tau^{\min}\right]$ is a positive constant. Thus, the sequence $\{\mathcal{X}_t\}$ is bounded, and it must have limit points. Besides, one of the following three cases must hold.

1. $\chi_1(\infty)$ is finite, $\chi_2(\infty)$ is infinite. Let $\tilde{\mathcal{X}}$ be a limit point of $\{\mathcal{X}_t\}$, and $\{\mathcal{X}_{j_t}\}$ be a subsequence that converges to $\tilde{\mathcal{X}}$. In this case, on using Lemma 24, we have

$$
\begin{aligned}
\lim_{j_t \to \infty} \left\|\mathcal{X}_{j_t+1} - \mathcal{X}_{j_t}\right\|_F^2 &= \lim_{j_t \to \infty} \left\|\text{prox}_{\frac{\bar{g}_\tau}{\tau}}\left(\mathcal{X}_{j_t} - \frac{1}{\tau}\nabla f(\mathcal{X}_{j_t})\right) - \mathcal{X}_{j_t}\right\|_F^2, \\
&= \left\|\text{prox}_{\frac{\bar{g}_\tau}{\tau}}\left(\tilde{\mathcal{X}} - \frac{1}{\tau}\nabla f(\tilde{\mathcal{X}})\right) - \tilde{\mathcal{X}}\right\|_F^2 = 0.
\end{aligned}
$$

   Thus, $\tilde{\mathcal{X}} = \text{prox}_{\frac{\bar{g}_\tau}{\tau}}(\tilde{\mathcal{X}} - \frac{1}{\tau}\nabla f(\tilde{\mathcal{X}}))$, and $\tilde{\mathcal{X}}$ is a critical point of F_τ from Lemma 23.

2. $\chi_1(\infty)$ is infinite, $\chi_2(\infty)$ is finite. Let $\tilde{\mathcal{X}}$ be a limit point of $\{\mathcal{X}_t\}$, and $\{\mathcal{X}_{j_t}\}$ be a subsequence that converges to $\tilde{\mathcal{X}}$. In this case, we have

$$
\begin{aligned}
\lim_{j_t \to \infty} \left\|\mathcal{X}_{j_t+1} - \mathcal{Y}_{j_t}\right\|_F^2 &= \lim_{j_t \to \infty} \left\|\text{prox}_{\frac{\bar{g}_\tau}{\tau}}\left(\mathcal{X}_{j_t} - \frac{1}{\tau}\nabla f(\mathcal{X}_{j_t})\right) - \mathcal{Y}_{j_t}\right\|_F^2, \\
&= \left\|\text{prox}_{\frac{\bar{g}_\tau}{\tau}}\left(\tilde{\mathcal{X}} - \frac{1}{\tau}\nabla f(\tilde{\mathcal{X}})\right) - \tilde{\mathcal{X}}\right\|_F^2 = 0.
\end{aligned}
$$

   Thus, $\tilde{\mathcal{X}} = \text{prox}_{\frac{\bar{g}_\tau}{\tau}}(\tilde{\mathcal{X}} - \frac{1}{\tau}\nabla f(\tilde{\mathcal{X}}))$, and $\tilde{\mathcal{X}}$ is a critical point of F_τ from Lemma 23.

3. Both $\chi_1(\infty)$ and $\chi_2(\infty)$ are infinite. From the above cases, we can see that either $\chi_1(\infty)$ or $\chi_2(\infty)$ is infinite, and limit points are also the critical points of F_τ.

Thus, all limit points of $\{\mathcal{X}_t\}$ are critical points of F_τ. ■

B.6 Corollary 10

This corollary can be easily derived from the proof of Theorem 9.
Proof Since $\mathcal{X}_{t+1} = \text{prox}_{\frac{\bar{g}_\tau}{\tau}}(\mathcal{V}_t - \frac{1}{\tau}\nabla f(\mathcal{V}_t))$, conclusion (i) directly follows from Lemma 23. From (62), we have

$$\begin{aligned}\min\nolimits_{1,\dots,T} \|\mathcal{X}_{t+1} - \mathcal{V}_t\|_F^2 \le \frac{1}{T}\sum\nolimits_{t=1\dots T} \|\mathcal{X}_{t+1} - \mathcal{V}_t\|_F^2\,, \\ \le \frac{2}{\eta T}(F_\tau(\mathcal{X}_1) - F_\tau(\mathcal{X}_{T+1})) \le \frac{2}{\eta T}(F_\tau(\mathcal{X}_1) - F_\tau^{\min}).\end{aligned}$$

Thus, we obtain Conclusion (ii). ■

B.7 Theorem 13

We first bound ∂F_τ in Lemma 25, then prove Theorem 13.

Lemma 25 *For iterations in Algorithm 2, we have* $\min_{\mathcal{U}_t \in \partial F_\tau(\mathcal{X}_t)} \|\mathcal{U}_t\|_F \le (\tau + \rho)\|\mathcal{X}_{t+1} - \mathcal{V}_t\|_F$.

Proof Since $\mathcal{X}_{t+1}$ is generated from the proximal step, i.e., $\mathcal{X}_{t+1} = \text{prox}_{\frac{\bar{g}_\tau}{\tau}}(\mathcal{V}_t - \frac{1}{\tau}\nabla f(\mathcal{V}))$, from its optimality condition, we have

$$\mathcal{X}_{t+1} - \big(\mathcal{V}_t - \frac{1}{\tau}\nabla f(\mathcal{V}_t)\big) + \frac{1}{\tau}\partial \bar{g}_\tau(\mathcal{X}_{t+1}) \ni \mathbf{0}.$$

Let $\mathcal{U}_t = \tau\left[\mathcal{X}_{t+1} - \mathcal{V}_t\right] - \left[\nabla f(\mathcal{V}_t) - \nabla f(\mathcal{X}_{t+1})\right]$. We have

$$\partial F_\tau(\mathcal{X}_{t+1}) = \left[\nabla f(\mathcal{X}_{t+1}) + \partial \bar{g}_\tau(\mathcal{X}_{t+1})\right] \in \mathcal{U}_t.$$

Thus, $\|\mathcal{U}_t\|_F \le \tau\|\mathcal{X}_{t+1} - \mathcal{V}_t\|_F + \|\nabla f(\mathcal{V}_t) - \nabla f(\mathcal{X}_{t+1})\|_F \le (\tau + \rho)\|\mathcal{X}_{t+1} - \mathcal{V}_t\|_F$. ■

Proof (**of Theorem 13**). From Theorem 9, we have $\lim\limits_{T\to\infty} F_\tau(\mathcal{X}_t) = F_\tau^{\min}$. Then, from Lemma 25, we have

$$\lim_{t\to\infty} \min\nolimits_{\mathcal{U}_t \in \partial F_\tau(\mathcal{X}_t)} \|\mathcal{U}_t\|_F \le \lim_{t\to\infty} (\tau + \rho)\|\mathcal{X}_{t+1} - \mathcal{V}_t\|_F = 0.$$

Thus, for any ϵ, $c > 0$ and $t > t_0$ where t_0 is a sufficiently large positive integer, we have

$$\mathcal{X}_t \in \left\{\mathcal{X} \,|\, \min\nolimits_{\mathcal{U} \in \partial F_\tau(\mathcal{X})} \|\mathcal{U}\|_F \le \epsilon, F_\tau^{\min} < F_\tau(\mathcal{X}) < F_\tau^{\min} + c\right\}.$$

