Robust Low-Rank Tensor Minimization via a New Tensor Spectral k-Support Norm

Jian Lou and Yiu-Ming Cheung, *Fellow, IEEE*

***Abstract*— Recently, based on a new tensor algebraic framework for third-order tensors, the tensor singular value decomposition (t-SVD) and its associated tubal rank definition have shed new light on low-rank tensor modeling. Its applications to robust image/video recovery and background modeling show promising performance due to its superior capability in modeling cross-channel/frame information. Under the t-SVD framework, we propose a new tensor norm called tensor spectral k-support norm (TSP-*k*) by an alternative convex relaxation. As an interpolation between the existing tensor nuclear norm (TNN) and tensor Frobenius norm (TFN), it is able to simultaneously drive minor singular values to zero to induce low-rankness, and to capture more global information for better preserving intrinsic structure. We provide the proximal operator and the polar operator for the TSP-*k* norm as key optimization blocks, along with two showcase optimization algorithms for medium- and large-size tensors. Experiments on synthetic, image and video datasets in medium and large sizes, all verify the superiority of the TSP-*k* norm and the effectiveness of both optimization methods in comparison with the existing counterparts.**

***Index Terms*— Robust low-rank tensor minimization, tensor robust principal component analysis, tensor singular value decomposition (t-SVD), alternating direction method of multipliers, proximal algorithm, conditional gradient descent.**

I. Introduction

MULTIDIMENSIONAL data, formally referred to as tensors, are high-order generalizations to vectors (i.e. first-order tensors) and matrices (i.e. second-order tensors). Tensor is a natural form of many real world data that appears in various areas ranging from image and video analysis in computer vision [1]–[4], social network analysis and recommendation system [5] in data mining [6], [7], to signal processing [8], [9], bioinformatics [10], and so on. One prominent example in computer vision and image processing is natural color images, which are 3−way tensors of size $n_1 \times n_2 \times 3$, where each of the three frontal slices corresponds to a color channel. In practice, the collected tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is often: 1) having exact or approximate intrinsic low-rank structure (denoted by a low-rank component $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$); 2) missing some entries due to unavailability or instrument failure; 3) contaminated with arbitrary corruption (denoted by a sparse component $\mathcal{E} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$). Robust low-rank tensor minimization (RLTM) is a popular tool for robustly recovering such complex multi-way data, which imposes the tensor low-rank structure by $\|\cdot\|_{tl}$ and the sparse structure by the well-known ℓ_1-norm $\|\cdot\|_1$, correspondingly. The following are two representative RLTM problems frequently used in image and video recovery tasks, where λ is a regularization parameter for balancing the low-rank and sparse terms.

Example 1 (Robust Tensor PCA):

$$\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_{tl} + \lambda\|\mathcal{E}\|_1, \quad s.t.\ \mathcal{X} - \mathcal{L} = \mathcal{E}. \tag{1}$$

Example 2 (Robust Low-Rank Tensor Completion):

$$\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_{tl} + \lambda\|\mathcal{E}\|_1, \quad s.t.\ \mathfrak{P}_\Omega(\mathcal{X} - \mathcal{L}) = \mathfrak{P}_\Omega(\mathcal{E}), \tag{2}$$

where Ω denotes the index set of observed tensor entries such that the projection $\mathfrak{P}_\Omega(\cdot)$ maps the entries in the observed positions to itself and maps the unobserved ones to zero.

In the literature, based on different tensor decomposition algebraic frameworks and their accompany rank definitions, there exist three lines for deriving the tensor low-rank regularization norm $\|\cdot\|_{tl}$, i.e. CANDECOMP/PARAFAC (CP) decomposition model [11], [12], Tucker decomposition model [13], and tensor singular value decomposition (t-SVD) model [14]. The CP model defines the rank to be the smallest number of rank one tensor decomposition, which then approximates a tensor as sum of rank-one outer products. The CP model has difficulty in determining the CP rank (known to be NP-hard problem). Also, its convex relaxation is ill-posed [15], [16]. The Tucker model unfolds a tensor to matrices along each mode (i.e. a single dimension) and defines the rank to be the matrix rank of each unfolded matrix. Many methods use sum of matrix nuclear norm (SNN) of each matricization to convexify the Tucker rank [3], [17]–[19]. A tensor is then folded back from the low-rank matrices. Albeit more favored than CP model in certain applications, it fails to exploit the correlations between modes. Also, the unfolding and folding processes tend to discard internal multidimensional structure information. In addition, each mode of matricization

Manuscript received April 21, 2018; revised November 24, 2018 and July 27, 2019; accepted September 27, 2019. Date of publication October 15, 2019; date of current version January 10, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61672444 and Grant 61272366, in part by Hong Kong Baptist University (HKBU), Research Committee, Initiation Grant, Faculty Niche Research Areas (IG-FNRA) 2018/19 under Grant RC-FNRA-IG/18-19/SCI/03, in part by the Innovation and Technology Fund of Innovation and Technology Commission of the Government of the Hong Kong SAR under Project ITS/339/18, in part by the Faculty Research Grant of HKBU under Project FRG2/17-18/082, and in part by the SZSTI under Grant JCYJ20160531194006833. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jean-Francois Aujol. *(Corresponding author: Yiu-Ming Cheung.)*

J. Lou is with the Department of Computer Science, Emory University, Atlanta, GA 30322 USA, and also with the Department of Computer Science, Hong Kong Baptist University, Hong Kong (e-mail: jian.lou@emory.edu).

Y.-M. Cheung is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong (e-mail: ymc@comp.hkbu.edu.hk).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TIP.2019.2946445

has the same number of entries with the original tensor, which leads to heavy computational burden for large size tensors.

A more promising approach which has received increasing interests is the recently proposed t-SVD model [14]. The t-SVD model decomposes a tensor $\mathcal{A}$ into a SVD-structure (i.e. $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$) similar to the matrix SVD, which is based on a new defined tensor-tensor product "$*$" (t-Product) [14]. The t-SVD naturally arises a new tensor tubal rank definition, which is the number of non-zero singular tubes of $\mathcal{S}$ [8], [14] and is also equivalent to the number of nonzero singular values of $\mathcal{A}$ [20]. By seeking low-rankness in terms of the tubal rank, t-SVD based methods expect to better capture the intrinsic structure of a tensor without much loss of correlation information as opposed to matricization of the Tucker model. By mimicking the relationship between matrix rank and matrix nuclear norm, most existing work utilize the tensor nuclear norm (TNN) as convex surrogate, which has achieved state-of-the-art performance in various computer vision and image processing tasks. For example, image and video completion (also called inpainting) [1], [8], [21]; robust image and video recovery [2]; outlier detection [4], [22]; moving object detection [23]. In terms of computation, TNN is equivalently defined as the sum of the matrix nuclear norm of each frontal slices after Fourier transformation, whose sizes are much smaller than matricization along modes. It reduces the computational cost with a certain degree when compared to the Tucker model.

The TNN relaxation shares similar rationale with ℓ_1-norm in the vector case and nuclear norm in the matrix case: seeking a convex relaxation on the unit max norm ball of the vector/singular vector (the max norm of a singular vector is also known as spectral norm). However, relaxing on the spectral norm ball can be less optimal. For example, in the vector cardinality case, papers [24], [25] show that seeking convex surrogate within unit ℓ_2 norm ball results into superior performance in sparse regression and feature selection tasks. In the matrix rank case, papers [26], [27] show that the convex relaxation of rank function within unit Frobenius norm ball is superior than the nuclear norm. Hence, we may ask: 1) Whether it is possible to derive an alternative (and better) convex surrogate to t-SVD ranks? 2) Whether the new norm allows convenient formulation that can be represented by matrices norms of the frontal slices in the Fourier domain? 3) Whether efficient optimization algorithms exist for the new norm regularized tensor minimization model?

In this paper, we focus on the new t-SVD framework and provide positive answers to each of the above three questions. For 1), we propose a new tensor norm, called tensor spectral k-support norm (TSP-k norm), which is derived by relaxing the t-SVD rank (sum of tubal multi-rank to be specific) within a *scaled tensor Frobenius norm* ball, rather than on the *scaled tensor spectral norm* ball as TNN does. For 2), we derive the closed-form formulation for the TSP-k norm, in terms of the matrix spectral k-support norm of the frontal slices of the tensor in the Fourier domain. For 3), we develop two key optimization components: the proximal operator and the polar operator for the TSP-k norm, which can be integrated into most proximal and conditional gradient algorithms to solve the TSP-k regularized problem, correspondingly. We then showcase the usage of the operators with the ADMM [28], [29] and universal primal dual algorithm [30], [31].

Our approach has several advantages as well as connections compared with TNN. First, we show that the TSP-k norm is an interpolation between the TNN and the tensor Frobenius norm. The tensor Frobenius norm factor of TSP-k norm contains additional global information, which can be helpful for better capturing the intrinsic structure among the entire tensor. Second, we derive the formulation for the TSP-k norm in terms of the singular values of Fourier transformed tensor. We find that rather than imposing sparsity penalties with ℓ_1 norm on all singular values, TSP-k only sums ℓ_1 norm over the minor singular values, which can avoid over penalizing large singular values that tends to leading to skewed estimation. The optimization algorithms also reveal new findings for TSP-k. For example, the polar operator based optimization for TSP-k amounts to decomposing the tensor into linear combinations of sum of tubal multi-rank k atom tensors. TNN exclusively decomposes with $k = 1$, which can lead to inferior estimation performance, as real tensors can have various intrinsic decomposition with $k > 1$. TSP-k provides such flexible choice of k.

In summary, our contributions are as follows:

(a) Section III proposes a new tensor spectral k-support norm for tensor tubal rank relaxation, wherein its advantages and connections with existing tensor norms under t-SVD framework are also discussed;
(b) Section IV-A and IV-B develop proximal and polar operators for the TSP-k, based on which an ADMM optimization for medium size data (Section V-A) and a universal primal dual optimization for large size data (Section V-B) are provided, correspondingly.
(c) Section VI conducts an extensive empirical study for the new norm and the algorithms with synthetic and real image/video datasets in both medium and large sizes.

II. Notation and Background

We summarize the frequently used notations in Table I, where the notations for scalar, vector, matrix, tensor and t-SVD, as well as their corresponding definitions/operations, can be found. As for the tensor norms, $\|\mathcal{A}\|_F$, $\|\mathcal{A}\|_1$, $\|\mathcal{A}\|_2$, $\|\mathcal{A}\|_\infty$ are called tensor Frobenius, ℓ_1, tensor spectral and tensor max norm, correspondingly. Their computations and their connections with Fourier transformed tensors can also be found in Table I.

We begin the introduction of the t-SVD algebraic framework with the following tensor-tensor product (t-product) definition:

Definition 3 (t-product [14]): The t-product between tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is defined as $\mathcal{A} * \mathcal{B} = \mathcal{C} \in \mathbb{R}^{n_1 \times n_4 \times n_3}$ with the (i, j)-th tube $\mathring{\mathbf{c}}_{ij}$ of $\mathcal{C}$ computed as

$$\mathring{\mathbf{c}}_{ij} = \mathcal{C}(i, j, :) = \sum_{k=1}^{n_2} \mathcal{A}(i, k, :) * \mathcal{B}(k, j, :), \tag{3}$$

where $*$ denotes the circular convolution between two tubes of same size.

[1] Matlab operation: $[\mathbf{a}^{\downarrow}, \texttt{idx}] = \texttt{sort}(\mathbf{a}, \text{'descend'})$, where $\mathbf{a}(\texttt{idx}) = \mathbf{a}^{\downarrow}$.

