Order-preserved Tensor Completion For Accurate Network-wide Monitoring

Xiaocan Li[1], Kun Xie[1], Xin Wang[2], Gaogang Xie[3], Kenli Li[1], Dafang Zhang[1], Jigang Wen[1]
[1] College of Computer Science and Electronics Engineering, Hunan University, Changsha, China
[2] Department of Electrical and Computer Engineering, State University of New York at Stony Brook, USA
[3] Computer Network Information Center, Chinese Academy of Sciences, Beijing, China

2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS) | 978-1-6654-6824-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/IWQoS54832.2022.9812910

***Abstract*—Network-wide monitoring is important for many network functions. However, monitoring data are often incomplete due to the need of sampling to reduce high measurement cost, system failure, and unavoidable transmission loss under severe communication. Instead of only targeting to estimate all missing monitoring data entries with a small set of measurement samples, we study a new order-preserved monitoring data estimation problem to accurately estimate the missing data entries while preserving the data entries' order in the dataset. We propose a novel order-preserved tensor completion model that integrates both the low rank property and the order information into a joint learning problem to estimate the missing data. With well designed non-convex function to directly approximate the tensor rank and order-preserved constraint under the linear self-recovery method, our model can not only more accurately capture the low-rank property of monitoring data to increase the estimation performance of missing data, but also can capture the order information in monitoring data to ensure the estimation accuracy. Extensive experiments using four real datasets demonstrate that compared with the state-of-the-art tensor completion algorithms, our proposed algorithm can provide more accurate estimation and keep the value order of recovered entries to more effectively retrieve top-k large entries.**

***Index Terms*—Order-preserved, Tensor Completion, Network-wide Monitoring**

I. Introduction

Network-wide monitoring is important for many important network management tasks, including anomaly detection, traffic engineering, and network troubleshooting. However, network management and data analysis usually suffer from the problem of data missing, due to the following reasons: 1) sampling-based measurement to reduce the cost; 2) failures of the monitoring systems or distributed monitoring nodes; 3) unavoidable transmission loss under severe communication, including network congestion and transmission over an unreliable transport protocol.

To provide network analyses with the complete monitoring data, an important step is to estimate the missing data entries from the partial measurement samples. For example, to track the SLA in a data center, studies [1], [2] estimate the network-wide latency data among all node pairs through the latency measurements among a few origin-destination pairs. To obtain the global traffic flow map for better traffic engineering, studies [3]–[6] estimate the network traffic matrix that records traffic volumes between every origin-destination (OD) pairs with a subset of traffic volume data between a small set of origin-destination (OD) pairs.

To estimate the missing monitoring data entries, various studies have been made to solve the data estimation problem, including purely spatial or temporal based algorithms [7], [8], matrix completion based algorithms [4], [5] and recent tensor completion based algorithms [6], [9]. Among which, tensor completion based algorithms are the most effective data estimation algorithms as these algorithms can combine multiple monitoring matrices together to take full advantage of the spatial and temporal correlations in the monitoring data to more accurately estimate the missing data.

In existing research studies [6], [9], tensor completion is applied to obtain the complete data, where missing entries in a tensor are estimated from a small set of measurement samples through tensor decomposition. In contrast, we formulate a new order-preserved tensor completion problem, which aims to not only accurately estimate the missing data but also preserve order of all data entries in the data set. That is, if two data entries x and x' are actually obtained through measurements and have the relationship $x > x'$, the estimated values p and p' should also guarantee that $p > p'$.

Order preservation is very important for network applications. With order preservation, we can return the top-k largest monitoring data entries more accurately. Depending on the type of KPI (Key Performance Indicator) recorded by the monitoring data, the top-k largest entries may correspond to elephant flows, the large network latency paths and the large packet loss paths in the network. The finding of top-k entries is especially important for advanced network applications such as traffic engineering, network anomaly detection, et.al. For example, the network trouble shooting is more concerned about the top-k largest latency and packet loss paths, the traffic engineering tasks are more concerned about the flows with top-k largest traffic volume.

Although tensor completion algorithms have been demonstrated to be more effective than the ones based on lower dimension data structures such as matrix and vector, current tensor completion based estimation algorithms usually adopt CP [6], [9] and Tucker decompositions [10], [11]. However, according to the recent tensor theory, these algorithms can not well characterize the intrinsic structure of a tensor [14]. Rather than using CP or tucker decomposition, a new tensor decomposition technique, called tensor Singular Value Decomposition (t-SVD) [12] is proposed, which can be considered as a more appropriate extension of the singular value decomposition

(SVD) from the matrix field to the tensor field to better characterize the intrinsic structure of a tensor.

In this paper, we would like to design our tensor completion algorithm based on t-SVD. Despite the strength of t-SVD, because of following drawbacks, existing tensor completion algorithm on t-SVD can not work well to preserve the order of data entries during the missing data estimation.