Then, the uniformized KL property implies for all $t \ge t_0$,

$$\begin{aligned}1 &\le \psi'\left(F_\tau(\mathcal{X}_{t+1}) - F_\tau^{\min}\right) \min\nolimits_{\mathcal{U}_t \in \partial F_\tau(\mathcal{X}_t)} \|\mathcal{U}_t\|_F\,, \\ &= \psi'\left(F_\tau(\mathcal{X}_{t+1}) - F_\tau^{\min}\right)(\tau + \rho)\|\mathcal{X}_{t+1} - \mathcal{V}_t\|_F\,.\end{aligned} \tag{63}$$

Moreover, from Lemma 22, we have

$$\|\mathcal{X}_{t+1} - \mathcal{V}_t\|_F^2 \le \frac{2}{\eta}\left[F_\tau(\mathcal{V}_t) - F_\tau(\mathcal{X}_{t+1})\right]. \tag{64}$$

Let $r_t = F_\tau(\mathcal{X}_t) - F_\tau^{\min}$, we have

$$\begin{aligned} r_t - r_{t+1} &= F_\tau(\mathcal{X}_t) - F_\tau^{\min} - \left[F_\tau(\mathcal{X}_{t+1}) - F_\tau^{\min}\right], \\ &\geq F_\tau(\mathcal{V}_t) - F_\tau^{\min} - \left[F_\tau(\mathcal{X}_{t+1}) - F_\tau^{\min}\right] = F_\tau(\mathcal{V}_t) - F_\tau(\mathcal{X}_{t+1}). \end{aligned} \tag{65}$$

Combine (63), (64) and (65), we have

$$\begin{aligned} 1 &\leq \left[\psi'(r_t)\right]^2 (\tau+\rho)^2 \left\|\mathcal{X}_{t+1} - \mathcal{V}_t\right\|_F^2, \\ &\leq \frac{2(\tau+\rho)^2}{\eta}\left[\psi'(r_t)\right]^2 \left[F_\tau(\mathcal{V}_t) - F_\tau(\mathcal{X}_{t+1})\right] \leq \frac{2(\tau+\rho)^2}{\eta}\left[\psi'(r_{t+1})\right]^2 (r_t - r_{t+1}). \end{aligned} \tag{66}$$

Since $\phi(\alpha) = \frac{C}{x}\alpha^x$, then $\phi'(\alpha) = C\alpha^{x-1}$, (66) becomes $1 \leq d_1 C^2 r_{t+1}^{2x-2}(r_t - r_{t+1})$, where $d_1 = \frac{2(\tau+\rho)^2}{\eta}$. Finally, it is shown in (Bolte et al., 2014; Li and Lin, 2015; Li et al., 2017) that the sequence $\{r_t\}$ satisfying the above inequality, convergence to zero with different rates stated in the Theorem. ■

B.8 Lemma 15

First, we introduce the following Lemma.

Lemma 26 (Theorem 1 in (Negahban and Wainwright, 2012)) *Consider a matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$. Let $d = \frac{1}{2}(m+n)$ and $m_{rank}(\boldsymbol{X}) = \frac{\|\boldsymbol{X}\|_*}{\|\boldsymbol{X}\|_F}$. Define a constraint set $\mathcal{C}$ (with parameters c_0, n) as*

$$\mathcal{C}(n, c_0) = \left\{\boldsymbol{X} \in \mathbb{R}^{m\times n}, \boldsymbol{X} \neq 0 \mid m_{spike}(\boldsymbol{X}) \cdot m_{rank}(\boldsymbol{X}) \leq \frac{1}{c_0 L}\sqrt{\frac{\|\boldsymbol{\Omega}\|_1}{d\log d}}\right\},$$

where L is a constant. There are constants (c_0, c_1, c_2, c_3) such that when $\|\boldsymbol{\Omega}\|_1 > c_3 \max(d\log d)$, we have

$$\frac{\|P_{\boldsymbol{\Omega}}(\boldsymbol{X})\|_F}{\|\boldsymbol{\Omega}\|_1} \geq \frac{1}{8}\|\boldsymbol{X}\|_F \left\{1 - \frac{128L\cdot m_{spike}(\boldsymbol{X})}{\sqrt{\|\boldsymbol{\Omega}\|_1}}\right\}, \quad \forall \boldsymbol{X} \in \mathcal{C}(\|\boldsymbol{\Omega}\|_1, c_0),$$

with a high probability greater at least of $1 - c_1\exp(-c_2 d\log d)$.

Proof (of Lemma 15) For a Mth-order tensor Δ, using Lemma 26 on each unfolded matrix $\Delta_{\langle i\rangle}$ ($i = 1, \dots, M$), we have

$$\frac{\left\|P_{\boldsymbol{\Omega}}\left(\Delta_{\langle i\rangle}\right)\right\|_F}{\|\boldsymbol{\Omega}\|_1} \geq \frac{1}{8}\|\Delta\|_F \left\{1 - \frac{128L\cdot m_{\text{spike}}(\Delta_{\langle i\rangle})}{\sqrt{\|\boldsymbol{\Omega}\|_1}}\right\}, \tag{67}$$

for all $\Delta_{\langle i\rangle} \in \mathcal{C}^i(\|\boldsymbol{\Omega}\|_1, c_0)$. Note that the L.H.S. of (67) is the same for all $i = 1, ..., M$. Thus, to ensure (67) holds for all $\Delta_{\langle i\rangle}$, we need to take the intersection of all $\Delta_{\langle i\rangle}$, which leads to

$$\left\{\mathcal{X} \in \mathbb{R}^{I_1\times\ldots\times I_M}, \mathcal{X} \neq 0 \mid m_{\text{spike}}(\mathcal{X}) \cdot m_{\text{rank}}(\mathcal{X}_{\langle i\rangle}) \leq \frac{1}{c_0 L}\sqrt{\frac{\|\boldsymbol{\Omega}\|_1}{d_i \log d_i}}\right\}. \tag{68}$$

Recall that $m_{\text{rank}}(\mathcal{X}) = \frac{1}{M}\sum_{i=1}^M m_{\text{rank}}(\mathcal{X}_{\langle i\rangle})$ as defined in (33). Thus, $\tilde{\mathcal{C}}(n, c_0)$ is a subset of (68). As a result, (36) holds. ■

B.9 Theorem 16

Here, we first introduce some auxiliary lemmas in Appendix B.9.1, which will be used to prove Theorem 16 in Appendix B.9.2.

B.9.1 AUXILIARY LEMMAS

Lemma 27 (Lemma 4 in (Loh and Wainwright, 2015)) *For κ in Assumption 1, we have*

(i). The function $\alpha \rightarrow \frac{\kappa(\alpha)}{\alpha}$ is nonincreasing on $\alpha > 0$;

(ii). The derivative of κ is upper bounded by κ_0;

(iii). The function $\alpha \rightarrow \kappa(\alpha) + \frac{\alpha^2 c}{2}$ is convex only when $c \geq \kappa_0$;

(iv). $\lambda|\alpha| \leq \lambda\kappa(|\alpha|) + \frac{\alpha^2 \kappa_0}{2}$.

Lemma 28 $\langle \mathcal{X}, \mathcal{Y} \rangle \leq \min_{i=1,\dots,K} \left\|\mathcal{X}_{\langle i \rangle}\right\|_\infty \left\|\mathcal{Y}_{\langle i \rangle}\right\|_*$.