TABLE I
SUMMARY OF NOTATIONS IN THIS PAPER

scalar	a	$a \in \mathbb{R}$	$\|a\|$	absolute value
vector	$\mathbf{a}$	$\mathbf{a} \in \mathbb{R}^{n_1}$	$\mathbf{a}_i$	i-th element
	$\mathbf{1}_{n_1}$	all-1 vector	$\mathbf{0}_{n_1}$	all-0 vector
	$\|\mathbf{a}\|_\infty$	$\max_i \|\mathbf{a}_i\|$	$\mathbf{a}^\downarrow$	decreasingly sorted[1]
	$\|\mathbf{a}\|_1$	$\sum_{i=1}^{n_1} \|\mathbf{a}_i\|$	$\|\mathbf{a}\|_2$	$\sqrt{\sum_{i=1}^{n_1} \mathbf{a}_i^2}$
matrix	$\mathbf{A}$	$\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$	$\mathbf{A}_{ij}$	(i,j)-th element
	$\mathbf{I}$	identity matrix	$\texttt{svd}(\cdot)$	matrix SVD
	$\mathbf{0}_{n \times n}$	zero matrix	$\boldsymbol{\sigma}$	$\boldsymbol{\sigma} = \texttt{diag}(\mathbf{S})$
	$\|\mathbf{A}\|_\infty$	$\max_{i,j} \|\mathbf{A}_{ij}\|$	$\|\mathbf{A}\|_2$	$\|\boldsymbol{\sigma}\|_\infty$, $\|\boldsymbol{\sigma}_1\|$
	$\|\mathbf{A}\|_1$	$\sum_{ij} \|\mathbf{A}_{ij}\|$	$\|\mathbf{A}\|_F$	$\sqrt{\sum_{ij} \mathbf{A}_{ij}^2}, \|\boldsymbol{\sigma}\|_2$
tensor	$\mathcal{A}$	$\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$	$\mathcal{A}(:,i,:)$	i-th lateral
	$\mathcal{A}^{(i)}$	i-th frontal	$\mathcal{A}(:,:,i)$	i-th frontal
	$\mathcal{A}_{ijk}$	(i,j,k)-th element	$\mathcal{A}(i,:,:)$	i-th horizontal
	D	$\min\{n_1, n_2\} \cdot n_3$	$\|\mathcal{A}\|_F$	$\sqrt{\sum_{ijk} \mathcal{A}_{ijk}^2}$
	$\|\mathcal{A}\|_1$	$\sum_{ijk} \|\mathcal{A}_{ijk}\|$	$\|\mathcal{A}\|_\infty$	$\max_{ijk} \|\mathcal{A}_{ijk}\|$
t-SVD	$(\cdot) * (\cdot)$	t-product (Def. II.1)	$(\cdot)^\top$	conjugate transpose
	$\mathcal{A}_\mathcal{F}$	$\texttt{fft}(\mathcal{A}, [\,], 3)$	$\texttt{tsvd}(\cdot)$	t-SVD
	$\boldsymbol{\sigma}_{\mathbf{A}_\mathcal{F}^{(i)}}$	singular vector of $\mathbf{A}_\mathcal{F}^{(i)}$	$\boldsymbol{\sigma}_{\mathcal{A}_\mathcal{F}}$	$[\boldsymbol{\sigma}_{\mathbf{A}_\mathcal{F}^{(1)}}; ...; \boldsymbol{\sigma}_{\mathbf{A}_\mathcal{F}^{(n_3)}}]$
	$\mathbf{A}_\mathcal{F}$	$\texttt{blockdiag}(\mathcal{A})$ eq.(5)	$\mathbf{A}_\mathcal{F}^{(i)}$	$\mathbf{A}_\mathcal{F}^{(i)} := \mathcal{A}_\mathcal{F}^{(i)}$
	$\|\mathcal{A}\|_2$	$\|\mathbf{A}_\mathcal{F}\|_2$, $\|\boldsymbol{\sigma}_{\mathcal{A}_\mathcal{F}}\|_\infty$	$\|\mathcal{A}\|_F$	$\frac{1}{\sqrt{n_3}}\|\mathbf{A}_\mathcal{F}\|_F$, $\frac{1}{\sqrt{n_3}}\|\boldsymbol{\sigma}_{\mathcal{A}_\mathcal{F}}\|_2$

TABLE II
SUMMARY OF T-SVD RELATED DEFINITIONS

identity tensor	$\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$	$\mathcal{I}^{(1)} = \mathbf{I}$, $\mathcal{I}^{(i)} = \mathbf{0}_{\mathbf{n} \times \mathbf{n}}$, $i = 2, ...n_3$
orthogonal tensor	$\mathcal{Q} \in \mathbb{R}^{n \times n \times n_3}$	$\mathcal{Q}^\top * \mathcal{Q} = \mathcal{I}$
f-diagonal tensor	$\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$	diagonal matrices: $\mathcal{A}^{(i)}$, $i = 1...n_3$
tubal multi-rank	$\mathbf{r}(\mathcal{A})$	$(r_1, .., r_i, .., r_{n_3})$, $r_i = rank(\mathbf{A}_\mathcal{F}^{(i)})$
tubal rank	$rank_t(\mathcal{A})$	$\#\{i : \mathcal{S}(i,i,:) \neq \mathbf{0}\} = \max_i r_i$

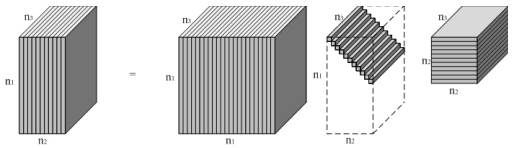

Fig. 1. Illustration of the t-SVD in Eq.(4) ([1], [14]). Tensors from left to right are $\mathcal{A}$, $\mathcal{U}$, $\mathcal{S}$ and $\mathcal{V}^\top$.

From Def. 3, the t-product can be seen as a generalization of the matrix product between $n_1 \times n_2$ and $n_2 \times n_4$ matrices by replacing the scalar to scalar multiplication (i.e. the $\cdot$ in $\mathbf{C}_{ij} = \sum_{k=1}^{n_2} \mathbf{A}(i,k) \cdot \mathbf{B}(k,j)$) with fiber to fiber circulant convolution (i.e. the $*$ in Eq.(3)).

The t-SVD definition is formalized in Def. 4 and Fig. 1 gives an illustration of the t-SVD on an $n_1 \times n_2 \times n_3$ tensor. The additional definitions of identity, orthogonal and f-diagonal tensors can be found in Table II, whose detailed definitions are in Appendix D of the supplement.

Definition 4 (t-SVD [14]): For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the t-SVD of $\mathcal{A}$ is given by

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top, \tag{4}$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$, $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors, and $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor.

Considering the equivalence between the t-product (essentially circulant convolution) in the original domain and the matrices multiplication in the Fourier domain, it is more convenient to carry out the t-SVD related computation in the Fourier domain, according to the well-known equivalence [14]. That is, for a third order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, let $\mathcal{A}_\mathcal{F}$ denote the Discrete Fourier transformation (DFT) of $\mathcal{A}$, which can be computed by Matlab command `fft` as $\mathcal{A}_\mathcal{F} = \texttt{fft}(\mathcal{A}, [\,], 3)$. The block diagonal matrix organized from $\mathcal{A}_\mathcal{F}$ is defined in Definition.5:

Algorithm 1 t-SVD: $(\mathcal{U}, \mathcal{S}, \mathcal{V}) = \texttt{tsvd}(\mathcal{A})$

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$
$\mathcal{A}_\mathcal{F} = \texttt{fft}(\mathcal{A}, [\,], 3)$;
for $i = 1, 2, ..., n_3$ **do**
$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \texttt{svd}(\mathbf{A}_\mathcal{F}^{(i)})$;
$\mathcal{U}_\mathcal{F}^{(i)} = \mathbf{U}$; $\mathcal{S}_\mathcal{F}^{(i)} = \mathbf{S}$; $\mathcal{V}_\mathcal{F}^{(i)} = \mathbf{V}$;
end for
$\mathcal{U} = \texttt{ifft}(\mathcal{U}_\mathcal{F}, [\,], 3)$; $\mathcal{S} = \texttt{ifft}(\mathcal{S}_\mathcal{F}, [\,], 3)$; $\mathcal{V} = \texttt{ifft}(\mathcal{V}_\mathcal{F}, [\,], 3)$;
Output: $\mathcal{U}, \mathcal{S}, \mathcal{V}$

Definition 5 (Block Diagonal Matrix [1]): Define the block diagonal operation by `blockdiag` and denote the computed block diagonal matrix by $\mathbf{A}_\mathcal{F}$, which are as follows,

$$\mathbf{A}_\mathcal{F} := \texttt{blockdiag}(\mathcal{A}_\mathcal{F}) := \begin{bmatrix} \mathbf{A}_\mathcal{F}^{(1)} & & & \\ & \mathbf{A}_\mathcal{F}^{(2)} & & \\ & & \ddots & \\ & & & \mathbf{A}_\mathcal{F}^{(n_3)} \end{bmatrix} \in \mathbb{C}^{n_1 n_3 \times n_2 n_3}, \tag{5}$$

where $\mathbf{A}_\mathcal{F}^{(i)} = \mathcal{A}_\mathcal{F}^{(i)}$, for $i = 1$ to n_3.

Algorithm 1 shows the algorithm for computing the t-SVD of $\mathcal{A}$, which is mainly based on the matrix SVD of $\mathbf{A}_\mathcal{F}^{(1)}$ to $\mathbf{A}_\mathcal{F}^{(n_3)}$ in eq.(5). By computing t-SVD and associated norms based on Fourier transformed $\mathbf{A}_\mathcal{F}$, the computation is more efficient and can be further parallelized since the matrix SVD of the n_3 Fourier transformed matrices $\mathbf{A}_\mathcal{F}^{(1)}$ to $\mathbf{A}_\mathcal{F}^{(n_3)}$ are independent. We denote the vector of singular values of $\mathbf{A}_\mathcal{F}^{(i)}$ as $\boldsymbol{\sigma}_{\mathbf{A}_\mathcal{F}^{(i)}}$ and let $\boldsymbol{\sigma}_{\mathcal{A}_\mathcal{F}} = [\boldsymbol{\sigma}_{\mathbf{A}_\mathcal{F}^{(1)}}; ...; \boldsymbol{\sigma}_{\mathbf{A}_\mathcal{F}^{(n_3)}}]$, which is the concatenation of all n_3 singular vectors. These definitions can also be found in Table I for quick reference.

The tensor nuclear norm (TNN) seeks a convex surrogate to the sum of the tensor tubal multi-rank (see Table II). There are two existing definitions of TNN, i.e. the one in [2], [32] and another in [1], [8], [33]. In this paper, we show both are special cases of the general α-tensor nuclear norm in Definition. 6, which can be derived by a unique convex relaxation of the $\frac{1}{\alpha}$ sum of the tubal multi-rank.

Definition 6 (General α-Tensor Nuclear Norm (α-TNN)): For a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the general α-tensor nuclear norm $\|\mathcal{A}\|_{t*,\alpha}$ is defined to be $\frac{1}{\alpha}$ of the sum of the matrix nuclear norm of all the frontal slices of $\mathcal{A}_\mathcal{F}$,

$$\|\mathcal{A}\|_{t*,\alpha} = \frac{1}{\alpha} \sum_{i=1}^{n_3} \|\mathbf{A}_\mathcal{F}^{(i)}\|_*. \tag{6}$$

For $\alpha = n_3$, $\|\mathcal{A}\|_{t*,n_3}$ takes the same form as [2], [32]; for $\alpha = 1$, $\|\mathcal{A}\|_{t*,1}$ becomes the one defined in [1], [8], [33]. For easy reference, we also introduce the following two concepts:

Definition 7: 1) α-scaled Tensor spectral norm ball constraint: $\{\mathcal{A} : \|\mathcal{A}\|_2 \leq \alpha\}$; 2) α-scaled Tensor Frobenius norm ball constraint: $\{\mathcal{A} : \|\mathcal{A}\|_F \leq \alpha\}$.

III. A New Convex Relaxation for Robust Tensor Recovery: Tensor Spectral k-Support Norm

In this section, we first derive the general α-TNN to unify n_3-TNN and 1-TNN. Then, we propose a new tensor norm under the t-SVD framework through a different convex relaxation. The new tensor norm contains n_3-TNN and tensor Frobenius norm as the special cases.

A. Derivation of the General α-TNN

We propose a new tensor norm by revisiting the relationship between the two existing different TNN notions: n_3-TNN in [2], [4], [32] and 1-TNN in [1], [8]. In [2], the authors observe that the n_3-TNN is the tightest convex relaxation of the *average of the tubal multi-rank* within the *unit spectral norm ball*, while the authors of [1], [8], [33] obtain the 1-TNN by relaxing the tubal multi-rank, where [1] shows that the 1-TNN is the tightest convex relaxation to ℓ_1 norm of the tensor multi-rank. Instead, the following Proposition 8 unifies both exiting TNN definitions by showing that it can also be viewed as the tightest convex relaxation of the *sum of the tubal multi-rank* within the *scaled tensor spectral norm ball*. The key to the unified relaxation is to relax based on $\mathcal{C}_k^{(sp)}$, which replaces the unit tensor spectral norm ball $\|\mathcal{A}\|_2 \leq 1$ with the more general α-scaled tensor spectral norm ball $\|\mathcal{A}\|_2 \leq \alpha$. The proof is in Appendix A1 of the supplementary material.