- **Convex relaxation to the low rank constraint in tensor completion compromises the recovery performance.** Due to the spatial-temporal correlations hidden in the monitoring data, network monitoring data have been proved to be low rank [6], which satisfies the requirement of the tensor completion. However, directly formulating the tensor completion problem using low rank function is NP-hard [13]. Most existing methods [10], [11], [14] use the nuclear norm as a convex approximation of the rank function to minimize. However, the nuclear norm simply adds all nonzero singular values together instead of treating them equally as the rank function does, which has a shrinkage effect that over-penalizes the large singular values and leads to a biased estimator.
- **Ignorance of order information impacts the recovery performance.** Current tensor completion algorithms estimate the sparse tensor solely based on a low rank function or its relaxation while ignoring the important intrinsic characteristic of the order information of entries, which further reduces the estimation accuracy and top-k largest data entries retrieving.

To address above issues, we propose a novel order-preserved tensor completion model that integrates both the low rank property and the order information into a joint learning problem for the estimation of missing data in network monitoring. Our model can not only more accurately capture the low-rank property of the monitoring data to increase the estimation performance of missing data, but also can capture the order information to keep the value sequence of the monitoring data. The later feature is particularly important for finding the top-k largest entries, and it also helps to improve the accuracy in missing data estimation. The following technique contributions are made in our paper:

- To provide a more accurate low rank constraint surrogate, we propose a novel nonconvex function to directly approximate the tensor rank. Compared with the nuclear norm, the penalties to different singular values in our constraint are more close to those with the low rank constraint, which can address the problems of shrinkage effect and biased estimator.
- To preserve the order information in monitoring data, we propose an order-preserved constraint under the linear self-recovery method, which can be easily integrated with the tensor completion model to enforce the learning process to capture both the low rank constraint and the order preservation among entries.
- We transform the complicate optimization problem into several convex subproblems, and also propose effective algorithms to solve the subproblems.
- We have performed extensive experiments using four network monitoring datasets including Abilene [15], GÈANT [16], Harvard226 [17], and WS-DREAM [18]. The results demonstrate the advantages of our order-preserved tensor completion model compared with 5 baselines. Compared with previous state of the art models, our proposed model can provide more accurate estimation and keep the order of top-k large entries.

The rest of the paper is organized as follows. We introduce the related work in Section II, and the preliminaries of t-SVD in Section III. We present the our basic problem in Section IV. We present the details of novel non-convex constraint surrogate and order-preserved constraint in Section V and Section VI, respectively. In section VII, we present our solution to the proposed order-preserved tensor completion. Finally, we evaluate our algorithm performance through extensive experiments in Section VIII and conclude the work in Section IX.

II. Related work

A. Tensor model based network monitoring data estimation

As the higher-order generalization of matrix, the tensor model has proven to be an effective data structure for dealing with the multi-dimensional data in a variety of fields [19]–[21]. It can take full advantage of the multilinear structures to extract richer information from the monitoring data.

Some recent studies propose tensor completion based algorithms to estimate the missing data with observed measurements. Study [22], for the first time, models the network traffic data as a tensor and estimate the unmeasured data from the partial measurement samples with its proposed sequential tensor completion algorithm. To achieve the quick missing monitoring data estimation, the study [6] proposes a GPU-accelerated parallel tensor completion algorithm to estimate the incomplete traffic tensor. By taking advantage of the stronger local correlation of data to form lower rank sub-tensors, the study [9] further proposes a localized tensor completion model (LTC) to increase the data recovery accuracy. To reduce the computational overhead, study [23] proposes a discrete tensor completion(DTC) model which uses binary codes to represent the factor matrices. Furthermore, with the help of binary code, this work can quickly estimate the top-k largest monitoring data entries.

Besides above studies, taking advantage of the low rank feature of the monitoring data, [24], [25] propose a tensor-based anomaly detection algorithms, which separates the monitoring data into two parts, to detect the traffic anomalies. Following these studies, several tensor model based anomaly detection algorithms [26]–[28] are proposed.

These studies demonstrate that compared with the vector and matrix based algorithm, tensor based algorithms are more effective to handle network monitoring data.

B. Tensor decomposition techniques

We notice that the tensor algorithms' ability relies on the underline tensor decomposition techniques, and two of the most well-known are CANDECOMP/PARAFAC (CP) style tensor factorization and Tucker decomposition (TKD). Above tensor-based network monitoring missing estimation and anomaly detection algorithms are all designed based on these two kinds of tensor decomposition.

CP decomposition (CPD) is a highly compact representation that factorizes a tensor into a sum of component rank-one tensor factors. The tensor rank in CP equals the smallest number of rank-1 tensors achieving the CP decomposition. To estimate the missing data based on CP, CP rank should be pre-identified. However, it is generally an NP-hard problem to identify the CP rank accurately [13] even though the whole tensor data is given. The tensor rank in Tucker-based algorithms is induced by Tucker decomposition, and is a multilinear rank formed by matrix rank. Although the rank is computable and equal to the rank of tensor's unfolding matrices, such a matricization technique fails to exploit the structure information embedded in a high dimensional tensor completely [29].

Different from the CP and Tucker decomposition, the algebraic framework of t-SVD induced by t-product [30] can be considered as a more appropriate extension of singular value decomposition (SVD) from the matrix field to the tensor field. Moreover, t-SVD provides an alternative tensor decomposition method without the drawbacks in finding the CP rank and Tucker rank [12], and can well characterize the intrinsic structure of a tensor. It is computable without the need of matricization technique. Thus, t-SVD is more promising to handle the monitoring data. Some recent studies [31], [32] propose an adaptive sampling algorithm to recover the unsampled data by t-SVD, which achieves a good recovery performance.