Proof First, we have $\langle \mathcal{X}, \mathcal{Y} \rangle = \left\langle \mathcal{X}_{\langle i \rangle}, \mathcal{Y}_{\langle i \rangle} \right\rangle$ for all $i \in \{1, \dots, M\}$. Then, since $\|\cdot\|_\infty$ and $\|\cdot\|_*$ are dual norm with each other, $\left\langle \mathcal{X}_{\langle i \rangle}, \mathcal{Y}_{\langle i \rangle} \right\rangle \leq \left\|\mathcal{X}_{\langle i \rangle}\right\|_\infty \left\|\mathcal{Y}_{\langle i \rangle}\right\|_*$. Thus, we have $\langle \mathcal{X}, \mathcal{Y} \rangle \leq \min_{i=1,\dots,K} \left\|\mathcal{X}_{\langle i \rangle}\right\|_\infty \left\|\mathcal{Y}_{\langle i \rangle}\right\|_*$. ■

Lemma 29 *For all $i \in \{1, ...m\}$, we have $\|\mathcal{X}\|_F \leq \left\|\mathcal{X}_{\langle i \rangle}\right\|_*$ and $\left\|\mathcal{X}_{\langle i \rangle}\right\|_* \leq \sqrt{\min(I^i, \frac{I^\pi}{I^i})}\, \|\mathcal{X}\|_F$.*

Proof Note that $\|\mathcal{X}\|_F = \left\|\mathcal{X}_{\langle i \rangle}\right\|_F$ and $\left\|\mathcal{X}_{\langle i \rangle}\right\|_F \leq \left\|\mathcal{X}_{\langle i \rangle}\right\|_*$, thus $\|\mathcal{X}\|_F \leq \left\|\mathcal{X}_{\langle i \rangle}\right\|_*$. Then, since $\|\boldsymbol{X}\|_* \leq \sqrt{\min(p, q)}\, \|\boldsymbol{X}\|_F$ for a matrix $\boldsymbol{X}$ of size $p \times q$, we have $\left\|\mathcal{X}_{\langle i \rangle}\right\|_* \leq \sqrt{\min(I^i, I^\pi / I^i)}\, \left\|\mathcal{X}_{\langle i \rangle}\right\|_F = \sqrt{\min(I^i, I^\pi / I^i)}\, \|\mathcal{X}\|_F$. ■

Lemma 30 *Define $h_i(\mathcal{X}) = \phi(\mathcal{X}_{\langle i \rangle})$. Let $\Phi_k(\boldsymbol{A})$ produce the best rank k approximation to matrix $\boldsymbol{A}$ and $\Psi_k(\boldsymbol{A}) = \boldsymbol{A} - \Phi_k(\boldsymbol{A})$. Suppose $\varepsilon_i > 0$ for $i \in \{1, ..., M\}$ are constants such that $\varepsilon_i h_i(\Phi_{k_i}(\mathcal{A}_{\langle i \rangle})) - h_i(\Psi_{k_i}(\mathcal{A}_{\langle i \rangle})) \geq 0$. Then,*

$$\varepsilon_i h_i(\Phi_{k_i}(\mathcal{A}_{\langle i \rangle})) - h_i(\Psi_{k_i}(\mathcal{A}_{\langle i \rangle})) \leq \kappa_0(\varepsilon_i \left\|\Phi_{k_i}(\mathcal{A}_{\langle i \rangle})\right\|_* - \left\|\Psi_{k_i}(\mathcal{A}_{\langle i \rangle})\right\|_*). \tag{69}$$

Moreover, if $\mathcal{X}^*_{\langle i \rangle}$ is of rank k_i, for any tensor $\mathcal{X}$ satisfying $\varepsilon_i h_i(\mathcal{X}^*_{\langle i \rangle}) - h_i(\mathcal{X}_{\langle i \rangle}) \geq 0$ and $\varepsilon_i > 1$, we have

$$\varepsilon_i h_i(\mathcal{X}^*_{\langle i \rangle}) - h_i(\mathcal{X}_{\langle i \rangle}) \leq \kappa_0(\varepsilon_i \left\|\Phi_{k_i}(\mathcal{V}_{\langle i \rangle})\right\|_* - \left\|\Psi_{k_i}(\mathcal{V}_{\langle i \rangle})\right\|_*), \tag{70}$$

where $\mathcal{V} = \mathcal{X}^* - \mathcal{X}$.

Proof **We first prove** (69). Let $h(\alpha) = \frac{\alpha}{\kappa(\alpha)}$ on $\alpha > 0$. From Lemma 27, we know $h(\alpha)$ is a non-decreasing function. Therefore,

$$\begin{aligned} \left\|\Psi_{k_i}(\mathcal{A}_{\langle i \rangle})\right\|_* &= \sum\nolimits_{j=k_i+1} \kappa\left(\sigma_j\left(\mathcal{A}_{\langle i \rangle}\right)\right) h\left(\sigma_j\left(\mathcal{A}_{\langle i \rangle}\right)\right), \\ &\leq h\left(\sigma_1\left(\mathcal{A}_{\langle i \rangle}\right)\right) \sum\nolimits_{j=k_i+1} \kappa\left(\sigma_j\left(\mathcal{A}_{\langle i \rangle}\right)\right) = h\left(\sigma_1\left(\mathcal{A}_{\langle i \rangle}\right)\right) \cdot h_i\left(\Psi_{k_i}(\mathcal{A}_{\langle i \rangle})\right). \end{aligned} \tag{71}$$

Again, using non-decreasing property of h, we have

$$\begin{aligned} h_i\left(\mathit{\Phi}_{k_i}(\mathcal{A}_{\langle i\rangle})\right) h\left(\sigma_{k_i+1}\left(\mathcal{A}_{\langle i\rangle}\right)\right) &= h\left(\sigma_{k_i+1}\left(\mathcal{A}_{\langle i\rangle}\right)\right) \sum\nolimits_{j=1}^{k_i} \kappa\left(\sigma_j\left(\mathcal{A}_{\langle i\rangle}\right)\right), \\ &\le \sum\nolimits_{j=1}^{k_i} \kappa\left(\sigma_j\left(\mathcal{A}_{\langle i\rangle}\right)\right) h\left(\sigma_j\left(\mathcal{A}_{\langle i\rangle}\right)\right) = \left\|\mathit{\Phi}_{k_i}(\mathcal{A}_{\langle i\rangle})\right\|_*. \end{aligned} \tag{72}$$

Note that $h(\alpha) \ge 1/\kappa_0$ from Lemma 27, and combining (71) and (72), we have

$$\begin{aligned} 0 \le \varepsilon_i \cdot h_i(\mathit{\Phi}_{k_i}(\mathcal{A}_{\langle i\rangle})) - h_i(\mathit{\Psi}_{k_i}(\mathcal{A}_{\langle i\rangle})) &\le \left(\varepsilon_i \left\|\mathit{\Phi}_{k_i}(\mathcal{A}_{\langle i\rangle})\right\|_* - \left\|\mathit{\Psi}_{k_i}(\mathcal{A}_{\langle i\rangle})\right\|_*\right) / h\left(\sigma_1(\mathcal{A}_{\langle i\rangle})\right) \\ &\le \kappa_0 \left(\varepsilon_i \left\|\mathit{\Phi}_{k_i}(\mathcal{A}_{\langle i\rangle})\right\|_* - \left\|\mathit{\Psi}_{k_i}(\mathcal{A}_{\langle i\rangle})\right\|_*\right). \end{aligned}$$