Proposition 8: Consider $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and the set

$$\mathcal{C}_k^{(sp)} = \{\mathcal{A} : \sum_{i=1}^{n_3} rank(\mathcal{A}_{\mathcal{F}}^{(i)}) \leq k,\ \|\mathcal{A}\|_2 \leq \alpha\}. \tag{7}$$

Then, the convex hull $conv(\cdot)$ of $\mathcal{C}_k^{(sp)}$ is given by

$$conv(\mathcal{C}_k^{(sp)}) = \{\mathcal{A} : \|\mathcal{A}\|_{t*,\alpha} \leq k,\ \|\mathcal{A}\|_2 \leq \alpha\}. \tag{8}$$

That is, the general α-TNN of $\|\mathcal{A}\|_{t*,\alpha}$ is the convex envelop of the sum of the tubal multi-rank within the α-scaled tensor spectral norm ball, which takes the form $\|\mathcal{A}\|_2 \leq \alpha$. In particular, substituting $\alpha = 1$ and $\alpha = \frac{1}{n_3}$ in, we have:

- *the 1-TNN is a special case of general α-TNN with $\alpha = 1$;*
- *the n_3-TNN is a special case of general α-TNN with $\alpha = n_3$.*

B. The New Tensor Spectral k-Support Norm

Instead of relaxing the sum of tubal multi-rank within the *α-scaled tensor spectral norm ball* $\mathcal{C}_k^{(sp)}$ to obtain TNN, this paper proposes a new tensor norm by seeking an alternative convex relaxation within an *α-scaled tensor Frobenius norm ball*:

$$\mathcal{C}_k^{(Fro)} = \{\mathcal{A} : \sum_{i=1}^{n_3} rank(\mathcal{A}_{\mathcal{F}}^{(i)}) \leq k,\ \|\mathcal{A}\|_F \leq \alpha\}. \tag{9}$$

The motivation is as follows. In relaxing TNN, the tensor spectral norm would mis-capture some global information because the tensor spectral norm only contains the information of the maximum singular value of all frontal slices. By contrast, our replacement with tensor Frobenius norm can provide more information across all frontal slices because it is computed based on all singular values. By relaxing on the Tensor Frobenius norm ball, we expect the TSP-k norm to induce the low-rankness with the better capture of global information. Figure 2 ([27]) illustrates the unit ball of the singular vector on $\mathbb{R}^3$. It shows that the singular values of the TSP-2 norm will contain both TNN and Tensor Frobenius norm factors, which implies that it is able to induce the low-rankness based on the TNN factor and capture more global correlation based on the Tensor Frobenius norm factor. In particular, with $\alpha = \sqrt{n_3}$, the following Definition defines the new tensor norm.

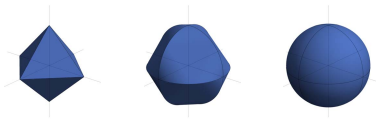

Fig. 2. Illustration of unit ball of singular values on $\mathbb{R}^3$ [27]. From left to right: TNN, TSP-2 norm and Tensor Frobenius norm.

Definition 9: The **Tensor Spectral k-Support norm** (TSP-k norm) $\|\cdot\|_{tsp,k}$ is defined to be the norm whose unit ball is the convex hull of set $\mathcal{C}_k^{(Fro)}$ (i.e. $conv(\mathcal{C}_k^{(Fro)})$).

In a similar form like Eq.(8), the TSP-k norm satisfies

$$conv(\mathcal{C}_k^{(Fro)}) = \{\mathcal{A} : \|\mathcal{A}\|_{tsp,k} \leq k,\ \|\mathcal{A}\|_F \leq \sqrt{n_3}\}. \tag{10}$$

That is, the TSP-k norm is the convex envelop of the sum of the tubal rank within the $\sqrt{n_3}$-scaled tensor Frobenius norm ball. By comparing eq.(8) and eq.(10), it can be seen that the TSP-k norm is the tightest convex relaxation on the $\sqrt{n_3}$ ball, while α-TNN is the tightest convex relaxation within the α-tensor spectral norm ball. As in TNN case, TSP-k norm can also be efficiently dealt with by taking FFT. The next Proposition details the relation between TSP-k norm of $\mathcal{A}$ with the vector k-support norm (denoted by $\|\cdot\|_{vp,k}$) of the singular values of $\mathcal{A}_{\mathcal{F}}$ and the matrix spectral k-support norm (denoted by $\|\cdot\|_{msp,k}$) of the block diagonal matrix $\mathbf{A}_{\mathcal{F}}$. The proof is in Appendix A2 of the supplementary material.

Proposition 10: For tensor $\mathcal{A}$, let $\mathcal{A}_{\mathcal{F}}$, $\mathbf{A}_{\mathcal{F}}$ and $\boldsymbol{\sigma}_{\mathcal{A}_{\mathcal{F}}}$ be FFT-transformed tensor, block diagonal matrix and the singular values of $\mathcal{A}_{\mathcal{F}}$, correspondingly. The TSP-k norm has the following relationships with the k-support norm of $\boldsymbol{\sigma}_{\mathcal{A}_{\mathcal{F}}}$ and the spectral k-support norm of $\mathbf{A}_{\mathcal{F}}$ as

$$\|\mathcal{A}\|_{tsp,k} = \frac{1}{n_3}\|\boldsymbol{\sigma}_{\mathcal{A}_{\mathcal{F}}}\|_{vp,k} = \frac{1}{n_3}\|\mathbf{A}_{\mathcal{F}}\|_{msp,k}. \tag{11}$$

Sort $\boldsymbol{\sigma}_{\mathcal{A}_{\mathcal{F}}}$ in non-increasing order and denote the result vector by $\boldsymbol{\sigma}^{\downarrow}_{\mathcal{A}_{\mathcal{F}}}$. According to Proposition 10 and the computation of the vector k-support norm [24] (please refer to Proposition 2.1 and its proof for more details), TSP-k norm has the following explicit computation as

$$\|\mathcal{A}\|_{tsp,k} = \frac{1}{n_3}\Big[\sum_{j=1}^{k-l-1}(\boldsymbol{\sigma}^{\downarrow}_{\mathcal{A}_{\mathcal{F}}})_j^2 + \frac{1}{l+1}\Big(\sum_{j=k-l}^{D}(\boldsymbol{\sigma}^{\downarrow}_{\mathcal{A}_{\mathcal{F}}})_j\Big)^2\Big]^{\frac{1}{2}}, \tag{12}$$

where l satisfies $(\boldsymbol{\sigma}^{\downarrow}_{\mathcal{A}_{\mathcal{F}}})_{k-l-1} > \frac{1}{l+1}\sum_{j=k-l}^{D}(\boldsymbol{\sigma}^{\downarrow}_{\mathcal{A}_{\mathcal{F}}})_j > (\boldsymbol{\sigma}^{\downarrow}_{\mathcal{A}_{\mathcal{F}}})_{k-l}$. By Eq.(12), the index l divides $\boldsymbol{\sigma}^{\downarrow}_{\mathcal{A}_{\mathcal{F}}}$ into larger part $(\boldsymbol{\sigma}^{\downarrow}_{\mathcal{A}_{\mathcal{F}}})_L = (\boldsymbol{\sigma}^{\downarrow}_{\mathcal{A}_{\mathcal{F}}})_{1:k-l-1}$ and smaller part $(\boldsymbol{\sigma}^{\downarrow}_{\mathcal{A}_{\mathcal{F}}})_S = (\boldsymbol{\sigma}^{\downarrow}_{\mathcal{A}_{\mathcal{F}}})_{k-l:D}$. The TSP-$k$ is a combination of the ℓ_2-norm of the larger part and the ℓ_1-norm of the smaller part, i.e.

$$\|\mathcal{A}\|_{tsp,k} = \frac{1}{n_3}(\|(\boldsymbol{\sigma}^{\downarrow}_{\mathcal{A}_{\mathcal{F}}})_L\|_2^2 + \frac{1}{l+1}\|(\boldsymbol{\sigma}^{\downarrow}_{\mathcal{A}_{\mathcal{F}}})_S\|_1^2)^{\frac{1}{2}}. \quad (13)$$

Hence, the TSP-k norm contains both the tensor nuclear norm factor and the tensor Frobenius factors. Formally, the following Proposition shows that the TSP-k norm interpolates between the tensor nuclear norm and the tensor Frobenius norm. The proof can be found in Appendix A3 of the supplementary material.

Proposition 11: The tensor spectral k-support norm becomes n_3-TNN when $k = 1$, while becomes tensor Frobenius norm when $k = D$.

By the preceding Proposition, both n_3-TNN and tensor Frobenius norm are special cases of TSP-k norm.

IV. Two Key Optimization Building Blocks for TSP-k Norm

A. Dual TSP-k Norm

Definition 12: The **Dual Tensor Spectral k-Support norm** (dual TSP-k norm) $\|\cdot\|^*_{tsp,k}$ is defined as

$$\|\mathcal{A}\|^*_{tsp,k} = \sup\{\langle\mathcal{A},\mathcal{B}\rangle \mid \mathcal{B} : \|\mathcal{B}\|_{tsp,k} \le 1\}. \quad (14)$$

The dual TSP-k norm can also be computed in terms of its singular values, which is given by the following Proposition:

Proposition 13: With the same notation as used in Proposition 10, the dual TSP-k norm of $\mathcal{A}$ can be computed as,

$$\|\mathcal{A}\|^*_{tsp,k} = \|\boldsymbol{\sigma}(\mathbf{A}_{\mathcal{F}})_{[1:k]}\|_2 = \|\boldsymbol{\sigma}(\mathbf{A}_{\mathcal{F}})\|^*_{vp,k} = \|\mathbf{A}_{\mathcal{F}}\|^*_{msp,k}, \quad (15)$$

where $\boldsymbol{\sigma}(\mathbf{A}_{\mathcal{F}})_{[1:k]}$ are the largest k singular values of $\mathcal{A}$ in the Fourier domain, $\|\cdot\|^*_{vp,k}$ and $\|\cdot\|^*_{msp,k}$ are the dual k-support norm and matrix k-spectral norm, correspondingly.

Remark 1: According to Proposition 13, the dual TSP-k norm is the ℓ_2 norm of the leading k singular values in the Fourier domain, which is much simpler in the formulation and easier in the computation than the primal TSP-k in eq.(12), which involves sorting and searching for the dividing index l. Intuitively, the dual TSP-k is more convenient to deal with.

Given the intuition above, we will develop two building blocks for the TSP-k norm regularized optimization: the proximal operator and polar operator, both of which make use of the dual TSP-k norm. It turns out that the dual norm is not only simpler, but also saves computational cost during optimization. First, the proximal operator of the primal TSP-k norm needs an exhaustive search step, while it uses the more efficient binary search step [25] for the dual norm. Second, the polar operator of the primal norm is exactly the dual norm, which suffices to calculate the leading k singular values to facilitate the adoption of Lanczos [34] and Power method [35] for partial SVD to reduce computational cost.

B. Proximal Operator for TSP-k Norm-Based Regularizer

The first building block is the proximal operator for the TSP-k norm, which can be incorporated with proximal algorithms. The proximal operator solves a second-order optimization subproblem and can be seen as a generalization to the projection operator. It is formally defined as follows:

Definition 14 (Proximal Operator [36]): The proximal operator of a closed proper convex function $r(\mathbf{x})$ at $\mathbf{v}$ is

$$\texttt{Prox}_r(\mathbf{v}) = \underset{\mathbf{x}}{\operatorname{argmin}}\, r(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|_2^2. \quad (16)$$

The proximal operator of the function r and its Fenchel conjugate r^* has the following useful relationship [36],

$$\mathbf{v} = \texttt{Prox}_r(\mathbf{v}) + \texttt{Prox}_{r^*}(\mathbf{v}). \quad (17)$$

For TSP-k norm-based regularizer $\frac{1}{2}\|\cdot\|^2_{tsp,k}$, the proximal operator has the following formulation,

$$\texttt{Prox}_{\frac{1}{2\beta}\|\cdot\|^2_{tsp,k}}(\mathcal{T}) = \underset{\mathcal{L}}{\operatorname{argmin}}\left[\frac{1}{2}\|\mathcal{L}-\mathcal{T}\|_F^2 + \frac{1}{2\beta}\|\mathcal{L}\|^2_{tsp,k}\right], \quad (18)$$

where β is a step-size related constant. Similar to the TNN case, the proximal operator for the TSP-k norm can also be converted from the tensor problem to the vector problem of its singular values in the Fourier domain. In the vector k-support norm case, the proximal operator for the primal norm is more difficult to compute than for the dual norm, because the former requires an exhaustive search sub-step on the singular values while the latter needs a binary search sub-step which is much more efficient. This fact affirms the intuition that the dual norm is more convenient to handle in the previous subsection. As a result, based on the conversion in Eq.(17), we calculate the proximal operator of the primal $\frac{1}{2}\|\cdot\|^2_{tsp,k}$ by instead computing the proximal operator of the dual $(\frac{1}{2}\|\cdot\|^*_{tsp,k})^2$ and converting back according to

$$\mathcal{T} = \texttt{Prox}_{\frac{1}{2\beta}\|\cdot\|^2_{tsp,k}}(\mathcal{T}) + \texttt{Prox}_{\frac{\beta}{2}(\|\cdot\|^*_{tsp,k})^2}(\mathcal{T}). \quad (19)$$

Definition 15: The proximal operator result of the dual TSP-k norm $\mathcal{L}^{\#}$is defined as

$$\mathcal{L}^{\#} = \texttt{Prox}_{\frac{\beta}{2}(\|\cdot\|^*_{tsp,k})^2}(\mathcal{T}). \quad (20)$$

In addition, $\mathcal{L}^{\#}_{\mathcal{F}}$ and $\boldsymbol{\sigma}_{\mathcal{L}^{\#}_{\mathcal{F}}}$ denote the Fourier transformed $\mathcal{L}^{\#}$ and its singular values in the Fourier domain, correspondingly.