In light of the advantage of t-SVD over CP and Tucker decomposition, we design our tensor completion algorithm based on t-SVD. Moreover, different from the traditional problem that exploits the tensor completion to estimate all missing monitoring data entries with the small set of samples, we study a new order-preserved tensor completion problem which aims to accurately estimate the missing data entries while preserving the value sequence of all entries in the data set.

III. Preliminaries

In this section, we introduce some preliminaries. Scalars are denoted by lowercase letters x, vectors are denoted by boldface lowercase $\mathbf{x}$, matrices are represented by boldface capitals $\mathbf{X}$, the 3-order tensors are written as calligraphic letters $\mathcal{X}$, the (i,j,k)-th entry of tensor $\mathcal{X}$ is denoted by $x_{i,j,k}$, and the i-th row and j-column of matrix $\mathbf{X}$ are denoted by $\mathbf{x}_{i,:}$ and $\mathbf{x}_{:,j}$, respectively .

Definition 1: The **tubes** [30] are the higher-order analogue of matrix rows and columns, a tube is defined by fixing every index but one, i.e., (i,j)-th tube of tensor $\mathcal{X}$ is denoted as $\mathbf{x}_{i,j,:}$

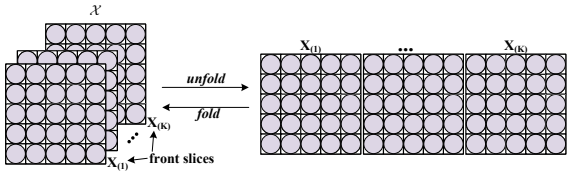

Fig. 1. The definition of **front slice**, $unfold$ and $fold$ operator.

Definition 2: The **slices** [30] are two-dimensional sections of a tensor, defined by fixing all but two indices.

In this paper, we only use the **front slices** as shown in Fig.1, which is written as $\mathbf{X}_{(:,:,k)}$. For simplicity, we also denote the front slices as $\mathbf{X}_{(k)}$.

Definition 3: We define the **unfold** and **fold** operator (as shown in Fig.1) of $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ as

$$\begin{aligned} &unfold\left(\mathcal{X}\right) = \left[\mathbf{X}_{(1)}, \mathbf{X}_{(2)}, \cdots, \mathbf{X}_{(K)}\right] \\ &fold\left(unfold\left(\mathcal{X}\right)\right) = \mathcal{X} \end{aligned} \tag{1}$$

Next, we give some basic definitions on t-SVD and outline the associated algebraic framework.

Definition 4: **Fourier transformation(DFT)** [30] on tensor $\mathcal{X}$ along the third dimension, we can use the MATLAB command fft, denoted as $\bar{\mathcal{X}} = fft\left(\mathcal{X}, [], 3\right)$. The **inverse DFT** is computed by command $ifft$, which is satisfied $\mathcal{X} = ifft\left(\bar{\mathcal{X}}, [], 3\right)$. The $\mathcal{X}$ is in original domain, and $\bar{\mathcal{X}}$ is in Fourier domain.

Definition 5: The **block diagonal matrix** [30] of tensor $\mathcal{X}$ denotes:

$$bdiag\left(\mathcal{X}\right) = \begin{bmatrix} \mathbf{X}_{(1)} & & & \\ & \mathbf{X}_{(2)} & & \\ & & \ddots & \\ & & & \mathbf{X}_{(K)} \end{bmatrix} \tag{2}$$

Definition 6: The **block circulant matrix** [30] of tensor $\mathcal{X}$ denotes:

$$bcirc\left(\mathcal{X}\right) = \begin{bmatrix} \mathbf{X}_{(1)} & \mathbf{X}_{(K)} & \cdots & \mathbf{X}_{(2)} \\ \mathbf{X}_{(2)} & \mathbf{X}_{(1)} & \cdots & \mathbf{X}_{(3)} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{X}_{(K)} & \mathbf{X}_{(K-1)} & \cdots & \mathbf{X}_{(1)} \end{bmatrix} \tag{3}$$

Definition 7: **T-Product** [30]: Let $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ and $\mathcal{Y} \in \mathbb{R}^{J\times l\times K}$. The t-product of these two tensors is defined as:

$$\mathcal{X} * \mathcal{Y} = fold\left(bcirc\left(\mathcal{X}\right) \cdot unfold\left(\mathcal{Y}\right)\right) \in \mathbb{R}^{I\times l\times K} \tag{4}$$

Definition 8: **Orthogonal Tensor** [30]: A tensor $\mathcal{Q} \in \mathbb{R}^{I\times I\times K}$ is orthogonal if it satisfies $\mathcal{Q}^T * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^T = \mathcal{I}$, where $\mathcal{Q}^T \in \mathbb{R}^{I\times I\times K}$ is obtained by transposing each frontal slice of $\mathcal{Q}$.

Definition 9: **F-Diagonal Tensor** [30]: A tensor is called f-diagnoal if each of its frontal slices is a diagnoal matrix.