Thus, (69) is obtained. **Next, we prove** (70). The triangle inequality and subadditivity of h_i (see Lemma 5 in (Loh and Wainwright, 2015)) imply that

$$\begin{aligned} 0 \le \varepsilon_i \cdot h_i(\mathcal{X}^*_{\langle i\rangle}) - h_i(\mathcal{X}_{\langle i\rangle}) &= \varepsilon_i \cdot h_i(\mathit{\Phi}_{m_i}(\mathcal{X}^*_{\langle i\rangle})) - h_i(\mathit{\Phi}_{m_i}(\mathcal{X}_{\langle i\rangle})) - h_i(\mathit{\Psi}_{m_i}(\mathcal{X}_{\langle i\rangle})), \\ &\le \varepsilon_i \cdot h_i\left(\mathit{\Phi}_{m_i}\left(\mathcal{V}_{\langle i\rangle}\right)\right) - h_i\left(\mathit{\Psi}_{m_i}\left(\mathcal{V}_{\langle i\rangle}\right)\right), \\ &\le \kappa_0 \left(\varepsilon_i \left\|\mathit{\Phi}_{k_i}(\mathcal{V}_{\langle i\rangle})\right\|_* - \left\|\mathit{\Psi}_{k_i}(\mathcal{V}_{\langle i\rangle})\right\|_*\right). \end{aligned}$$

Thus, (70) is obtained. ■

Lemma 31 $\left\|\partial\phi(\boldsymbol{X})\right\|_\infty \le \kappa_0$ *where* ϕ *is defined in* (7).

Proof Let $\boldsymbol{X}$ be of size $m \times n$ with $m \le n$, and SVD of $\boldsymbol{X}$ be $\boldsymbol{U\Sigma V}^\top$ where $\boldsymbol{\Sigma} = \text{Diag}\,(\sigma_1, \dots, \sigma_m)$. From Theorem 3.7 in (Lewis and Sendov, 2005), we have

$$\partial\phi(\boldsymbol{X}) = \boldsymbol{U}\,\text{Diag}\left(\kappa'(\sigma_1), ..., \kappa'(\sigma_m)\right)\boldsymbol{V}^\top.$$

From Lemma 27, we have $\kappa'(\sigma_1) \le \kappa'(\sigma_2) \le ... \le \kappa_0$. Since $\left\|\boldsymbol{X}\right\|_\infty$ returns the maximum singular value of $\boldsymbol{X}$, we have $\left\|\partial\phi(\boldsymbol{X})\right\|_\infty \le \kappa'(\sigma_m) \le \kappa_0$. ■

Lemma 32 $\phi(\boldsymbol{X}) + \frac{\kappa_0}{2}\left\|\boldsymbol{X}\right\|_F^2$ *is convex.*

Proof Using the definition of ϕ in (7) and the fact $\left\|\boldsymbol{X}\right\|_F^2 = \sum_i \sigma_i(\boldsymbol{X})$, we have

$$\gamma(\boldsymbol{X}) = \phi(\boldsymbol{X}) + \kappa_0/2\left\|\boldsymbol{X}\right\|_F^2 = \sum\nolimits_i \psi(\sigma_i(\boldsymbol{X})),$$

where $\psi(\alpha) = \kappa(\alpha) + \kappa_0\alpha^2/2$. Since $\psi(\alpha)$ is convex (Lemma 27), $\gamma(\boldsymbol{X})$ is convex (using Proposition 6.1 in (Lewis and Sendov, 2005)). ■

B.9.2 PROOF OF THEOREM 16

Proof Part 1). Let $\tilde{\mathcal{V}} = \tilde{\mathcal{X}} - \mathcal{X}^*$, we begin by proving $\|\tilde{\mathcal{V}}\|_F \leq 1$. If not, then the second condition in (35) holds, i.e.,

$$\left\langle \nabla f(\tilde{\mathcal{X}}) - \nabla f(\mathcal{X}^*), \tilde{\mathcal{V}} \right\rangle \geq \alpha_2 \|\tilde{\mathcal{V}}\|_F^2 - \tau_2 \sqrt{\log I^\pi / \|\mathbf{\Omega}\|_1} \sum\nolimits_{i=1}^M \left\|\Delta_{\langle i \rangle}\right\|_* . \tag{73}$$

Since $\tilde{\mathcal{X}}$ is a first-order critical point, then

$$\left\langle \nabla f(\tilde{\mathcal{X}}) + \partial r(\tilde{\mathcal{X}}), \mathcal{X} - \tilde{\mathcal{X}} \right\rangle \geq 0, \tag{74}$$

$$\nabla f(\tilde{\mathcal{X}}) + \partial r(\tilde{\mathcal{X}}) \ni \mathbf{0}. \tag{75}$$

Taking $\mathcal{X} = \mathcal{X}^*$, from (74), we have

$$\left\langle \nabla f(\tilde{\mathcal{X}}) + \partial r(\tilde{\mathcal{X}}), -\tilde{\mathcal{V}} \right\rangle \geq 0. \tag{76}$$

Combining (73) and (76), we have

$$\left\langle -\partial r(\tilde{\mathcal{X}}) - \nabla f\left(\mathcal{X}^*\right), \tilde{\mathcal{V}} \right\rangle \geq \alpha_2 \|\tilde{\mathcal{V}}\|_F^2 - \tau_2 \sqrt{\log I^\pi / \|\mathbf{\Omega}\|_1} \sum\nolimits_{i=1}^M \left\|\Delta_{\langle i \rangle}\right\|_* . \tag{77}$$

Let $\tilde{v}_i = \|\tilde{\mathcal{V}}_{\langle i \rangle}\|_*$ and $\tilde{v} = \sum_{i=1}^M \tilde{v}_i$. For the L.H.S of (77),

$$\left\langle \partial r(\tilde{\mathcal{X}}) + \nabla f\left(\mathcal{X}^*\right), \tilde{\mathcal{V}} \right\rangle = \left\langle \nabla f\left(\mathcal{X}^*\right), \tilde{\mathcal{V}} \right\rangle + \lambda \sum\nolimits_{i=1}^M \left\langle \partial \phi(\mathcal{X}_{\langle i \rangle}), \tilde{\mathcal{V}}_{\langle i \rangle} \right\rangle \tag{78}$$

$$\leq \max_i \left\| [\nabla f\left(\mathcal{X}^*\right)]_{\langle i \rangle} \right\|_\infty \tilde{v}_i + \lambda \sum\nolimits_{i=1}^M \left\| \partial \phi(\mathcal{X}_{\langle i \rangle}) \right\|_\infty \tilde{v}_i, \tag{79}$$

Next, note that the following inequalities hold.

- From the left part of (37) in Theorem 16, we have $\max_i \left\| [\nabla f\left(\mathcal{X}^*\right)]_{\langle i \rangle} \right\|_\infty \leq \frac{\lambda \kappa_0}{4}$.
- From Lemma 31, we have $\|\partial \phi(\boldsymbol{X})\|_\infty \leq \kappa_0$.