The following Proposition 16 describes the computation of $\mathcal{L}^{\#}$, which first reduces the proximal subproblem from the tensor in the original domain to the vector of singular values in the Fourier domain and second applies the proximal operator of the dual k-support norm to the vector of singular values. The proof is in Appendix B1 of the supplementary material.

Proposition 16: Let $\mathcal{T}_{\mathcal{F}}$ be the FFT tensor of $\mathcal{T}$ and denote all singular values of $\mathcal{T}_{\mathcal{F}}$ by $\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}} = [\boldsymbol{\sigma}^{(1)}_{\mathcal{T}_{\mathcal{F}}}, ..., \boldsymbol{\sigma}^{(i)}_{\mathcal{T}_{\mathcal{F}}}, ..., \boldsymbol{\sigma}^{(n_3)}_{\mathcal{T}_{\mathcal{F}}}] \in \mathbb{R}^D$, where $\boldsymbol{\sigma}^{(i)}_{\mathcal{T}_{\mathcal{F}}} \in \mathbb{R}^{\min\{n_1,n_2\}}$

are the singular values of the i-th frontal slice $\mathcal{T}_{\mathcal{F}}^{(i)}$. Let $[\boldsymbol{\sigma}_{\mathcal{T}}^{\downarrow}, \mathtt{idx}] = \mathtt{sort}(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}, \text{'descend'})$. *Then,* $\boldsymbol{\sigma}_{\mathcal{L}_{\mathcal{F}}^{\#}}^{\downarrow}$ *is*

$$(\boldsymbol{\sigma}_{\mathcal{L}_{\mathcal{F}}^{\#}}^{\downarrow})_j = \begin{cases} \frac{\frac{1}{\beta n_3}\cdot(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})_j}{1+\frac{1}{\beta n_3}}, & j < k^{low} \\ \frac{\frac{1}{\beta n_3}\sum_{j=k^{low}}^{k^{upp}}(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})_j}{(1+\frac{1}{\beta n_3})(k-k^{low}+1)+\frac{1}{\beta n_3}(k^{upp}-k)}, & j \in \mathbb{I}_k^* \\ (\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})_j, & j > k^{upp}, \end{cases} \tag{21}$$

where $\mathbb{I}_k^* = [k^{low}, k^{upp}]$ *is the maximum subset containing k such that the following conditions hold*

$$\begin{cases} (\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})_{k^{low}} < \frac{(1+\frac{1}{\beta n_3})\sum_{j=k^{low}}^{k^{upp}}(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})_j}{(1+\frac{1}{\beta n_3})(k-k^{low}+1)+\frac{1}{\beta n_3}(k^{upp}-k)}, \\ (\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})_{k^{upp}} > \frac{\sum_{j=k^{low}}^{k^{upp}}(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})_j}{(1+\frac{1}{\beta n_3})(k-k^{low}+1)+\frac{1}{\beta n_3}(k^{upp}-k)}. \end{cases} \tag{22}$$

The maximality of $\mathbb{I}_k^*$ means that 1) including one more a larger $(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})_j$ (i.e. $(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})_{k^{low}-1}$) then the first inequality of eq.(22) cannot hold; 2) including one more smaller $(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})_j$ (i.e. $(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})_{k^{upp}+1}$) then the second inequality of Eq.(22) cannot hold. $\mathbb{I}_k^*$ can be obtained by two repeated binary search over $[1, k]$ and $[k, D]$ given the searching conditions Eq.(22). Based on Proposition 16 and Eq.(19), the proximal operator of $\frac{1}{2}\|\cdot\|_{tsp,k}^2$ is summarized in the following Corollary.

Corollary 17: The proximal operator of $\frac{1}{2}\|\cdot\|_{tsp,k}^2$ at $\mathcal{T}$ with constant β can be computed by

$$\mathtt{Prox}_{\frac{1}{2\beta}\|\cdot\|_{tsp,k}^2}(\mathcal{T}) = \mathcal{T} - \mathcal{L}^{\#}, \tag{23}$$

where $\mathcal{L}^{\#}$ is obtained by Proposition 16.

Algorithm 2 summarizes the proximal operator calculation. Its computational cost is given by the following Proposition and a step-by-step analysis is in Appendix E2:

Proposition 18: For input tensor of size $n_1 \times n_2 \times n_3$, Algorithm 2 has computational complexity $O(n_1 n_2 n_3 \min\{n_1, n_2\})$.

C. Polar Operator for TSP-k Norm-Based Regularizer

The second building block is the polar operator for the TSP-k norm, which can be incorporated with a type of "projection-free" algorithms, including the Frank-Wolfe algorithm (a.k.a. conditional gradient method) [37], [38] and generalized conditional gradient [31], [39], [40] which are variations for specific problems. The polar operator solves a linear subproblem as opposed to the second-order subproblem of the proximal operator. Compared to the proximal operator, it takes less computation but also leads to smaller per-iteration update. It is formally defined as follows:

Algorithm 2 Proximal Operator for TSP-k Norm

Input: $\mathcal{T}$ and β

$\mathcal{T}_{\mathcal{F}} = \mathtt{fft}(\mathcal{T}, [\,], 3)$;
for $i = 1, ..., n_3$ **do**
  $[\mathcal{U}_{\mathcal{F}}^{(i)}, \mathtt{diag}(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}^{(i)}}), \mathcal{V}_{\mathcal{F}}^{(i)}] = \mathtt{svd}(\mathcal{T}_{\mathcal{F}}^{(i)})$;
end for
$[\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow}, \mathtt{idx}] = \mathtt{sort}(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}, \text{'descend'})$;
$\mathtt{k}^{\mathtt{low}} = \mathtt{RepeatBinarySearch}(1, \mathtt{k}, \boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})$;
$\mathtt{k}^{\mathtt{upp}} = \mathtt{RepeatBinarySearch}(\mathtt{k}, \mathtt{D}, \boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})$;
Compute $\boldsymbol{\sigma}_{\mathcal{L}_{\mathcal{F}}^{\#}}^{\downarrow}$ according to Eq.(21);
$\boldsymbol{\sigma}_{\mathcal{L}_{\mathcal{F}}^{\#}}(\mathtt{idx}) = \boldsymbol{\sigma}_{\mathcal{L}_{\mathcal{F}}^{\#}}^{\downarrow}$;
for $i = 1, ..., n_3$ **do**
  $(\mathcal{L}_{\mathcal{F}}^{\#})^{(i)} = \mathcal{U}_{\mathcal{F}}^{(i)}\mathtt{diag}(\boldsymbol{\sigma}_{\mathcal{L}_{\mathcal{F}}^{\#}}^{(i)})(\mathcal{V}_{\mathcal{F}}^{(i)})^{\top}$;
end for
$\mathcal{L}^{\#} = \mathtt{ifft}(\mathcal{L}_{\mathcal{F}}^{\#}, [\,], 3)$;

Output: $\mathtt{Prox}_{\frac{1}{2\beta}\|\cdot\|_{tsp,k}^2}(\mathcal{T}) = \mathcal{T} - \mathcal{L}^{\#}$

Definition 19 (Polar Operator [39], [41]): For a norm $r(\mathbf{x})$, the polar operator at $\mathbf{v}$ is defined as

$$\mathtt{Polar}_r(\mathbf{v}) = \underset{\mathbf{x}}{\mathrm{argmax}}\{\langle \mathbf{x}, \mathbf{v}\rangle : \ r(\mathbf{x}) \le 1\}. \tag{24}$$

Definition 20: The polar operator result of the TSP-k norm $\mathcal{A}^{\#}$ is defined as

$$\mathcal{A}^{\#} = \underset{\mathcal{A}}{\mathrm{argmax}}\{\langle \mathcal{T}, \mathcal{A}\rangle : \ \|\mathcal{A}\|_{tsp,k} \le 1\}. \tag{25}$$

In addition, $\mathcal{A}_{\mathcal{F}}^{\#}$ and $\boldsymbol{\sigma}_{\mathcal{A}_{\mathcal{F}}^{\#}}$ denote the Fourier transformed $\mathcal{A}^{\#}$ and its singular values in the Fourier domain, correspondingly.

It is immediate to observe from eq.(25) and Definition 12 that the polar operator of the TSP-k norm is its dual norm, i.e., $\mathcal{A}^{\#} = \|\mathcal{T}\|_{tsp,k}^*$. The following Proposition gives the detailed computation for the polar operator. The proof is in Appendix B2 of the supplementary material.

Proposition 21: Let $\mathcal{T}_{\mathcal{F}} = \mathtt{fft}(\mathcal{T}, [\,], 3)$, $\mathcal{T}_{\mathcal{F}}^{(i)} = \mathcal{U}_{\mathcal{F}}^{(i)}\mathtt{diag}(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{(i)})(\mathcal{V}_{\mathcal{F}}^{(i)})^{\top}$, $\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}} = [\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{(1)}, ..., \boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{(n_3)}]$ *and* $[\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow}, \mathtt{idx}] = \mathtt{sort}(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}, \text{'descend'})$. *The polar operator result of eq.(25) satisfies,*

$$\mathcal{A}^{\#} = \mathtt{ifft}(\mathcal{A}_{\mathcal{F}}^{\#}, [\,], 3), \ (\mathcal{A}_{\mathcal{F}}^{\#})^{(i)} = \mathcal{U}_{\mathcal{F}}^{(i)}\mathtt{diag}(\boldsymbol{\sigma}_{\mathcal{A}_{\mathcal{F}}^{\#}}^{(i)})(\mathcal{V}_{\mathcal{F}}^{(i)})^{\top}, \tag{26}$$

where $\boldsymbol{\sigma}_{\mathcal{A}_{\mathcal{F}}^{\#}}(\mathtt{idx}) = \boldsymbol{\sigma}_{\mathcal{A}_{\mathcal{F}}^{\#}}^{\downarrow}$ *and*

$$(\boldsymbol{\sigma}_{\mathcal{A}_{\mathcal{F}}^{\#}}^{\downarrow})_j = \begin{cases} \frac{n_3(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})_j}{\|(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow})_{[1:k]}\|_2}, & j \in [1:k], \\ 0, & j \in [k:D]. \end{cases} \tag{27}$$

Algorithm 3 summarizes the polar operator computation. Its computational cost is given by the following Proposition and a step-by-step analysis is in Appendix E2 of the supplement:

Proposition 22: For input tensor of size $n_1 \times n_2 \times n_3$, Algorithm 3 has computational complexity $O(kn_1 n_2 n_3)$.

Eq.(27) indicates that $\mathbf{A}_{\mathcal{F}}^{\#}$ only depends on the largest k singular values among $\boldsymbol{\sigma}^{(1)}, ..., \boldsymbol{\sigma}^{(n_3)}$. Hence, it suffices to calculate the leading k singular values of each $\mathcal{X}_{\mathcal{F}}^{(i)}$, which

Algorithm 3 Polar Operator for TSP-k Norm

Input: $\mathcal{T}$;

$\mathcal{T}_{\mathcal{F}} = \texttt{fft}(\mathcal{T}, [\,], 3)$;
for $i = 1, ..., n_3$ **do**
  $[\mathbf{U}_{\mathcal{F}}^{(i)}, \texttt{diag}(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}^{(i)}}), \mathbf{V}_{\mathcal{F}}^{(i)}] = \texttt{partial\_svd}(\mathcal{T}_{\mathcal{F}}^{(i)}, k)$;
end for
$[\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{\downarrow}, \texttt{idx}] = \texttt{sort}(\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}, \text{'descend'})$ $\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}} = [\boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{(1)}, ..., \boldsymbol{\sigma}_{\mathcal{T}_{\mathcal{F}}}^{(n_3)}]$;
Compute $\boldsymbol{\sigma}_{\mathcal{A}_{\mathcal{F}}^{\#}}^{\downarrow}$ by Eq.(27);
Rearrange $\boldsymbol{\sigma}_{\mathcal{A}_{\mathcal{F}}^{\#}}^{\downarrow}$ back to corresponding frontal slices by $\boldsymbol{\sigma}_{\mathcal{A}_{\mathcal{F}}^{\#}}(\texttt{idx}) = \boldsymbol{\sigma}_{\mathcal{A}_{\mathcal{F}}^{\#}}^{\downarrow}$;
for $i = 1, ..., n_3$ **do**
  $(\mathcal{A}_{\mathcal{F}}^{\#})^{(i)} = \mathbf{U}_{\mathcal{F}}^{(i)} \texttt{diag}((\boldsymbol{\sigma}^{\#})^{(i)})(\mathbf{V}_{\mathcal{F}}^{(i)})^{\top}$;
end for
$\mathcal{A}^{\#} = \texttt{ifft}(\mathcal{A}_{\mathcal{F}}^{\#}, [\,], 3)$;

Output: $\texttt{Polar}(\mathcal{T}) = \mathcal{A}^{\#}$

TABLE III
COMPARISON OF THE TWO PROPOSED OPTIMIZATION ALGORITHMS

Properties	Algorithm 4	Algorithm 5
Key operator	Proximal operator (Alg.2)	Polar operator (Alg.3)
Fast convergence	✓	✗
Good scalability	✗	✓
Target problem size	Medium	Large

contain $\boldsymbol{\sigma}_j^{\#}$ for $j \in [1, k]$ for sure. That is, to compute $\mathbf{A}_{\mathcal{F}}^{\#}$, we only need to calculate partial svd, by Lanczos method [34] or Power method [35], which cost only $O(kn_1n_2n_3)$ per-iteration computation and is much smaller than $O(n_1n_2n_3 \min(n_1, n_2))$ of the proximal operator, since in practice k is much smaller than $\min(n_1, n_2)$.