IV. Problem

For a network consisting of N nodes, we use a 3-way tensor ($\mathcal{X} \in \mathbb{R}^{I\times J\times K}$) to record K days' network monitoring data of every end-to-end OD pairs where a day contains J time slots. We call the 3-way tensor as the monitoring tensor. In the monitoring tensor (as shown in Fig.2), I denotes the number of OD pairs with $I = N \times N$. If the tensor records the traffic volume data of the network, an entry x_{ijk} indicates the flow traffic volume data of OD pair i in the time slot j in the k-th day. If the tensor records the latency data of a network, the entry x_{ijk} indicates the latency between OD pair i in the time slot j of the k-th day.

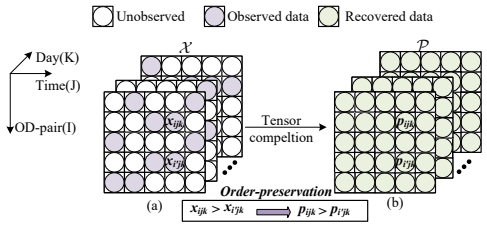

Fig. 2. The illustration of order-preserved tensor completion.

As introduced in Section I, the monitoring data are often incomplete (as shown in Fig.2). Different from the traditional network monitoring data estimation problem that targets to estimate all missing data entries with a small set of measurement samples, we study an **order-preserved tensor completion problem** to not only accurately estimate the missing data entries but also preserve the value sequence of all entries in the data set. That is, if the ground true values of two entries (i, j, k), (i', j, k) corresponding to the OD pairs i and i' in the same time slot j satisfy $x_{ijk} > x_{i'jk}$, the estimated values should also satisfy that $p_{ijk} > p_{i'jk}$.

Order preservation during the estimation of missing data is very important for network applications. With the order-preserved estimation of missing data, we can return the top-k largest monitoring data entries, which are especially important for advanced network applications such as congestion control, traffic engineering, network anomaly, et.al.

V. Accurate Tensor Completion With Nonconvex Surrogate

A. Traditional Tensor Completion Model and Its Drawback

Given an incomplete network monitoring tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$, low-rank tensor completion algorithms have been demonstrated to be feasible to estimate the unobserved entries given a small set of observed ones. If the set Ω of observed entries contains enough information, there is a unique low rank tensor $\mathcal{P}$ that is consistent with the observed entries can be estimated by solving the following rank minimization problem

$$\begin{aligned} &\min_{\mathcal{P}} rank\left(\mathcal{P}\right) \\ &subject\ to\ p_{i,j,k} = x_{i,j,k},\ (i,j,k) \in \Omega \end{aligned} \tag{5}$$

where $\mathcal{P}$ is the estimated tensor, the $rank\left(\mathcal{P}\right)$ denotes the rank of tensor $\mathcal{P}$.

However, solving this rank minimization problem in (5) is often impractical because it is NP-hard. Due to the inherent computational complexity of rank problems, the non-convex rank function is often relaxed to use the nuclear norm, and the problem (5) can be rewritten as:

$$\begin{aligned} &\min_{\mathcal{P}} \left\|\mathcal{P}\right\|_* \\ &subject\ to\ p_{i,j,k} = x_{i,j,k},\ (i,j,k) \in \Omega \end{aligned} \tag{6}$$

where $\left\|\mathcal{P}\right\|_*$ is the nuclear norm of tensor $\mathcal{P}$.

Although the convex relaxation is becoming a popular scheme and the problem can be solved efficiently, the use of convex relaxation is also associated with some drawbacks. Next, we present the drawbacks of using relaxation with nuclear norm.

Tensor completion relies on the technique of tensor decomposition. As introduced in Section 2, we use t-SVD as the basic tensor decomposition method in this paper. Before we give the definition of the $rank$ and nuclear norm of tensor $\mathcal{P}$, we first introduce t-SVD.

Definition 10: **t-SVD** decomposes the tensor $\mathcal{X}$ into three tensors $\mathcal{U}$, $\mathcal{S}$, $\mathcal{V}$, expressed as

$$\mathcal{X}{=}\mathcal{U} * \mathcal{S} * \mathcal{V}^T \tag{7}$$

where $\mathcal{U} \in \mathbb{R}^{I\times I\times K}$ and $\mathcal{V} \in \mathbb{R}^{J\times J\times K}$ are orthogonal tensors, $\mathcal{S} \in \mathbb{R}^{I\times J\times K}$ is a f-diagonal tensor.

Algorithm 1 t-SVD [30]

Input: Tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$.
Output: $\mathcal{U} \in \mathbb{R}^{I\times I\times K}$, $\mathcal{S} \in \mathbb{R}^{J\times J\times K}$, $\mathcal{V} \in \mathbb{R}^{I\times J\times K}$.
1: Compute $\bar{\mathcal{X}} = fft\left(\mathcal{X}, [\,], 3\right)$.
2: Compute each front slice of $\bar{\mathcal{U}}$, $\bar{\mathcal{S}}$ and $\bar{\mathcal{V}}$ by:
3: **for** $k = 1, \cdots, K$ **do**
4: $\quad\left[\bar{\mathbf{U}}_{(k)}, \bar{\mathbf{S}}_{(k)}, \bar{\mathbf{V}}_{(k)}\right] = SVD\left(\bar{\mathbf{X}}_{(k)}\right)$.
▷ SVD is the singular value decomposition
5: **end for**
6: Compute $\mathcal{U} = ifft\left(\bar{\mathcal{U}}, [\,], 3\right)$.
7: Compute $\mathcal{S} = ifft\left(\bar{\mathcal{S}}, [\,], 3\right)$.
8: Compute $\mathcal{V} = ifft\left(\bar{\mathcal{V}}, [\,], 3\right)$.