Combining with (79), we have

$$\left\langle \partial r(\tilde{\mathcal{X}}) + \nabla f\left(\mathcal{X}^*\right), \tilde{\mathcal{V}} \right\rangle \leq \frac{\lambda \kappa_0}{4} + 3\lambda\kappa_0 = \frac{13\lambda\kappa_0}{4}. \tag{80}$$

Combining (77) and (80), then rearranging terms, we have

$$\|\tilde{\mathcal{V}}\|_F \leq \frac{1}{\alpha_2}\left(\tau_2 \sqrt{\log I^\pi / \|\mathbf{\Omega}\|_1} + \lambda\kappa_0\right) \tilde{v} \leq \frac{1}{\alpha_2}\left(\tau_2 \sqrt{\log I^\pi / \|\mathbf{\Omega}\|_1} + \frac{13\lambda\kappa_0}{4}\right) R.$$

Finally, using assumptions on $\|\mathbf{\Omega}\|_1$ and λ, we have $\|\tilde{\mathcal{V}}\|_F \leq 1$, which is in the contradiction with our assumption at the beginning of Part 1). Thus, $\|\tilde{\mathcal{V}}\|_F \leq 1$ must hold.

Part 2). Let $h_i(\mathcal{X}) = \phi(\mathcal{X}_{\langle i \rangle})$. Since the function $h_i(\mathcal{X}) + \mu/2 \, \|\mathcal{X}\|_F^2$ is convex (Lemma 32), we have

$$\left\langle \partial h_i(\tilde{\mathcal{X}}), \mathcal{X}^* - \tilde{\mathcal{X}} \right\rangle \leq \tilde{h}_i(\tilde{\mathcal{X}}). \tag{81}$$

where $\tilde{h}_i(\tilde{\mathcal{X}}) = h_i(\mathcal{X}^*) - h_i(\tilde{\mathcal{X}}) + \frac{L}{2}\|\tilde{\mathcal{X}} - \mathcal{X}^*\|_F^2$. From the first condition in (35), we have

$$\langle \nabla f(\tilde{\mathcal{V}}) - \nabla f\left(\mathcal{X}^*\right), -\tilde{\mathcal{X}}\rangle \geq \alpha_1 \|\tilde{\mathcal{V}}\|_F^2 - \tau_1 \frac{\log I^\pi}{\|\mathbf{\Omega}\|_1}\tilde{v}^2. \tag{82}$$

Combining (74) and (82), we have

$$\begin{aligned}\alpha_1 \|\tilde{\mathcal{V}}\|_F^2 - \tau_1 \frac{\log I^\pi}{\|\mathbf{\Omega}\|_1}\tilde{v}^2 &\leq \left\langle \partial r(\tilde{\mathcal{X}}), \tilde{\mathcal{V}} \right\rangle - \left\langle \nabla f(\mathcal{X}^*), \tilde{\mathcal{V}} \right\rangle, \\ &= \lambda \sum\nolimits_{i=1}^M \left\langle \partial h_i(\tilde{\mathcal{X}}), \tilde{\mathcal{V}} \right\rangle - \left\langle \nabla f(\mathcal{X}^*), \tilde{\mathcal{V}} \right\rangle.\end{aligned}$$

Together with (81), we have

$$\begin{aligned}\alpha_1 \|\tilde{\mathcal{V}}\|_F^2 - \tau_1 \frac{\log I^\pi}{\|\mathbf{\Omega}\|_1}\tilde{v}^2 &\leq \lambda \sum\nolimits_{i=1}^M \tilde{h}_i(\tilde{\mathcal{X}}) - \left\langle \nabla f(\mathcal{X}^*), \tilde{\mathcal{V}} \right\rangle, \\ &\leq \lambda \sum\nolimits_{i=1}^M \tilde{h}_i(\tilde{\mathcal{X}}) + \max_i \left\| [\nabla f(\mathcal{X}^*)]_{\langle i \rangle} \right\|_\infty \tilde{v}_i \\ &\leq \lambda \sum\nolimits_{i=1}^M \tilde{h}_i(\tilde{\mathcal{X}}) + \max_i \left\| [\nabla f(\mathcal{X}^*)]_{\langle i \rangle} \right\|_\infty \tilde{v},\end{aligned}$$

where the second inequality is from Lemma 28. Rearranging items in the above inequality, we have

$$\left(\alpha_1 - \frac{\mu M}{2}\right)\|\tilde{\mathcal{V}}\|_F^2 \leq \lambda \sum\nolimits_{i=1}^M \left(h_i(\mathcal{X}^*) - h_i(\tilde{\mathcal{X}})\right) + \left(\max_i \left\| [\nabla f(\mathcal{X}^*)]_{\langle i \rangle} \right\|_\infty + \tau_1 \frac{\log I^\pi}{\|\mathbf{\Omega}\|_1}\tilde{v}\right)\tilde{v}. \tag{83}$$

Note that from the Assumption in Theorem 16, we have the following inequalities.

- $\max_i \left\| \left[\nabla f(\mathcal{X}^*)\right]_{\langle i \rangle} \right\|_\infty \leq \kappa_0 \lambda / 4$.
- Since $\|\mathbf{\Omega}\|_1 \geq 16 R^2 \max\left(\tau_1^2, \tau_2^2\right) \log(I^\pi)/\alpha_2^2$ and $\alpha_2 \sqrt{\log I^\pi / \|\mathbf{\Omega}\|_1} \leq \kappa_0 \lambda / 4$, then

$$\frac{\tau_1 \log I^\pi}{\|\mathbf{\Omega}\|_1}\tilde{v} \leq \frac{\tau_1}{\alpha_2}\sqrt{\frac{\log I^\pi}{\|\mathbf{\Omega}\|_1}}\tilde{v} \cdot \alpha_2 \sqrt{\frac{\log I^\pi}{\|\mathbf{\Omega}\|_1}} \leq \frac{\tau_1}{\alpha_2}\sqrt{\frac{\alpha_2^2 \log I^\pi}{16 \tilde{v}^2 \tau_1^2}}\tilde{v} \cdot \alpha_2 \sqrt{\frac{\log I^\pi}{\|\mathbf{\Omega}\|_1}} \leq \frac{\lambda \kappa_0}{4}.$$

Combing above inequalities into (83), we further have

$$\left(\alpha_1 - \frac{\mu M}{2}\right)\|\tilde{\mathcal{V}}\|_F^2 \leq \lambda \sum\nolimits_{i=1}^M \left(h_i(\mathcal{X}^*) - h_i(\tilde{\mathcal{X}})\right) + \frac{\lambda \kappa_0}{2}\tilde{v}. \tag{84}$$

Part 3). Combining (84) and Lemma 27, as well as the subadditivity of h_i, we have

$$\begin{aligned}\left(\alpha_1 - \frac{LM}{2}\right)\|\tilde{\mathcal{V}}\|_F^2 &\leq \lambda \sum\nolimits_{i=1}^M \left(h_i(\mathcal{X}^*) - h_i(\tilde{\mathcal{X}})\right) + \frac{\lambda \kappa_0}{2}\left(\frac{\sum_{i=1}^M h_i(\tilde{\mathcal{V}})}{LM} + \frac{LM}{2\lambda\kappa_0}\|\tilde{\mathcal{V}}\|_F^2\right), \\ &\leq \lambda \sum\nolimits_{i=1}^M \left(h_i(\mathcal{X}^*) - h_i(\tilde{\mathcal{X}})\right) + \frac{\lambda \sum_{i=1}^M h_i(\mathcal{X}^*) + h_i(\tilde{\mathcal{X}})}{2D} + \frac{LM}{4}\|\tilde{\mathcal{V}}\|_F^2.\end{aligned} \tag{85}$$