V. TWO OPTIMIZATION ALGORITHMS FOR TSP-k REGULARIZED RLTM PROBLEM

In this section, we present two algorithms for solving the TSP-k regularized RLTM problem, which utilize the proximal operator (Subsection V-A Algorithm 4) and polar operator (Subsection V-B Algorithm 5) as their core building blocks, correspondingly. Due to the different operators they rely on, there is a trade-off between convergence rate and scalability. That is, the proximal operator-based method has larger per-iteration progress, thus it converges faster but has higher per-iteration complexity. On the contrary, the polar operator-based method has better scalability because of its lower per-iteration complexity, but converges slower due to the smaller per-iteration progress. As a result, they suit different applications. That is, the proximal operator-based method is suitable for medium size problems, where faster convergence rate can be guaranteed without worrying about the scalability issue. By contrast, the polar operator-based method suits larger scale problems, where scalability can be a computational bottleneck. Table III summarizes their comparisons.

A. Proximal Operator-Based Optimization Algorithm

1) Proximal Algorithm Choosing: With proximal operator developed in the preceding section, the TSP-k norm regularization can be integrated and optimized by a popular proximal algorithm which uses proximal operator as the core building block. The RLTM problem is a linear constrained two variable convex optimization problem:

$$\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x}) + g(\mathbf{y}), \quad s.t.\ \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{c}, \tag{28}$$

where f and g are TSP-k and ℓ_1 norm regularization functions. Among many proximal algorithms, we showcase the usage of the TSP-k norm proximal with the ADMM method (preconditioned ADMM in specific) in this subsection. The ADMM is a good candidate for the RLTM problem when compared to the other state-of-the-art splitting proximal algorithms: i) the primal-dual algorithm in [42] and primal-dual splitting algorithm [43] assume $\mathbf{c} = \mathbf{0}$ and one of the mappings $\mathbf{A}, \mathbf{B}$ is identity, which do not satisfied by RTLM, though [42] is equivalent to the preconditioned ADMM when both requirements are satisfied; ii)ADMM is a special ALM with the GaussSeidel decomposition, which allows the separated handling of f and g. In RTLM case, ADMM is more favored than ALM because the proximal operators for f and g are known and it is better to deal with them separately; iii) the Douglas-Rachford splitting algorithm [44] is more general and ADMM applied to eq.(28) is equivalent to the Douglas-Rachford splitting algorithm applied to the Fenchel dual of eq.(28). Since both methods eventually rely on the proximal operators of the TSP-k and the ℓ_1 norm, there is no much benefits worth of the additional conjugation transform. The ADMM applied to eq.(28) is simpler and more explicit. Finally, since most RTLM methods [45] are using the ADMM method, it is more convenient and fair to adopt the same ADMM algorithm skeleton for the TSP-k norm here when compared to the other low-rank norm-based methods.

2) Objective Function: We incorporate the TSP-k norm into RTLM problem:

$$\min_{\mathcal{L},\mathcal{E}} \frac{1}{2}\|\mathcal{L}\|_{tsp,k}^2 + \lambda\|\mathcal{E}\|_1, \quad s.t.\ \mathfrak{M}(\mathcal{X} - \mathcal{L}) = \mathcal{E}, \tag{29}$$

where $\mathfrak{M}$ can be as general as a linear tensor operator defined as $\mathfrak{M}(\mathcal{A}) = \mathcal{M} * \mathcal{A}$ ($*$ is the t-Product). Under this general modeling, Example 1 and 2 are special cases with $\mathfrak{M}$ being the identity mapping and the element-wise projection $\mathfrak{P}_{\Omega}(\cdot)$, correspondingly.

3) Algorithm Description: The ADMM based method mainly takes alternative updating for the $\mathcal{L}$ and $\mathcal{E}$ variables. In the following, ρ is a penalty parameter from the augmented Lagrangian formulation and η is a constant from linearization of the map $\mathfrak{M}$. We provide the update for $\mathcal{L}$ and $\mathcal{E}$ in the paper, while the detailed derivation can be found in Appendix E1 of the supplement. Algorithm 4 summarizes the procedure.

Update of $\mathcal{L}$:

$$\mathcal{L}_t = \texttt{Prox}_{\frac{1}{2\rho\eta}\|\cdot\|_{tsp,k}^2}\Big(\mathcal{L}_{t-1} + \frac{\mathcal{J}_{t-1}}{\rho\eta} + \frac{1}{\eta}\mathfrak{M}^{\top}(\mathfrak{M}(\mathcal{X} - \mathcal{L}_{t-1}) - \mathcal{E}_{t-1})\Big). \tag{30}$$

The computation of the proximal map in Eq.(30) has been given by Proposition 16 and detailed in Algorithm 2. In particular, to apply the general update to Examples 1 and 2, it suffices to set $\eta = 1$ and substitute $\mathfrak{M}$ by identity map and $\mathfrak{P}_{\Omega}$, correspondingly.

Algorithm 4 Preconditioned ADMM for (29)

Algorithm 4 Preconditioned ADMM for (29)

Input: Initialize $\mathcal{J}_0, \mathcal{L}_0, \mathcal{E}_0$ and set parameters T, ρ.

1: **for** $t = 0, 1, ..., T-1$ **do**

2: $\mathcal{L}_t = \texttt{Prox}_{\frac{1}{2\rho\eta}\|\cdot\|^2_{tsp,k}}\Big(\mathcal{L}_{t-1} + \frac{1}{\eta}\mathfrak{M}^\top(\mathfrak{M}(\mathcal{X} - \mathcal{L}_{t-1}) - \mathcal{E}_{t-1}) + \frac{\mathcal{J}_{t-1}}{\rho\eta}\Big)$ by Algorithm 2;

3: $\mathcal{E}_t = \texttt{Prox}_{\frac{\lambda}{\rho}\|\cdot\|_1}\Big(\mathfrak{M}(\mathcal{X} - \mathcal{L}_t) + \frac{\mathcal{J}_{t-1}}{\rho}\Big)$ by Eq.(32);

4: $\mathcal{J}_t = \mathcal{J}_{t-1} + \rho(\mathfrak{M}(\mathcal{X} - \mathcal{L}_t) - \mathcal{E}_t)$;

5: **end for**

Output: $\mathcal{L}_T, \mathcal{E}_T$

Update of $\mathcal{E}$*:*

$$\mathcal{E}_t = \texttt{Prox}_{\frac{\lambda}{\rho}\|\cdot\|_1}\Big(\mathfrak{M}(\mathcal{X} - \mathcal{L}_t) + \frac{\mathcal{J}_{t-1}}{\rho}\Big), \tag{31}$$

where the proximal operator of the ℓ_1 norm is well-known and can be efficiently computed by the element-wise soft-thresholding operation, i.e. with $\mathcal{T} = (\mathfrak{M}(\mathcal{X} - \mathcal{L}_t) + \frac{1}{\rho}\mathcal{J}_{t-1})$,

$$(\texttt{Prox}_{\frac{\lambda}{\rho}\|\cdot\|_1}(\mathcal{T}))_{ijk} = \texttt{sign}(\mathcal{T}_{ijk})\max\{|\mathcal{T}_{ijk}| - \frac{\lambda}{\rho}, 0\}. \tag{32}$$

4) Complexity and Convergence Analysis: The dominating per-iteration complexity comes from the proximal operator for the TSP-k norm in Step 2, whose computational cost is given by Proposition 18. The overall complexity is $O(n_1 n_2 n_3 \min\{n_1, n_2\})$. The high super-linear cost is attributed to the full SVD of the n_3 frontal slices in the Fourier domain, which are indispensable for computing the proximal map since all singular values are involved (see steps 6-8 of Algorithm 2).

Algorithm 4 is a two-block preconditioned ADMM algorithm applied to the linear constrained two variable convex optimization Problem (29). The convergence analysis of the ADMM-type algorithms is extensively studied. For example, paper [29], [42] are comprehensive and general references. For our specific Algorithm 4 with a linearized quadratic term in updating $\mathcal{L}$, it is an instance of the algorithm coming with guaranteed convergence in eq.(1.4) of paper [28]. The following Proposition is a paraphrase of Theorem 4.1 in [28]:

Proposition 23: Algorithm 4 is guaranteed to converge to the global optimum with ϵ precision in $O(\frac{1}{\epsilon})$ iterations.

B. Polar Operator-Based Optimization Algorithm

1) "Projection-Free" Algorithm Choosing: The polar operator-based methods are considered "projection-free" as opposed to proximal methods which use the "generalized" projection operator. We showcase the usage of the TSP-k polar operator with the recent universal primal dual method [30], [31]. Compared to the other polar operator-based methods, it provides the explicit handling of the linear constraint and explores the smoothness of the problem, which may partially resolve the slower convergence weakness. To this end, we show that the developed method is in essence greedy by adding one atom at one iteration. It not only provides a scalable optimization, but also shows that the low-rank tensor can be viewed as a linear combination of sum of tubal multi-rank k-tensors. From this perspective, TNN is a special case that builds a low-rank tensor exclusively by $k = 1$ combinations, which can be suboptimal for some applications, where the intrinsic low rank tensor is a $k > 1$ combination of atoms.

2) Objective Function: To utilize the polar operator, instead of the regularized form in eq.(29), we consider the ℓ_1-norm constrained form:

$$\underset{\mathcal{L},\mathcal{E}}{\text{argmin}}\ \frac{1}{2}\|\mathcal{L}\|^2_{tsp,k}, \quad s.t.\ \|\mathcal{E}\|_1 \le \tau \ \textit{and}\ \mathfrak{M}(\mathcal{X} - \mathcal{L}) = \mathcal{E}, \tag{33}$$

which is equivalent to the regularized form in eq.(29) with proper pair of τ and λ. The constraints amounts to $\|\mathfrak{M}(\mathcal{X} - \mathcal{L})\|_1 \le \tau$. By directly signifying the tolerance on the misfit, it is considered more natural than regularization formulation [46]. Also, for some applications where the misfit can be estimated, the constrained form eq.(33) can better utilize it as a priori.

Following the universal primal dual method [30], [31], we first convert eq.(33) to a dual TSP-k norm related equivalence by Fenchel conjugation as in the next Proposition. The proof is in Appendix C1 of the supplement.

Proposition 24: Let $\mathcal{J}$ denote the dual variable. The primal formulation in Eq.(33) has the following equivalent dual form,

$$\min_{\mathcal{J}} \mathfrak{D}(\mathcal{J}) = \min_{\mathcal{J}} \mathfrak{f}(\mathcal{J}) + \mathfrak{h}(\mathcal{J}), \tag{34}$$

$$\textit{where}\ \mathfrak{f}(\mathcal{J}) = \frac{1}{2}(\|-\mathfrak{M}^\top(\mathcal{J})\|^*_{tsp,k})^2 + \langle \mathcal{J}, \mathfrak{M}(\mathcal{X})\rangle, \tag{35}$$

$$\textit{and}\ \mathfrak{h}(\mathcal{J}) = \tau\|-\mathcal{J}\|_\infty. \tag{36}$$

In the following, we call $\mathfrak{f}$ as the dual loss function and $\mathfrak{h}$ as the dual regularizer. To solve the dual objective with gradient descent based methods, the next Proposition reveals a particular choice of the (sub)gradient of $\mathfrak{f}(\mathcal{J})$. The proof is in Appendix C2 of the supplement.