Fig.3 gives an illustration of the t-SVD. One efficient algorithm for computing the t-SVD is presented in Algorithm 1 [30]. To perform t-SVD on tensor $\mathcal{X}$, $\mathcal{X}$ should firstly be transformed in Fourier domain by DFT, and the transformed tensor is denoted by $\bar{\mathcal{X}}$. After applying the Singular Value Decomposition (SVD) on each front slice $\bar{\mathbf{X}}_{(k)}$ of the transformed tensor $\bar{\mathcal{X}}$, we obtain $\left[\bar{\mathbf{U}}_{(k)}, \bar{\mathbf{S}}_{(k)}, \bar{\mathbf{V}}_{(k)}\right] = SVD\left(\bar{\mathbf{X}}_{(k)}\right)$ for the front slices of $\bar{\mathcal{U}}$, $\bar{\mathcal{S}}$, $\bar{\mathcal{V}}$, respectively. Finally, by transforming $\bar{\mathcal{U}}$, $\bar{\mathcal{S}}$, $\bar{\mathcal{V}}$ to the original domain through inverse DFT, we obtain the resulted $\mathcal{U}$, $\mathcal{S}$, $\mathcal{V}$.

We define a singular value matrix $\sigma\left(\mathcal{X}\right) \in \mathbb{R}^{m\times K}$ with $\sigma_{lk} = \bar{\mathcal{S}}\left(l, l, k\right)$ and $m = min\left(I, J\right)$. Thus, we have $\sigma_{:k} = diag\left(\bar{\mathcal{S}}\left(:, :, k\right)\right)$, where $1 \le k \le K$.

Definition 11: **tensor tubal rank** [12] Based on t-SVD, the tensor tubal rank can be denoted as $rank_t\left(\mathcal{X}\right)$, which is

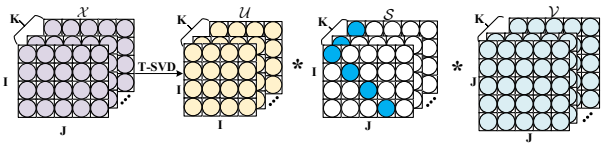

Fig. 3. The illustration of t-SVD algorithm.

defined as the maximum number of nonzero singular values of $\|\sigma_{:k}\|$, that is

$$rank_t(\mathcal{X}) = \max\left(\|\sigma_{:1}\|_0, \cdots, \|\sigma_{:K}\|_0\right) \tag{8}$$

where $\|\sigma_{:k}\|$ is actually the rank of k-th front slice $\mathbf{X}_{(k)}$, and $1 \le k \le K$. In this way, $rank_t(\mathcal{P})$ is the maximum rank of all front slices.

Definition 12: **tensor nuclear norm** Based on t-SVD, the nuclear norm can be expressed as

$$\|\mathcal{X}\|_* = \frac{1}{K}\sum_{l=1}^{m}\sum_{k}^{K}\sigma_{lk} = \frac{1}{K}\sum_{k=1}^{K}\|\sigma_{:k}\|_1 \tag{9}$$

where $\|\cdot\|_1$ is L_1 norm, which calculates the sum of absolute of the entries in $\sigma_{:k}$.

In the optimization problem, the loss is the penalty for failing to achieve a desired objective. Problem (5) uses tensor rank constraint as the objective function. For any $\sigma_{lk} > 0$, it will contribute 1 to the loss value $\|\sigma_{:k}\|_0$, thus at most contribute 1 to the objective function value $rank(\mathcal{P})$ in (5). Problem (6) uses the nuclear norm constraint as the objective function. For any $\sigma_{lk} > 0$, it will contribute the loss value $\frac{|\sigma_{lk}|}{K}$ to the objective function value $\|\mathcal{P}\|_*$ in (6). When minimizing the objective function, we call 1 and $\frac{|\sigma_{lk}|}{K}$ the penalties of σ_{lk} under the two problems (5) and (6), respectively.

We vary the value of σ_{lk} and draw its penalty under the two problems (5) and (6). With the increase of singular value, the singular value's penalty in problem (5) is equal to 1 constantly, while the singular value's penalty in problem (6) increases. Compared with minimizing the objective function in (5) with tensor rank, minimizing the objective function in the problem (6) with the nuclear norm always over-penalizes the large singular values, which results in the bias in the estimation of the tensor $\mathcal{P}$.

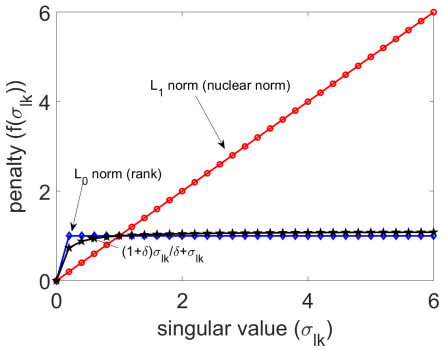

Fig. 4. The different penalties with respect to a varying singular value.