Next, define

$$a_v = \alpha_1 - \frac{3M}{4}\kappa_0, \quad b_v = 1 + \frac{1}{2M}, \quad c_v = 1 - \frac{1}{2M}.$$

Rearranging terms in (85), we have

$$a_v \|\tilde{\mathcal{V}}\|_F^2 \leq \lambda \sum\nolimits_{i=1}^{M} b_v h_i(\mathcal{X}^*) - c_v h_i(\tilde{\mathcal{X}}). \tag{86}$$

From Lemma 30, we have

$$b_v h_i(\mathcal{X}^*) - c_v h_i(\tilde{\mathcal{X}}) \leq L\big(b_v \left\|\mathit{\Phi}_{k_i}(\tilde{\mathcal{V}}_{\langle i \rangle})\right\|_* - c_v \left\|\mathit{\Psi}_{k_i}(\tilde{\mathcal{V}}_{\langle i \rangle})\right\|_*\big). \tag{87}$$

Besides, we have the cone condition

$$\left\|\mathit{\Phi}_{k_i}(\tilde{\mathcal{V}}_{\langle i \rangle})\right\|_* \leq \frac{c_v}{b_v} \left\|\mathit{\Psi}_{k_i}(\tilde{\mathcal{V}}_{\langle i \rangle})\right\|_*. \tag{88}$$

Combining (86), (87) and (88), we have

$$\begin{aligned} a_v \|\tilde{\mathcal{V}}\|_F^2 &\leq \lambda \kappa_0 \sum\nolimits_{i=1}^{M} \big(b_v \left\|\mathit{\Phi}_{k_i}(\tilde{\mathcal{V}}_{\langle i \rangle})\right\|_* - c_v \left\|\mathit{\Psi}_{k_i}(\tilde{\mathcal{V}}_{\langle i \rangle})\right\|_*\big), \\ &\leq \lambda \kappa_0 \sum\nolimits_{i=1}^{M} b_v \left\|\mathit{\Phi}_{k_i}(\tilde{\mathcal{V}}_{\langle i \rangle})\right\|_* \leq \lambda \kappa_0 \sum\nolimits_{i=1}^{M} c_v \sqrt{k_i} \|\tilde{\mathcal{V}}\|_F. \end{aligned}$$

where the last inequality comes from Lemma 29. Since $a_v > 0$ as assumed, we conclude that

$$\|\tilde{\mathcal{V}}\|_F \leq \frac{\lambda \kappa_0 c_v}{a_v} \sum\nolimits_{i=1}^{M} \sqrt{k_i}.$$

which proves the theorem. ■

B.10 Corollary 17

Proof When noisy level is sufficiently small, (37) reduces to

$$\sqrt{\log I^{\pi} / \|\mathbf{\Omega}\|_1} \leq \lambda \leq \frac{1}{4R\kappa_0}. \tag{89}$$

Let $\lambda = b_1 \max_i \| \left[P_{\mathbf{\Omega}}\left(\mathcal{E}\right)\right]_{\langle i \rangle} \|_\infty$ where $b_1 \in \left[\frac{4}{\kappa_0}, \frac{\alpha_2}{4R\kappa_0 \max_i \|[P_{\mathbf{\Omega}}(\mathcal{E})]_{\langle i \rangle}\|_\infty}\right]$. It is easy to check (89) holds. Then, from Theorem 16, we will have

$$\left\|\mathcal{X}^* - \tilde{\mathcal{X}}\right\|_F \leq b_1 \max_i \| \left[P_{\mathbf{\Omega}}\left(\mathcal{E}\right)\right]_{\langle i \rangle} \|_\infty \cdot \frac{\kappa_0 c_v}{a_v} \sum\nolimits_{i=1}^{M} \sqrt{k_i}. \tag{90}$$

Next, note that

$$\mathbb{E}\left[\| \left[P_{\mathbf{\Omega}}\left(\mathcal{E}\right)\right]_{\langle i \rangle} \|_\infty\right] \leq \mathbb{E}\left[\| \left[P_{\mathbf{\Omega}}\left(\mathcal{E}\right)\right]_{\langle i \rangle} \|_F\right] = \mathbb{E}\left[\|\xi \cdot \mathbf{\Omega}\|_F\right] = \sigma \|\mathbf{\Omega}\|_F \leq \sigma \sqrt{I^{\pi}}. \tag{91}$$

Combining (90) and (91), we then have

$$\mathbb{E}\left[\left\|\mathcal{X}^* - \tilde{\mathcal{X}}\right\|_F\right] \leq \sigma \frac{\kappa_0 c_v \sqrt{I^{\pi}}}{a_v} \sum\nolimits_{i=1}^{M} \sqrt{k_i}.$$

■

B.11 Corollary 18

Proof When $\|\mathbf{\Omega}\|_1$ is sufficiently larger, (37) reduces to

$$4\sqrt{\log I^{\pi} / \|\mathbf{\Omega}\|_1} \le \lambda \le \frac{1}{4R}. \tag{92}$$

Let $\lambda = b_3 \sqrt{\frac{\log I^{\pi}}{\|\mathbf{\Omega}\|_1}}$ where $b_3 \in \left[4, \frac{1}{4R\sqrt{\log I^{\pi}/\|\mathbf{\Omega}\|_1}}\right]$. It is easy to check (92) holds. Then, from Theorem 16, we will have

$$\|\mathcal{X}^{*} - \tilde{\mathcal{X}}\|_F \le b_3 \sqrt{\frac{\log I^{\pi}}{\|\mathbf{\Omega}\|_1}} \cdot \frac{\kappa_0 c_v}{a_v} \sum\nolimits_{i=1}^{M} \sqrt{k_i}.$$

■

B.12 Proposition 19

Proof First, from Proposition 1 in (Yao and Kwok, 2018), we know that the function $\kappa(|a|) - \kappa_0 \cdot |a|$ is smooth. Since $\tilde{\ell}$ is also smooth, thus $\tilde{\kappa}_{\ell}$ is differentiable. Finally, note that $\lim_{\delta \to 0} \ell(a; \delta) = |a|$. Then, we have

$$\begin{aligned} \lim\nolimits_{\delta \to 0} \tilde{\kappa}_{\ell}(|a|; \delta) &= \lim\nolimits_{\delta \to 0} \left[\kappa_0 \cdot \tilde{\ell}(|a|; \delta) + (\kappa_{\ell}(|a|) - \kappa_0 \cdot |a|)\right], \\ &= \kappa_0 |a| + (\kappa_{\ell}(|a|) - \kappa_0 |a|) = \kappa_{\ell}(|a|). \end{aligned}$$

Thus, the Proposition holds. ■

B.13 Theorem 20

Proof First, by the definition of $\tilde{\kappa}_{\ell}$ in (41), when $|a| \le \delta$, we have

$$\lim_{\delta \to 0} \partial \tilde{\kappa}_{\ell}(a; \delta) = \frac{a}{\delta} \kappa_0 \in \begin{cases} [0, \kappa_0) & \text{if } a \ge 0 \\ (-\kappa_0, 0) & \text{otherwise} \end{cases}.$$

Thus,

$$\lim\nolimits_{\delta \to 0} \partial \tilde{\kappa}_{\ell}(a; \delta) = \partial \kappa_{\ell}(|a|). \tag{93}$$