Proposition 25: The (sub)gradient of the dual loss function $\mathfrak{f}$ at $\mathcal{J}$, denoted by $\mathfrak{g}(\mathcal{J})$, can be computed as

$$\mathfrak{g}(\mathcal{J}) = -\mathfrak{M}(\mathcal{L}^\#) + \mathfrak{M}(\mathcal{X}), \tag{37}$$

$$\textit{where}\ \mathcal{L}^\# = (\|-\mathfrak{M}^\top(\mathcal{J})\|^*_{tsp,k}) \cdot \mathcal{A}^\#, \tag{38}$$

$$\textit{and}\ \mathcal{A}^\# = \underset{\|\mathcal{A}\|_{tsp,k}\le 1}{\text{argmax}}\ \langle -\mathfrak{M}^\top(\mathcal{J}), \mathcal{A}\rangle. \tag{39}$$

The core part for computing $\mathfrak{g}(\mathcal{J})$ is to compute $\mathcal{A}^\#$, which is the polar operator of TSP-k norm in Definition 20. Based on $\mathcal{A}^\#$, the computation for $\mathcal{L}^\#$ in eq.(39), referred as the atom of TSP-k hereafter in brief, is summarized in the following Corollary 26. The proof is in Appendix C3 of the supplement.

Corollary 26: Let $\mathcal{T} = -\mathfrak{M}^\top(\mathcal{J})$ and $\mathcal{T}_\mathcal{F}$, $\boldsymbol{\sigma}_{\mathcal{T}_\mathcal{F}}$, $\boldsymbol{\sigma}^\downarrow_{\mathcal{T}_\mathcal{F}}$, $\texttt{idx}$, $\mathcal{U}^{(i)}_\mathcal{F}$ and $\mathcal{V}^{(i)}_\mathcal{F}$ be the same notation as defined in Proposition 21. Then $\mathcal{L}^\# = \texttt{ifft}(\mathcal{L}^\#_\mathcal{F}, [\,], 3)$, where $\forall i \in [n_3]$,

$$(\mathcal{L}^\#_\mathcal{F})^{(i)} = \mathcal{U}^{(i)}_\mathcal{F}\texttt{diag}(\boldsymbol{\sigma}^{(i)}_{\mathcal{L}^\#_\mathcal{F}})(\mathcal{V}^{(i)}_\mathcal{F})^\top, \ \textit{and} \tag{40}$$

$$(\boldsymbol{\sigma}_{\mathcal{L}^\#_\mathcal{F}}(\texttt{idx}))_j = \begin{cases} n_3(\boldsymbol{\sigma}^\downarrow_{\mathcal{T}_\mathcal{F}})_j, & j \in [1:k], \\ 0, & j \in [k:D]. \end{cases} \tag{41}$$

3) Algorithm Description: Equipped with the (sub)gradient in the preceding subsections, we elaborate the universal primal-dual optimization [31] for the TSP-k regularized RTLM. The algorithm is summarized in Algorithm 5, where each iteration is composed by two ingredients:

Update of $\mathcal{J}$: An accelerated proximal gradient descent (APG) is used for updating the Lagrangian dual variable $\mathcal{J}_t$:

$$\mathcal{J}_{t+1} = \arg\min_{\mathcal{J}} \mathfrak{f}(\breve{\mathcal{J}}_t) + \langle \mathfrak{g}, \mathcal{J} - \breve{\mathcal{J}}_t \rangle + \frac{H_t}{2}\|\mathcal{J} - \breve{\mathcal{J}}_t\|_F^2 + \mathfrak{h}(\mathcal{J}); \tag{42}$$

$$\breve{\mathcal{J}}_{t+1} = \mathcal{J}_{t+1} + \frac{\theta_t - 1}{\theta_{t+1}}(\mathcal{J}_{t+1} - \mathcal{J}_t), \tag{43}$$

where θ_{t+1} is a constant sequence updated according to $\theta_{t+1} = \frac{1+\sqrt{1+4\theta_t^2}}{2}$ with initialization $\theta_1 = 1$, and $\breve{\mathcal{J}}$ is an acceleration sequence kept by the APG method. H_t can be a constant parameter such that the R.H.S. of Eq.(42) is an upper estimation of the dual objective, which is also the reciprocal of the step size. A better choice of H_t that enables the algorithm to adapt to both the degree ν and magnitude of the Hölder smoothness notion of the dual loss function is provided in Appendix C1 of the supplement. Eq.(42) is the proximal mapping of the dual regularizer of ℓ_1 norm, i.e. $\mathfrak{h}(\mathcal{J}) = \tau\|\mathcal{J}\|_\infty$, as

$$\mathcal{J}_{t+1} = \texttt{Prox}_{\frac{\tau}{H_t}\|\cdot\|_\infty}\big(\breve{\mathcal{J}}_t - \frac{1}{H_t}\mathfrak{g}(\breve{\mathcal{J}}_t)\big). \tag{44}$$

The proximal mapping of the dual norm can be performed by the projection on the unit ball of the original primal norm, as

$$\mathcal{J}_{t+1} = \big(\breve{\mathcal{J}}_t - (1/H_t)\mathfrak{g}(\breve{\mathcal{J}}_t)\big) - \frac{\tau}{H_t}\texttt{Proj}_{\|\cdot\|_1}\big(\frac{1}{\tau/H_t}(\breve{\mathcal{J}}_t - (1/H_t)\mathfrak{g}(\breve{\mathcal{J}}_t))\big). \tag{45}$$

Update of $\mathcal{L}$: Alongside the dual variable updating, a linear combination step is used for updating the primal variable $\mathcal{L}_t$:

$$\mathcal{L}_{t+1} = (1-\gamma^t)\mathcal{L}_t + \gamma^t\mathcal{L}_t^\#, \tag{46}$$

where the weighting sequence is $\gamma^t = \frac{\theta_t/H_t}{\sum_{j=1}^t \theta_j/H_j}$. The update of Eq.(46) is by nature greedy as it combines one atom to the low rank estimation at one time. Thus, we can observe from Eq.(46) that the low rank tensor induced by TSP-k norm is a linear combination of sum of tubal multi-rank k tensors. By Proposition 11, TNN is limited to $k = 1$ combination of atoms, which can be suboptimal for modeling tensors with intrinsic $k > 1$ combination of atoms.

The update of Eq.(46) is also closely related to the conditional gradient [39] (or called Frank-Wolfe method [38]). A key difference between Eq.(46) and the conditional gradient update is that the weight γ^t takes into consideration the smoothness H_t of the dual loss function when the linear-search subroutine in Algorithm 1 in Appendix E3 of the supplement is adopted.

4) Complexity and Convergence Analysis: One of the dominating per-iteration complexity comes from the polar operator in Step 2, whose computational cost is given by Proposition 22. Unlike proximal algorithms for TNN and TSP-k which take $O(n_1n_2n_3\min(n_1,n_2))$ for proximal operator, ours costs only $O(kn_1n_2n_3)$. Note that for some real problems, where the underlying tensor is of very low-rank, k here can be a very small constant. Considering another dominant computational costs of `fft` and `ifft` steps which are shared by all tensor SVD methods, our polar operator-based method effectively reduces the complexity from the super-linear complexity to nearly linear of $O((k+\log n_3)n_1n_2n_3)$.

Algorithm 5 Scalable Tensor Spectral k-Support Norm Regularized Robust Low-Rank Tensor Minimization

Input: $\mathcal{X}, \mathfrak{M}, \tau, H_0, T$;

1: **for** $t = 0, 1, ..., T-1$ **do**
2: Compute $\mathcal{L}_t^\#$ by Eq.(40) by calling $\mathcal{A}_t^\# = \texttt{Polar\_Operator}(\breve{\mathcal{J}}_t)$;
3: Compute $\mathfrak{g}(\breve{\mathcal{J}}_t)$ by Eq.(37);
4: `Backtracking_Line_Search` for H_t (**Optional**);
5: Update dual variable: $\mathcal{J}_{t+1} = \texttt{Prox}_{\frac{\tau}{H_t}\|\cdot\|_s^*}\big(\breve{\mathcal{J}}_t - \frac{1}{H_t}\mathfrak{g}(\breve{\mathcal{J}}_t)\big)$ according to Eq.(45);
6: Compute the primal recovery weight: $\gamma_t = \frac{\theta_t/H_t}{\sum_{j=1}^t \theta_j/H_j}$;
7: Update the acceleration sequence: $\theta_{t+1} = \frac{1+\sqrt{1+4\theta_t^2}}{2}$;
8: Update the interpolation sequence: $\breve{\mathcal{J}}_{t+1} = \mathcal{J}_{t+1} + \frac{\theta_t-1}{\theta_{t+1}}(\mathcal{J}_{t+1} - \mathcal{J}_t)$;
9: Update the primal variable: $\mathcal{L}_{t+1} = (1-\gamma_t)\mathcal{L}_t + \gamma_t\mathcal{L}_t^\#$;
10: **end for**

Output: $\mathcal{L}_T$

Algorithm 5 applies the accelerated universal primal-dual algorithm [31] to the constrained convex optimization Problem (33), which is guaranteed to converge to the global optimum. The following Proposition paraphrases Theorem 4.2 in [31]:

Proposition 27: Algorithm 5 is guaranteed to converge to the global optimum. For the primal variable $\mathcal{L}_T$ to achieve ϵ precision, Algorithm 5 takes the number of iterations by $O(\inf_{\nu\in[0,1]}(\frac{H_\nu}{\epsilon})^{\frac{2}{1+3\nu}})$ in the worst case.

In above, ν is the degree of the Hölder smoothness. For example, for smooth objective, $\nu = 1$ and the worst case iteration number is of order $O(\frac{1}{\epsilon^{1/2}})$, which is as fast as APG for regularized smooth primal loss problems.

VI. Experiment

A. Experiments on Medium Size Datasets

1) Synthetic Dataset: We first compare the proposed TSP-k norm with TNN norm on synthetic dataset. To generate the low-rank ground truth tensor $\mathcal{X}_{gt}$, we uniformly sample from [0, 1] an $n\times n\times n$ tensor and then truncate it to $Rank_{tubal}$ by t-SVD. We then add arbitrary corruption by randomly sampling 10% of the n^3 entries and set them to -20 or 20 with equal probability. For fairness of comparison, we use the same ADMM optimization algorithm for both methods. We also use the same stopping criterion for TNN and TSP-k norm, which is also used by [2], [20]: $\max\{\|\mathcal{L}_{t+1}-\mathcal{L}_t\|_\infty, \|\mathcal{E}_{t+1}-\mathcal{E}_t\|_\infty, \|\mathcal{L}_{t+1}+\mathcal{E}_t-\mathcal{X}\|_\infty\} \le tol$. In the paper, we have reported the results under $tol = 1e-5$. We choose $\alpha = \sqrt{n_3}$ for our TSP-k norm and choose $\alpha = 1$ for the TNN (i.e. we compare with n_3-TNN). We take two steps for this experiment: i) study the impact and choice for the balancing parameter λ; ii) the recovery performance after deciding the λ.

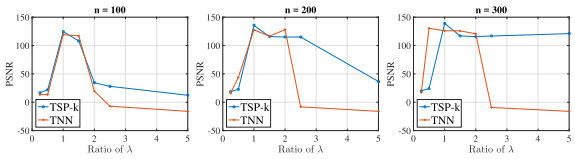

Fig. 3. Comparison of TNN and TSP-k under varying λ on synthetic data.

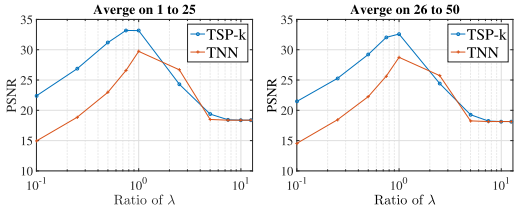

Fig. 4. Comparison of TNN and TSP-k under varying λ on BSD.

TABLE IV
RECOVERY RESULTS ON $n \times n \times n$ RANDOM DATA WITH DIFFERENT TUBAL RANKS

	$Rank_{tubal}(\mathcal{X}_{gt}) = 10$		$Rank_{tubal}(\mathcal{X}_{gt}) = 20$	
n	TNN	TSP-k	TNN	TSP-k
100	114.72 ± 0.09	135.79 ± 3.45	119.17 ± 0.15	124.58 ± 0.19
200	132.46 ± 0.03	141.18 ± 4.37	126.06 ± 4.77	133.52 ± 2.25
300	122.35 ± 0.02	142.76 ± 3.44	125.96 ± 0.02	139.17 ± 3.06

First, to study the effect of λ for TNN and TSP-k norms, we test a series of factors and run experiments on the random tensor of size $100 \times 100 \times 100$, $200 \times 200 \times 200$ and $300 \times 300 \times 300$ with tubal rank 20. We plot the PSNR versus the ratio of λ in Figure 3, which shows that the TSP-k outperforms the TNN in a wider range and the best achievable recovery performance is also better.

Second, fixing the λ at the best achievable PSNR, Table IV reports the recovery results with different n in terms of PSNR (relative to the ground truth), where the results are based on 20 times random realizations. As can be seen, TSP-k achieves higher PSNR with varying tensor sizes.