B. Tensor Completion with Nonconvex Surrogate

Although tensor completion based on the nuclear norm has been applied in many different fields, as discussed above, given any singular value larger than K, its penalty (loss) to nuclear norm is $\frac{|\sigma_{lk}|}{K}$ which is larger than 1, while its penalty to $rank$ function is 1. With the increase of singular value, its penalty to nuclear norm increases, which may bring the bias in the estimation because it over-penalizes the large singular value. As a result, a solution based on nuclear norm is a loose approximation of the $rank$ and not a good surrogate to the tensor rank.

In order to have a tighter approximation of the tensor rank, we define a nonconvex penalty function f of the singular values σ_{lk} as follows:

$$f(\sigma_{lk}) = \frac{(1+\delta)\sigma_{lk}}{\delta + \sigma_{lk}}, \qquad \delta > 0 \tag{10}$$

We can find that, $\lim_{\delta\to 0} f(\sigma_{lk}) = \|\sigma_{lk}\|_0$ and $\lim_{\delta\to\infty} f(\sigma_{lk}) = \|\sigma_{lk}\|_1$. In Fig.4, we also plot the penalty under (10) with respect to a varying singular value. As we can see, the black line ($f(\sigma_{lk})$) is very close to the blue line ($rank$ constraint) with the $\delta = 0.001$, while the red line (nuclear norm constraint) deviates from the blue line when the singular value is large. Hence, the proposed non-convex function f can address the over-penalization of larger singular values. Based on this function, we define a novel tensor completion problem as:

$$\begin{aligned} &\min_{\mathcal{P}} \|\mathcal{P}\|_\delta \\ &subject\ to\ p_{i,j,k} = x_{i,j,k},\ (i,j,k) \in \Omega \end{aligned} \tag{11}$$

where $\|\mathcal{P}\|_\delta = \sum_{l}^{m}\sum_{k}^{K}\frac{(1+\delta)\sigma_{lk}(\mathcal{P})}{\delta+\sigma_{lk}(\mathcal{P})}$.

VI. Order-preserved Tensor Completion

As one of our main goals, we want to design a new tensor completion scheme that can accurately estimate the missing data entries while preserving the data entries' order in the data set. To achieve this, we formulate our order-preserved tensor completion problem in this section.

For two entries (i,j,k) and (i',j,k) which correspond to different OD pairs' monitoring data in the same time slot, if $x_{i,j,k} > x_{i',j,k}$, we obtain a quadruple instance (i,i',j,k). We define a quadruple set Δ consisting all the possible instances in the observed tensor $\mathcal{X}$ for i,i',j,k, where $1 \le i,i' \le I$, $i \ne i'$, $1 \le j \le J$ and $1 \le k \le K$.

To preserve the entry order, we reformulate the minimization problem in Eq.(11) as follows:

$$\begin{aligned} &\min_{\mathcal{P}} \|\mathcal{P}\|_\delta + \frac{\beta}{2}\sum_{(i,i',j,k)\in\Delta} L(p_{i,j,k} - p_{i',j,k}) \\ &subject\ to\ p_{i,j,k} = x_{i,j,k},\ (i,j,k) \in \Omega \end{aligned} \tag{12}$$

where $L(\bullet)$ is the max-margin squared hinge loss function $L(x) = \max(0, 1-x)^2$ (a monotonically non-increasing function). Thus, this loss function can preserve the estimated values $p_{i,j,k} > p_{i',j,k}$ when $x_{i,j,k} > x_{i',j,k}$, efficiently. In

(12), β is the trade-off parameter that balances the low-rank regularization and order-preserve regularization.

Although the minimization problem in Eq.(12) meets our goal, the order-preserving regularization $\sum_{(i,i',j,k)\in\Delta} L\left(p_{i,j,k}-p_{i',j,k}\right)$ does not include any trackable or trainable statistical parameters to help the training of $\mathcal{P}$.

To solve this problem, we define the estimated tensor $\mathcal{P}$ using a linear self-recovery model [33] $unfold\left(\mathcal{P}\right)=\mathbf{WQ}$, where $unfold\left(\mathcal{P}\right)$ is the $unfold$ matrix of tensor $\mathcal{P}$, $\mathbf{Q}\in\mathbb{R}^{I\times JK}$ is the $unfold$ matrix of incomplete observed tensor $\mathcal{X}$, $\mathbf{W}$ is the randomly initialized parameter matrix with constraints $\mathbf{W}\geq 0$ and $diag\left(\mathbf{W}\right)=0$. Accordingly, we have $q_{i,n}=x_{i,j,k}$, where $n=(k-1)\times J+j$. Using $unfold\left(\mathcal{P}\right)=\mathbf{WQ}$, we rewrite the tensor completion problem in (12) as follows

$$\begin{array}{l}\min\limits_{\mathcal{P},\mathbf{W}}\left\|\mathcal{P}\right\|_{\delta}+\frac{\beta}{2}\sum\limits_{(i.i',j,k)\in\Delta}L\left(\left(\mathbf{w}_{i,:}-\mathbf{w}_{i',:}\right)\mathbf{q}_{:,n}\right)\\ subject\ to\ p_{i,j,k}=x_{i,j,k}=q_{i,n},(i,j,k)\in\Omega\\ \ n=(k-1)\times J+j,\mathbf{W}\geq 0,diag\left(\mathbf{W}\right)=0\\ \ unfold\left(\mathcal{P}\right)=\mathbf{WQ}\end{array}\tag{13}$$