Define $\tilde{F}_{\tau}(\mathcal{X}; \delta) = \sum_{(i_1 \dots i_M) \in \mathbf{\Omega}} \tilde{\ell}\,(\mathcal{X}_{i_1 \dots i_M} - \mathcal{O}_{i_1 \dots i_M}; \delta) + \sum_{i=1}^{D} \lambda_i \phi(\mathcal{X}_{\langle i \rangle})$. Since $\mathcal{X}_s$ is obtained from solving (42) at step 4 of Algorithm 3, we have $\mathcal{X}_s \in \partial \tilde{F}_{\tau}(\mathcal{X}; (\delta_0)^s)$. Take $s \to \infty$ and use (93), we have $\lim_{s \to \infty} \mathcal{X}_s \in \lim_{s \to \infty} \partial \tilde{F}_{\tau}(\mathcal{X}; (\delta_0)^s) = \lim_{\delta \to 0} \partial \tilde{F}_{\tau}(\mathcal{X}; \delta) = \partial \tilde{F}_{\tau}(\mathcal{X})$. Thus, Theorem 20 holds. ■

References

E. Acar, D.M. Dunlavy, T. Kolda, and M. Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106(1):41–56, 2011.

A. Agarwal, S. Negahban, and M. J. Wainwright. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In *Advances in Neural Information Processing Systems*, pages 37–45, 2010.

H. Attouch, J. Bolte, and B. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss-Seidel methods. *Mathematical Programming*, 137(1-2):91–129, 2013.

B. Bader and T. Kolda. Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 30(1):205–231, 2007.

M. Bahadori, Q. Yu, and Y. Liu. Fast multivariate spatio-temporal analysis via low rank tensor learning. In *Advances in Neural Information Processing Systems*, pages 3491–3499, 2014.

I. Balazevic, C. Allen, and T. Hospedales. TuckER: Tensor factorization for knowledge graph completion. In *Conference on Empirical Methods in Natural Language Processing*, pages 5188–5197, 2019.

H. Bauschke, R. Goebel, Y. Lucet, and X. Wang. The proximal average: Basic theory. *SIAM Journal on Optimization*, 19(2):766–785, 2008.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(Nov): 2399–2434, 2006.

J. Bengua, H. Phien, H. Tuan, and M. Do. Efficient tensor completion for color image and video recovery: Low-rank tensor train. *IEEE Transactions on Image Processing*, 26(5):2466–2479, 2017.

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD International Conference on Management of Data*, pages 1247–1250, 2008.

J. Bolte, A. Daniilidis, O. Ley, and L. Mazet. Characterizations of lojasiewicz inequalities and applications. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2010.

J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization or nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

N. Boumal and P.-A. Absil. Low-rank matrix completion via preconditioned optimization on the grassmann manifold. *Linear Algebra and its Applications*, 475:200–239, 2015.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2009.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

E. J Candès and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5-6):877–905, 2008.

E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11, 2011.

X. Chen, S. Liu, K. Xu, X. Li, X. Lin, M. Hong, and D. Cox. ZO-AdaMM: Zeroth-order adaptive momentum method for black-box optimization. In *Advances in Neural Information Processing Systems*, pages 7204–7215, 2019.

X. Chen, J. Yang, and L. Sun. A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies*, 117:102673, 2020.

H. Cheng, Y. Yu, X. Zhang, E. Xing, and D. Schuurmans. Scalable and sound low-rank tensor learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1114–1123, 2016.

A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163, 2015.

D. Davis, R. Lichtenwalter, and N. V. Chawla. Multi-relational link prediction in heterogeneous information networks. In *International Conference on Advances in Social Networks Analysis and Mining*, pages 281–288, 2011.

T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel. Convolutional 2D knowledge graph embeddings. In *AAAI Conference on Artificial Intelligence*, 2018.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32 (2):407–499, 2004.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems & Imaging*, 27(2):025010, 2011.

S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

Q. Gu, H. Gui, and J. Han. Robust tensor decomposition with gross corruption. In *Advances in Neural Information Processing Systems*, pages 1422–1430, 2014.

S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang. Weighted nuclear norm minimization and its applications to low level vision. *International Journal of Computer Vision*, 121(2):183–208, 2017.

H. Gui, J. Han, and Q. Gu. Towards faster rates and oracle property for low-rank matrix estimation. In *International Conference on Machine Learning*, pages 2300–2309, 2016.

X. Guo, Q. Yao, and J. Kwok. Efficient sparse low-rank tensor completion using the Frank-Wolfe algorithm. In *AAAI Conference on Artificial Intelligence*, pages 1948–1954, 2017.

X. Han, S. Cao, X. Lv, Y. Lin, Z. Liu, M. Sun, and J. Li. OpenKE: An open toolkit for knowledge embedding. In *Conference on Empirical Methods in Natural Language Processing*, pages 139–144, 2018.

W. Hare and C. Sagastizábal. Computing proximal points of nonconvex functions. *Mathematical Programming*, 116(1-2):221–258, 2009.

P. Hartman. On functions representable as a difference of convex functions. *Pacific Journal of Mathematics*, 9(3):707–713, 1959.

W. He, Q. Yao, C. C. Li, N. Yokoya, and Q. Zhao. Non-local meets global: An integrated paradigm for hyperspectral denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6861–6870, 2019.

C. J. Hillar and L.-H. Lim. Most tensor problems are NP-hard. *Journal of the ACM*, 60(6), 2013.

D. Hong, T. Kolda, and J. A. Duersch. Generalized canonical polyadic tensor decomposition. *SIAM Review*, 62(1):133–163, 2020.

M. Hong, Z.-Q. Luo, and Meisam R. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

C.-J. Hsieh, N. Natarajan, and I. Dhillon. PU learning for matrix completion. In *International Conference on Machine Learning*, pages 2445–2453, 2015.

Y. Hu, D. Zhang, J. Ye, X. Li, and X. He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9): 2117–2130, 2013.

H. Huang, Y. Liu, J. Liu, and C. Zhu. Provable tensor ring completion. *Signal Processing*, 171: 107486, 2020.

P. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, pages 73–101, 1964.

P. Indyk, A. Vakilian, and Y. Yuan. Learning-based low-rank approximations. In *Advances in Neural Information Processing Systems*, pages 7402–7412, 2019.

M. Janzamin, R. Ge, J. Kossaifi, and A. Anandkumar. Spectral learning on matrices and tensors. *Foundations and Trends in Machine Learning*, 2020.

W. Jiang, F. Nie, and H. Huang. Robust dictionary learning with capped ℓ_1-norm. In *International Joint Conference on Artificial Intelligence*, 2015.

H. Kasai and B. Mishra. Low-rank tensor completion: A Riemannian manifold preconditioning approach. In *International Conference on Machine Learning*, pages 1012–1021, 2016.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014.

T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

D. Kressner, M. Steinlechner, and B. Vandereycken. Low-rank tensor completion by Riemannian optimization. *BIT Numerical Mathematics*, 54(2):447–468, 2014.

T. Lacroix, N. Usunier, and G. Obozinski. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*, 2018.

H. A. Le Thi and P. D. Tao. The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. *Annals of Operations Research*, 133(1-4):23–46, 2005.

T. Lei, X. Wang, and H. Liu. Uncoverning groups via heterogeneous interaction analysis. In *IEEE International Conference on Data Mining*, pages 503–512, 2009.

A. S. Lewis and H. S. Sendov. Nonsmooth analysis of singular values. *Set-Valued Analysis*, 13(3): 243–264, 2005.

H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. In *Advances in Neural Information Processing Systems*, pages 379–387, 2015.