2) Image Denoising: We consider the image denoising task with the entire 200 images from the Berkeley Segmentation Database (BSD) [47]. The BSD dataset contains 200 color images of medium size (e.g. $321 \times 481 \times 3$) and the content spans a wide variety of natural scenes and objects. Please note that nature color images are often considered to be approximately low-rank, since their leading singular values of a small number dominate the main information and a large number of singular values are very close to zero. We generate random corruptions by randomly sampling 10% of the 3-way tensor entries and set them to random values in $[0, 255]$, which results up to 30% of the pixels to be randomly corrupted.

First, we study the effect of λ for TNN and TSP-k by plotting the PSNR versus a series λ. We conduct the experiment on 50 images randomly chosen from the 140 images whose k is set at 5 in the BSD dataset. Figure.4 reports the averaged PSNR on images 1 to 25, and 26 to 50. The TSP-k outperforms the TNN in a wider range and the best achievable PSNR is also higher.

Second, on the entire 200 images, we compare our method with tensor low-rank inducing norms TNN and SNN. Also, we compare it with the matrix low-rank inducing norm based methods: RPCA (matrix NN), PSSV [48], and CBD3M [49],

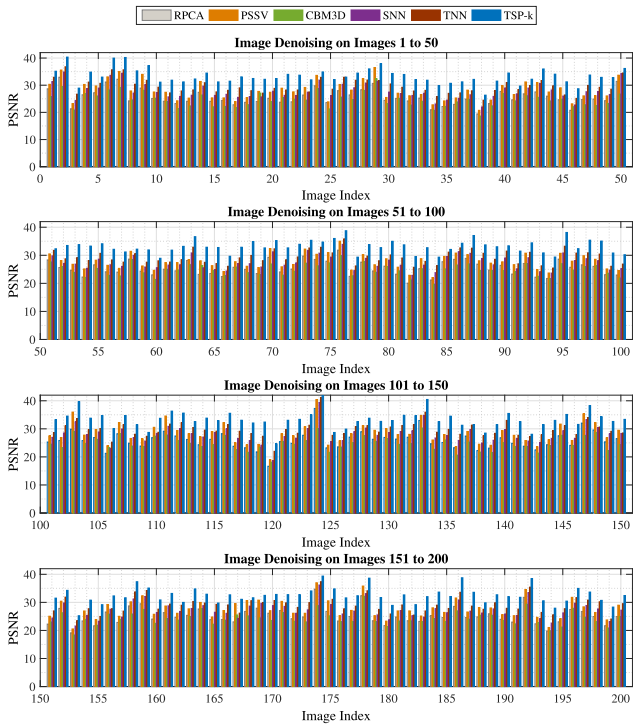

Fig. 5. Comparison of PSNR value for the image denoising experiment on all images of Berkeley Segmentation Database.

where PSSV is a nonconvex relaxation of the nuclear norm by discarding top singular values out of the nuclear norm, and CBD3M is a popular denoising method based on block-matching and collaborative filtering. For fair of comparison, we again use ADMM-type methods for optimizing all tensor and matrix-based norms. We set $tol = 10^{-5}$ for tensor based methods (we also test 10^{-7}, but the results barely change) and $tol = 10^{-7}$ for matrix based methods, which are default values chosen by the corresponding authors. As for λ, we follow [2] for the compared methods. For our TSP-k norm, we find that with the parameter pair $[k, \lambda] = [5, 20]$, TSP-k is superior than all compared low-rank inducing norms on 140 images of the entire 200 images. While for the remaining 60 cases, TSP-k achieves the best performance by slightly adjusting the choice to pairs like $[3, 20]$ (for 43 images) and $[7, 15]$ (for 13 images) and the remaining three also use $k = 5$ but set λ to 10 or 15. In terms of parameter setting, we can see that: 1) the λ parameter is not difficult to tune to get superior performance, since TSP-k norm performance much better than all compared norms on 183 of 200 images (over 90%) with a single $\lambda = 20$; 2) the setting of $k = 3, 5, 7$ indicates that natural images have different underlying linear combination of rank k sum of tubal multi-rank and $k = 1$ (choice of TNN) is not optimal.

The recovery performance is summarized in Fig. 5, which reports the PSNR values on all 200 images in BSD. Fig.5 exhibits recovered images on example images. Table V shows the PSNR values of the example images and the average PSNR value on the entire BSD dataset. From both the visual and quantitative experiment results, we summarize that 1) the t-SVD methods (TNN and TSP-k) are better than matrix-based norms (RPCA and PSSV), the block-matching and collaborative filtering based method CBM3D and Tucker model based

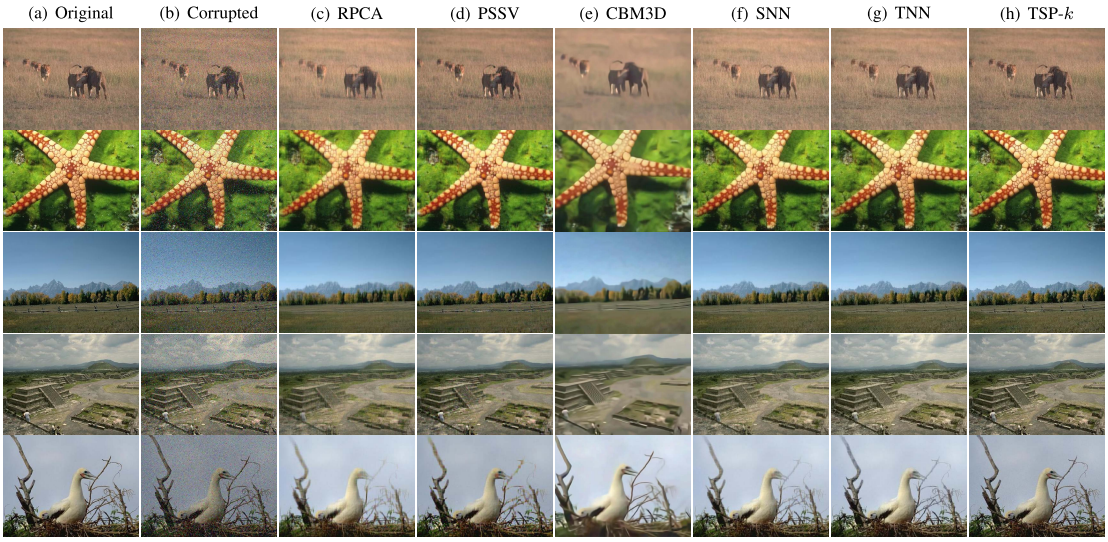

Fig. 6. Example images and recovery results of the image denoising experiment on Berkeley Segmentation Database. The indices of selected images from top to bottom are 006,020,087,144,193, where 020,087,144 and 193 are popularly demonstrated by previous related work.

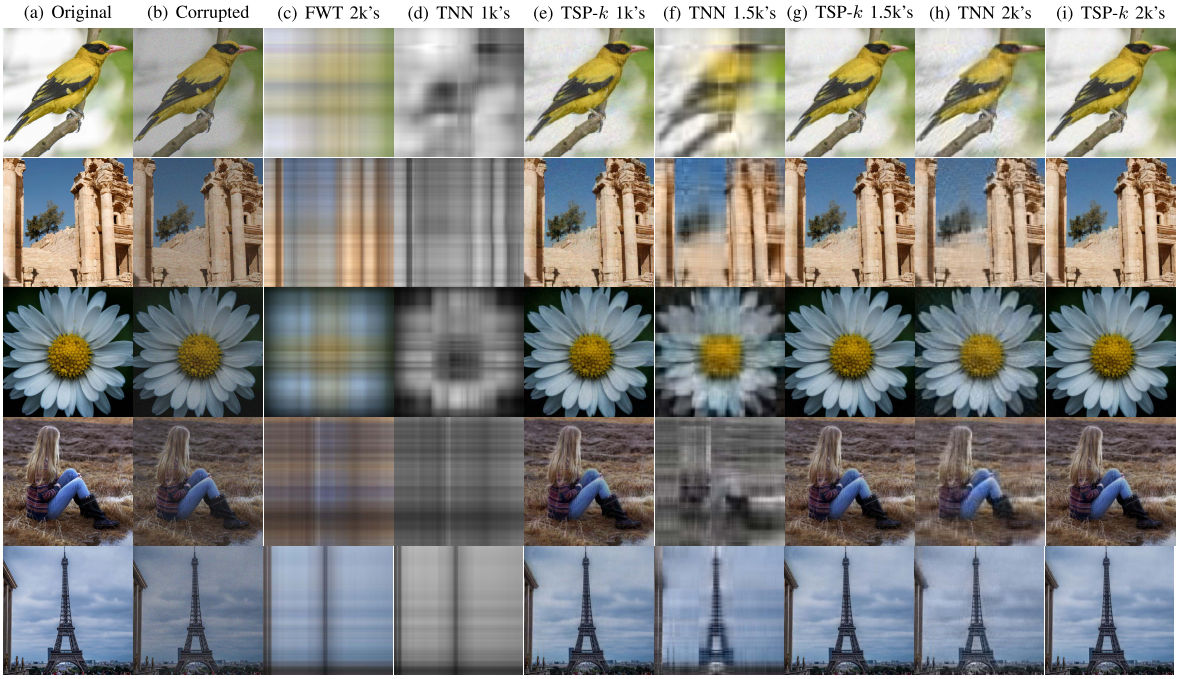

Fig. 7. Large size image recovery with 20% missing and 20% corruption. Recovery results of compared methods at CPU time 1000, 1500 and 2000 seconds.

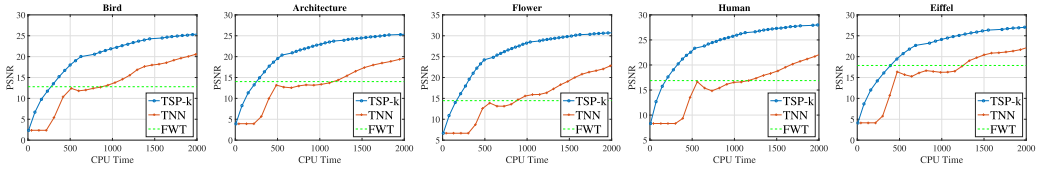

Fig. 8. Comparison of large size image recovery with 20% missing and 20% corruption: PSNR versus CPU time.

norm (SNN); 2) TSP-k norm is a better convex relaxation than TNN for the sum of tubal multi-rank; 3) to achieve the superior performance, the choice of parameters for TSP-k is not difficult.

3) Video Recovery From Observation With Simultaneous Corrupted and Missing Entries: We consider the video recovery task, where the video frames are corrupted by random noise and some values in the RGB channels are missing. We consider three video subsequences with different characteristics: 1) the Bike clip where the camera is moving to chase the moving bike, so that the background is gradually changing and a single object in the foreground is moving; 2) the Basketball clip where the camera chasing the players, so that the background is gradually changing and multiple objects in the foreground is moving; 3) the Walking clip where a steady surveillance camera captures a walking human, so that

TABLE V
COMPARISON OF PSNR VALUES ON EXAMPLE IMAGES FROM BSD

Image index	RPCA	PSSV	CBM3D	SNN	TNN	TSP-k
006	31.16	33.02	28.27	33.62	35.67	**39.88**
020	25.18	27.43	23.98	27.77	28.84	**32.52**
087	28.91	30.17	27.16	30.66	32.58	**37.25**
144	24.31	26.07	24.02	26.51	29.27	**32.91**
193	22.39	24.66	25.01	24.24	26.43	**30.56**
Avg on 200 images	25.53	27.92	24.68	27.63	29.46	**33.04**

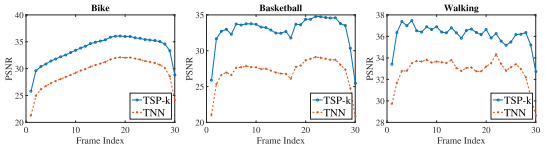

Fig. 9. Comparison of PSNR of each frame for the simultaneous video completion and denoising experiment.

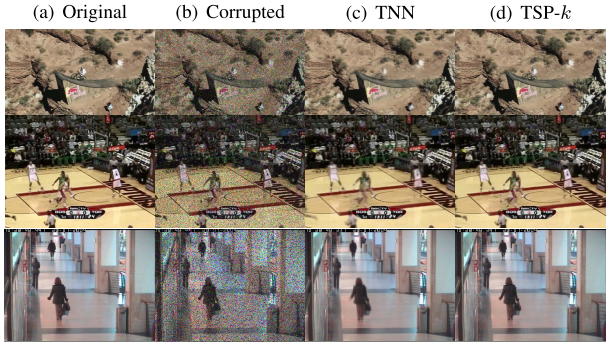

Fig. 10. Sample frames of simultaneous video completion and denoising experiment on videos: bike, basketball, and walking (from top to bottom).

the background is still and a main object in the foreground is moving. All three videos are of medium sizes with frame dimensions 640×360, 576×432 and 384×288. We randomly selects 30 consecutive frames for the experiment. With each frame being one frontal slice of the tensor, we reshape the video data into tensors of corresponding sizes $640\times1080\times30$, $576\times1296\times30$ and $384\times864\times30$, by vertically concatenating the RGB channel of each color frame into a frontal slice. We add 10% of all tensor values in $[0, 255]$ randomly as the corruption and set 20% tensor values as random missing. Fig. 9 presents the PSNR values of every frame in all video clips. Our method achieves better recovery results on all frames of the three video clips. Fig. 10 shows the sample frames and our method is also better by visual comparison.