To increase the feasibility of the problem in (13), instead of keeping the equality constraint $unfold\left(\mathcal{P}\right)=\mathbf{WQ}$, we encode it as a regularization term in problem (Eq.(13)) and obtain the following minimization problem:

$$\begin{array}{l}\min\limits_{\mathcal{P},\mathbf{W}}\left\|\mathcal{P}\right\|_{\delta}+\frac{\alpha}{2}\left\|unfold\left(\mathcal{P}\right)-\mathbf{WQ}\right\|_F^2\\ +\frac{\beta}{2}\sum\limits_{(i.i',j,k)\in\Delta}L\left(\left(\mathbf{w}_{i,:}-\mathbf{w}_{i',:}\right)\mathbf{q}_{:,n}\right)\\ subject\ to\ p_{i,j,k}=x_{i,j,k}=q_{i,n},(i,j,k)\in\Omega\\ \ n=(k-1)\times J+j,\mathbf{W}\geq 0,diag\left(\mathbf{W}\right)=0\end{array}\tag{14}$$

where α and β are the trade-off parameters. Note in the problem Eq.(14), the linear self-recovery parameter matrix $\mathbf{W}$ is learned mainly based on the order-preserving loss function $\sum_{(i.i',n)\in\Delta} L\left(\left(\mathbf{w}_{i,:}-\mathbf{w}_{i',:}\right)\mathbf{q}_{:,n}\right)$, while the regularization $\left\|unfold\left(\mathcal{P}\right)-\mathbf{WQ}\right\|_F^2$ is a relaxation of the equality constaint $unfold\left(\mathcal{P}\right)=\mathbf{WQ}$.

As Eq.(14) builds a connection between the low rank constraint on tensor $\mathcal{P}$ and the order-preserving constraint on $\left\|unfold\left(\mathcal{P}\right)-\mathbf{WQ}\right\|_F^2$, it can propagate the order-preserving information to help the training of the low rank tensor estimation, thus further increasing the estimation accuracy.

VII. Solution for the Order Perserved Tensor Completion

To solve our order-preserved optimization problem (14), we propose an effective optimization strategy based on the Alernating Direction Method Multipliers (ADMM) technique. By introducing an auxiliary variable $\mathbf{Y}=unfold\left(\mathcal{P}\right)$, we reformulate the problem as follows:

$$\begin{array}{l}\min\limits_{\mathcal{P},\mathbf{W}}\left\|\mathcal{P}\right\|_{\delta}+\frac{\alpha}{2}\left\|\mathbf{Y}-\mathbf{WQ}\right\|_F^2+\frac{\beta}{2}\sum\limits_{(i.i',n)\in\Delta}L\left(\left(\mathbf{w}_{i,:}-\mathbf{w}_{i',:}\right)\mathbf{q}_{:,n}\right)\\ s.t.\ p_{i,j,k}=x_{i,j,k}=q_{i,n},\ (i,j,k)\in\Omega\\ n=(k-1)\times J+j,\ \mathbf{W}\geq 0,\ diag\left(\mathbf{W}\right)=0\\ \mathbf{Y}=unfold\left(\mathcal{P}\right)\end{array}\tag{15}$$

After using the augmented Lagrangian dual formulation to encode the equality constraint $\mathbf{Y}=unfold\left(\mathcal{P}\right)$, we have

$$\begin{array}{l}\min\limits_{\mathcal{P},\mathbf{W}}\left\|\mathcal{P}\right\|_{\delta}+\frac{\alpha}{2}\left\|\mathbf{Y}-\mathbf{WQ}\right\|_F^2+\frac{\mu}{2}\left\|unfold\left(\mathcal{P}\right)-\mathbf{Y}+\frac{\mathbf{Z}}{\mu}\right\|_F^2\\ \qquad+\frac{\beta}{2}\sum\limits_{(i.i',n)\in\Delta}L\left(\left(\mathbf{w}_{i,:}-\mathbf{w}_{i',:}\right)\mathbf{q}_{:,n}\right)\\ s.t.\ p_{i,j,k}=x_{i,j,k}=q_{i,n},(i,j,k)\in\Omega\\ n=(k-1)\times J+j,\mathbf{W}\geq 0,diag\left(\mathbf{W}\right)=0\end{array}\tag{16}$$

where $\mathbf{Z}$ is a Lagrange multiplier and $\mu>0$ is a trade-off parameter.

Next, we apply the alternating minimization method to solve the problem (16) iteratively.

1) The update of $\mathcal{P}$

By fixing $\mathbf{Y}^t$, $\mathbf{Z}^t$, and μ^t, we update $\mathcal{P}$ by solving following sub-problem:

$$\mathcal{P}^{t+1}=\min_{\mathcal{P}}\left\|\mathcal{P}\right\|_{\delta}+\frac{\mu^t}{2}\left\|\mathcal{P}-fold\left(\mathcal{Y}^t\right)+\frac{fold\left(\mathcal{Z}^t\right)}{\mu^t}\right\|_F^2\tag{17}$$

where $fold\left(\mathcal{Y}^t\right)$ and $fold\left(\mathcal{Z}^t\right)$ are in the tensor form obtained by applying the $fold$ operator to matrix $\mathbf{Y}^t$ and $\mathbf{Z}^t$, t denotes the number of iteration. To solve this problem, we first introduce the following theorem.