Q. Li, Y. Zhou, Y. Liang, and P. Varshney. Convergence analysis of proximal gradient with momentum for nonconvex optimization. In *International Conference on Machine Learning*, pages 2111–2119, 2017.

J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

P. Loh and M. Wainwright. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16:559–616, 2015.

C. Lu, J. Shi, and J. Jia. Online robust dictionary learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 415–422, 2013.

C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan. Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5249–5257, 2016a.

C. Lu, J. Tang, S. Yan, and Z. Lin. Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm. *IEEE Transactions on Image Processing*, 25(2):829–839, 2016b.

R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010.

R. Mazumder, D. F. Saldana, and H. Weng. Matrix completion with nonconvex regularization: Spectral operators and scalable algorithms. *Statistics and Computing*, 30:1113–1138, 2020.

G. A. Miller. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

C. Mu, B. Huang, J. Wright, and D. Goldfarb. Square deal: Lower bounds and improved relaxations for tensor recovery. In *International Conference on Machine Learning*, pages 73–81, 2014.

A. Narita, K. Hayashi, R. Tomioka, and H. Kashima. Tensor factorization using auxiliary information. *Data Mining and Knowledge Discovery*, 25(2):298–324, 2012.

S. Negahban and M. J. Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13(May):1665–1697, 2012.

S. Negahban, P. Ravikumar, M. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science & Business Media, 2013.

M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.

M. Nimishakavi, P. Jawanpuria, and B. Mishra. A dual framework for low-rank tensor completion. In *Advances in Neural Information Processing Systems*, 2018.

I. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

E. Papalexakis, C. Faloutsos, and N. Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Transactions on Intelligent Systems and Technology*, 8(2):16, 2017.

N. Parikh and S. P. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3): 123–231, 2013.

E. Phipps and T. Kolda. Software for sparse tensor decomposition on emerging computing architectures. *SIAM Journal on Scientific Computing*, 41:C269–C290, 2019.

H. Rauhut, R. Schneider, and Ž. Stojanac. Low rank tensor recovery via iterative hard thresholding. *Linear Algebra and its Applications*, 523:220–262, 2017.

S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *ACM International Conference on Web Search and Data Mining*, pages 81–90, 2010.

L. Shen, W. Liu, J. Huang, Y.-G. Jiang, and S. Ma. Adaptive proximal average approximation for composite convex minimization. In *AAAI Conference on Artificial Intelligence*, 2017.

M. Signoretto, R. Van de Plas, B. De Moor, and J. Suykens. Tensor versus matrix completion: A comparison with application to spectral data. *Signal Processing Letter*, 18(7):403–406, 2011.

Q. Song, H. Ge, J. Caverlee, and X. Hu. Tensor completion algorithms in big data analytics. *ACM Transactions on Knowledge Discovery from Data*, 2017.

N. Srebro, J. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1329–1336, 2005.

R. Tomioka and T. Suzuki. Convex tensor decomposition via structured schatten norm regularization. In *Advances in Neural Information Processing Systems*, pages 1331–1339, 2013.

R. Tomioka, K. Hayashi, and H. Kashima. Estimation of low-rank tensors via convex optimization. Technical report, arXiv preprint, 2010.

R. Tomioka, T. Suzuki, K. Hayashi, and H. Kashima. Statistical performance of convex tensor decomposition. In *Advances in Neural Information Processing Systems*, pages 972–980, 2011.

K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon. Representing text for joint embedding of text and knowledge bases. In *Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, 2015.

S. Tu, R. Boczar, M. Simchowitz, M. Soltanolkotabi, and B. Recht. Low-rank solutions of linear matrix equations via procrustes flow. In *International Conference on Machine Learning*, 2016.

L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3): 279–311, 1966.

Bart Vandereycken. Low-rank matrix completion by riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

L. Wang, X. Zhang, and Q. Gu. A unified computational and statistical framework for nonconvex low-rank matrix estimation. In *International Conference on Artificial Intelligence and Statistics*, 2017.

Y. Wang, W. Yin, and J. Zeng. Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1):29–63, 2019.

Y. Wang, Q. Yao, and J. Kwok. A scalable, adaptive and sound nonconvex regularizer for low-rank matrix learning. In *The Web Conference*, pages 1798–1808, 2021.

Z. Wang, Y. Zhou, Y. Liang, and G. Lan. Cubic regularization with momentum for nonconvex optimization. In *Uncertainty in Artificial Intelligence*, pages 313–322, 2020.

K. Wimalawarne, M. Yamada, and H. Mamitsuka. Convex coupled matrix and tensor completion. *Neural Computation*, 30:3095–3127, 2018.

Y. Xu, R. Hao, W. Yin, and Z. Su. Parallel matrix factorization for low-rank tensor completion. *Inverse Problems & Imaging*, 9(2):601–624, 2013.

S. Xue, W. Qiu, F. Liu, and X. Jin. Low-rank tensor completion by truncated nuclear norm regularization. *International Conference on Pattern Recognition*, pages 2600–2605, 2018.

Q. Yao and J. Kwok. Efficient learning with a family of nonconvex regularizers by redistributing nonconvexity. *Journal of Machine Learning Research*, 18(1):6574–6625, 2018.

Q. Yao, J. Kwok, F. Gao, W. Chen, and T.-Y. Liu. Efficient inexact proximal gradient algorithm for nonconvex problems. In *International Joint Conference on Artificial Intelligence*, pages 3308–3314, 2017.

Q. Yao, J. Kwok, and B. Han. Efficient nonconvex regularized tensor completion with structure-aware proximal iterations. In *International Conference on Machine Learning*, pages 7035–7044, 2019a.

Q. Yao, J. Kwok, T. Wang, and T.-Y. Liu. Large-scale low-rank matrix learning with nonconvex regularizers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019b.

Y.-L. Yu. Better approximation and faster algorithm using the proximal average. In *Advances in Neural Information Processing Systems*, pages 458–466, 2013.

Y.-L. Yu, Z. Xun, M. Micol, and E. Xing. Minimizing nonconvex non-separable functions. In *International Conference on Artificial Intelligence and Statistics*, pages 1107–1115, 2015.

L. Yuan, C. Li, D. Mandic, J. Cao, and Q. Zhao. Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion. *AAAI Conference on Artificial Intelligence*, 2019.

M. Yuan and C.-H. Zhang. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, 16(4):1031–1068, 2016.

C. H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38(2):894–942, 2010a.

T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11:1081–1107, 2010b.

Z. Zhang and S. Aeron. Exact tensor completion using t-SVD. *IEEE Transactions on Signal Processing*, 65(6):1511–1526, 2017.

Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki. Tensor ring decomposition. Technical report, arXiv, 2016.

Q. Zheng and J. Lafferty. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. In *Advances in Neural Information Processing Systems*, 2015.

X. Zheng, H. Ding, H. Mamitsuka, and S. Zhu. Collaborative matrix factorization with multiple similarities for predicting drug-target interactions. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

W. Zhong and J. Kwok. Gradient descent with proximal average for nonconvex and composite regularization. In *AAAI Conference on Artificial Intelligence*, pages 2206–2212, 2014.

Z. Zhu, Q. Li, G. Tang, and M. Wakin. Global optimality in low-rank matrix optimization. *IEEE Transactions on Signal Processing*, 66(13):3614–3628, 2018.