B. Experiments on Larger Size Datasets

In this part, we consider the large size observation tensor setting as in Section V, where the existing proximal ADMM based t-SVD methods have scalability problem. Please note that there is no existing work has ever studied the t-SVD based low-rank tensor method under this larger scale robust low-rank tensor completion setting. We show that our TSP-k norm with the proposed universal primal dual method in Algorithm 5 is able to get better recovery result on large size color images and high resolution videos in much less CPU time.

1) Large Size Image Recovery From Observation With Simultaneous Missing Entries and Random Corruption: We utilize large size color images collected from Flickr as shown

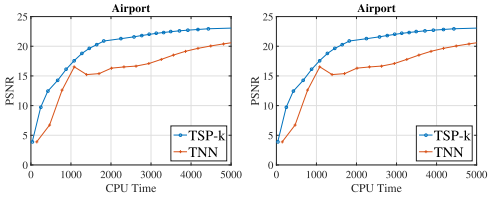

Fig. 11. Comparison of simultaneous large scale video completion and denoising experiment: PSNR versus CPU time.

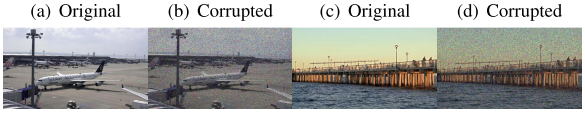

Fig. 12. Sample frames of simultaneous high resolution video (Airport and Bridge) completion and denoising experiment.

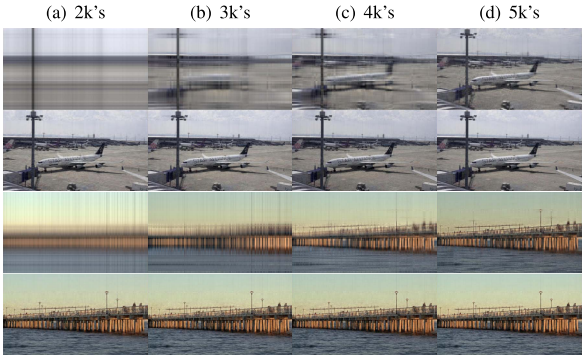

Fig. 13. Recovered frames, from top to bottom: airport by TNN, airport by TSP-k with Alg.5; bridge by TNN, bridge by TSP-k with Alg.5.

in the first column of Fig. 7, which have different content including: bird, ancient architecture, flower, human and Eiffel Tower. All images have frame sizes exceeding 3000×3000, which is much larger than the images in BSD. The second column of 7 shows the observation images, which are generated by setting 20% of the corresponding original image tensor entries with random corruption in $[0, 255]$ and randomly selecting 20% entries as missing. We compare with the ADMM-based TNN method as well as matrix NN method solved by a mixed Frank-Wolfe method called FWT in [40] which is a variant of conditional gradient and greedy in nature. As shown in the third column of Fig. 7, the matrix NN has unsatisfiable recovery performance visually even at CPU time 2000s. From the 4th to 9th column of Fig. 7, we present the recovery output of TNN and our TSP-k method at CPU time 1000s, 1500s and 2000s. Our method has better recovery result by using same CPU time. In addition, visually, our output at 1000s CPU time is better than the output of TNN at 2000s CPU time, which indicates our method can have better performance by using less than half of the time. Fig. 8 shows the quantitative comparison by plotting ℓ_1 and ℓ_2 error versus the CPU time, from which we can see that our method decreases the misfit much faster.

2) High Resolution Video Recovery From Observation With Simultaneous Missing Entries and Random Corruption: We consider video recovery from random corruption and missing entries on two 1440p resolution video clips with randomly selected 10 consecutive frames, which are of frame size 2560×1440 and of reshaped tensor size $2560\times4320\times10$. It is much

larger than the videos considered in the preceding section and presents great scalability issue for proximal ADMM based methods due to the full SVD per-iteration. The first is an airport video shot by hand held camera, where the whole image is slightly shaken and the plane in the center of the frame is pushing back. The second is a still camera capturing a bridge, where the water and people on the bridge are moving. Fig. 12 presents the sample frames of the original and corrupted and missing entries observation video. Fig. 13 shows the output sample frame recovered by proximal ADMM-based TNN and our greedy dual based TSP-k at 2000s,3000s,4000s and 5000s CPU time. Visually, our method has much better recovery performance at every reported time. Also, our output at 2000s is better than the output at 5000s returned by TNN. Fig. 11, by plotting ℓ_1 and ℓ_2 relative error against the ground-truth original video, also confirms that our method recovers low-rank video more efficiently and more accurately.

VII. Conclusion

In this paper, we have studied the robust low-rank tensor minimization problem under the t-SVD framework. We have derived an alternative relaxation to the sum of tubal multi-rank by providing a novel tensor spectral k-support norm. In particular, we have shown that TSP-k norm interpolates between TNN and TFN, which is helpful for preserving more global information of the intrinsic low-rank tensor. We provide two optimization methods for dealing with both medium and larger scale data, based on the proximal operator and the polar operator, correspondingly. Experiments on synthetic, image and video datasets with medium and larger-scale dimensions verified the superiority of TSP-k over TNN for low rank tensor modeling as well as the effectiveness of the two proposed optimization methods for their targeted data scales.

References

[1] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-SVD," in *Proc. CVPR*, Jun. 2014, pp. 3842–3849.

[2] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization," in *Proc. CVPR*, Jun. 2016, pp. 5249–5257.

[3] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

[4] P. Zhou and J. Feng, "Outlier-robust tensor PCA," in *Proc. CVPR*, Jul. 2017, pp. 3938–3946.

[5] N. Boumal and P.-A. Absil, "RTRMC: A Riemannian trust-region method for low-rank matrix completion," in *Proc. NIPS*, 2011, pp. 406–414.

[6] J. Sun, S. Papadimitriou, C.-Y. Lin, N. Cao, S. Liu, and W. Qian, "MultiVis: Content-based social network exploration through multi-way visual analysis," in *Proc. SIAM Int. Conf. Data Mining*. Philadelphia, PA, USA: SIAM, 2009, pp. 1064–1075.

[7] X. Li, M. K. Ng, and Y. Ye, "MultiComm: Finding community structure in multi-dimensional networks," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 4, pp. 929–941, Apr. 2014.

[8] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2015.

[9] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 1, pp. 225–253, 2014.

[10] L. Omberg, G. H. Golub, and O. Alter, "A tensor higher-order singular value decomposition for integrative analysis of DNA microarray data from different studies," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 47, pp. 18371–18376, 2007.

[11] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of 'Eckart–Young' decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[12] A. H. Kiers, "Towards a standardized notation and terminology in multiway analysis," *J. Chemometrics*, vol. 14, no. 3, pp. 105–122, 2000.

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

[14] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, 2013.

[15] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *J. ACM*, vol. 60, no. 6, p. 45, Nov. 2009.

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

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

[18] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. A. K. Suykens, "Learning with tensors: A framework based on convex optimization and spectral regularization," *Mach. Learn.*, vol. 94, no. 3, pp. 303–351, 2014.

[19] B. Romera-Paredes and M. Pontil, "A new convex relaxation for tensor completion," in *Proc. NIPS*, 2013, pp. 2967–2975.

[20] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," 2018, *arXiv:1804.03728*. [Online]. Available: https://arxiv.org/abs/1804.03728

[21] W. Hu, D. Tao, W. Zhang, Y. Xie, and Y. Yang, "The twist tensor nuclear norm for video completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 2961–2973, Dec. 2016.

[22] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, "Joint reconstruction and anomaly detection from compressive hyperspectral images using Mahalanobis distance-regularized tensor RPCA," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2919–2930, May 2018.

[23] W. Hu, Y. Yang, W. Zhang, and Y. Xie, "Moving object detection using tensor-based low-rank and saliently fused-sparse decomposition," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 724–737, Feb. 2017.

[24] A. Argyriou, R. Foygel, and N. Srebro, "Sparse prediction with the κ-support norm," in *Proc. NIPS*, 2012, pp. 1457–1465.

[25] A. Eriksson, T. T. Pham, T.-J. Chin, and I. Reid, "The κ-support norm and convex envelopes of cardinality and rank," in *Proc. CVPR*, Jun. 2015, pp. 3349–3357.

[26] A. M. McDonald, M. Pontil, and D. Stamos, "Spectral κ-support norm regularization," in *Proc. NIPS*, 2014, pp. 3644–3652.

[27] A. M. McDonald, M. Pontil, and D. Stamos, "New perspectives on κ-support and cluster norms," *J. Mach. Learn. Res.*, vol. 17, no. 155, pp. 1–38, 2016.

[28] B. He and X. Yuan, "On the $o(1/n)$ convergence rate of the Douglas–Rachford alternating direction method," *SIAM J. Numer. Anal.*, vol. 50, no. 2, pp. 700–709, 2012.

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

[30] Y. Nesterov, "Universal gradient methods for convex optimization problems," *Math. Program.*, vol. 152, nos. 1–2, pp. 381–404, 2015.

[31] A. Yurtsever, Q. T. Dinh, and V. Cevher, "A universal primal-dual convex optimization framework," in *Proc. NIPS*, 2015, pp. 3150–3158.

[32] J. Q. Jiang and M. K. Ng, "Exact tensor completion from sparsely corrupted observations via convex optimization," 2017, *arXiv:1708.00601*. [Online]. Available: https://arxiv.org/abs/1708.00601

[33] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller, "Tensor-based formulation and nuclear norm regularization for multienergy computed tomography," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1678–1693, Apr. 2014.

[34] R. M. Larsen, "Lanczos bidiagonalization with partial reorthogonalization," *DAIMI Rep. Ser.*, vol. 27, no. 537, p PB-537, 1998.

[35] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.

[36] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, Jan. 2014.

[37] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Res. Logistics Quart.*, vol. 3, nos. 1–2, pp. 95–110, 1956.

[38] M. Jaggi, "Revisiting Frank–Wolfe: Projection-free sparse convex optimization," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 427–435.

[39] Y. Yu, X. Zhang, and D. Schuurmans, "Generalized conditional gradient for sparse estimation," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 5279–5324, 2017.
[40] C. Mu, Y. Zhang, J. Wright, and D. Goldfarb, "Scalable robust matrix recovery: Frank–Wolfe meets proximal methods," *SIAM J. Sci. Comput.*, vol. 38, no. 5, pp. A3291–A3317, 2016.
[41] R. T. Rockafellar, *Convex Analysis*, vol. 28. Princeton, NJ, USA: Princeton Univ. Press, 1970.
[42] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vis.*, vol. 40, no. 1, pp. 120–145, 2011.
[43] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *J. Optim. Theory Appl.*, vol. 158, no. 2, pp. 460–479, 2013.
[44] P. L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM J. Numer. Anal.*, vol. 16, no. 6, pp. 964–979, 1979.
[45] C. Lu, J. Feng, S. Yan, and Z. Lin, "A unified alternating direction method of multipliers by majorization minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 527–541, Mar. 2018.
[46] A. Y. Aravkin, J. V. Burke, D. Drusvyatskiy, M. P. Friedlander, and S. Roy, "Level-set methods for convex optimization," 2016, *arXiv:1602.01506*. [Online]. Available: https://arxiv.org/abs/1602.01506
[47] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2. Jul. 2001, pp. 416–423.
[48] T.-H. Oh, Y.-W. Tai, J.-C. Bazin, H. Kim, and I. S. Kweon, "Partial sum minimization of singular values in robust PCA: Algorithm and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 744–758, Apr. 2016.
[49] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3d transform-domain collaborative filtering," in *Proc. 15th Eur. Signal Process. Conf.*, Sep. 2007, pp. 145–149.

Jian Lou received the Ph.D. degree from the Department of Computer Science, Hong Kong Baptist University. His research interests include statistical learning, numerical optimization, and privacy-preserving for machine learning.

Yiu-Ming Cheung (SM'06–F'18) received the Ph.D. degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.

He is currently a Full Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His current research interests include machine learning, pattern recognition, visual computing, and optimization.

Prof. Cheung is an IET Fellow, a BCS Fellow, an RSA Fellow, and a Distinguished Fellow of the IETI. He is the Founding Chair of the Computational Intelligence Chapter, IEEE Hong Kong Section, and the Chair of the Technical Committee on Intelligent Informatics, IEEE Computer Society. He serves as an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON CYBERNETICS, *Pattern Recognition*, *Knowledge and Information Systems*, and *Neurocomputing*.