Theorem 1 [34]. Let $\mathbf{A}=\mathbf{US_AV^T}$ be the SVD of matrix $\mathbf{A}$ and $\sigma_{\mathbf{A}}=diag\left(\mathbf{S_A}\right)$. Let $F\left(\mathbf{Z}\right)=f\circ\sigma\left(\mathbf{Z}\right)$ be a unitarily invariant function. Then the optimal solution to the following problem

$$\min_{\mathbf{Z}}F\left(\mathbf{Z}\right)+\frac{\kappa}{2}\left\|\mathbf{Z}-\mathbf{A}\right\|_F^2\tag{18}$$

is $\mathbf{Z}^*=\mathbf{US}_{\mathbf{Z}}^*\mathbf{V^T}$, where $\sigma^*=diag\left(\mathbf{S}_{\mathbf{Z}}^*\right)$ and $\sigma^*=prox_{f,u}\left(\sigma_A\right)$. Here $prox_{f,u}\left(\sigma_A\right)$ is defined as:

$$prox_{f,u}\left(\sigma_{\mathbf{A}}\right)=\arg\min f\left(\sigma\right)+\frac{\mu}{2}\left\|\sigma-\sigma_{\mathbf{A}}\right\|_2^2\tag{19}$$

The above objective function Eq.(19) is a combination of concave and convex functions. Inspired by the difference of convex (DC) programing [35], we can solve (19) iteratively by updating

$$\sigma^{l+1}=\max\left(\sigma_A-\frac{\omega^l}{\mu},0\right)\tag{20}$$

where $\sigma_{\mathbf{A}}=diag\left(\mathbf{S_A}\right)$ and ω^l is the gradient of $f\left(\cdot\right)$ at σ^l.

Based on the t-SVD, tensor $\mathcal{P}$ can be decomposed into three tensors $\mathcal{U}$, $\mathcal{S}$, $\mathcal{V}$ and expressed as following:

$$\mathcal{P}=\mathcal{U}*\mathcal{S}*\mathcal{V}^T\tag{21}$$

which is equivalent to the following equation in the Fourier domain [30]:

$$\begin{array}{l}\begin{bmatrix}\bar{\mathbf{P}}_{(1)}&&\\&\ddots&\\&&\bar{\mathbf{P}}_{(K)}\end{bmatrix}=\begin{bmatrix}\bar{\mathbf{U}}_{(1)}&&\\&\ddots&\\&&\bar{\mathbf{U}}_{(K)}\end{bmatrix}\times\\ \begin{bmatrix}\bar{\mathbf{S}}_{(1)}&&\\&\ddots&\\&&\bar{\mathbf{S}}_{(K)}\end{bmatrix}\times\begin{bmatrix}\bar{\mathbf{V}}_{(1)}&&\\&\ddots&\\&&\bar{\mathbf{V}}_{(K)}\end{bmatrix}^T.\end{array}\tag{22}$$

where $\bar{\mathbf{P}}_{(k)}$, $\bar{\mathbf{U}}_{(k)}$, $\bar{\mathbf{S}}_{(k)}$ and $\bar{\mathbf{V}}_{(k)}$ are front slices of $\bar{\mathcal{P}}$, $\bar{\mathcal{U}}$, $\bar{\mathcal{S}}$ and $\bar{\mathcal{V}}$, respectively, and $\bar{\mathcal{P}}$, $\bar{\mathcal{U}}$, $\bar{\mathcal{S}}$ and $\bar{\mathcal{V}}$ are transformed from $\mathcal{P}$, $\mathcal{U}$, $\mathcal{S}$ and $\mathcal{V}$ by DFT. Thus, we have

$$\|\mathcal{P}\|_{\delta} = \sum_{k=1}^{K} \left\|\bar{\mathbf{P}}_{(k)}\right\|_{\delta} \tag{23}$$

Consequently, we transform the $\mathcal{P}$ and $fold\left(\mathcal{Y}^t\right) + \frac{fold\left(\mathcal{Z}^t\right)}{\mu^t}$ into Fourier domain as follows:

$$\begin{aligned} \bar{\mathcal{P}} &= fft\left(\mathcal{P}, [], 3\right) \\ \bar{\mathcal{D}}^t &= fft\left(fold\left(\mathcal{Y}^t\right) + \frac{fold\left(\mathcal{Z}^t\right)}{\mu^t}, [], 3\right) \end{aligned} \tag{24}$$

The subproblem Eq.(17) can be rewritten as follows:

$$\overline{P}^{t+1} = \min_{\overline{P}} \sum_{k=1}^{K} \left(\left\|\bar{\mathbf{P}}_{(k)}\right\|_{\delta} + \frac{\mu^t}{2} \left\|\bar{\mathbf{P}}_{(k)} - \overline{\mathbf{D}}^t{}_{(k)}\right\|_F^2 \right) \tag{25}$$

where $\bar{\mathcal{P}}^{t+1} = fft\left(\mathcal{P}^{t+1}, [], 3\right)$, $\bar{\mathbf{P}}_{(k)}$ and $\bar{\mathbf{D}}^t_{(k)}$ are the front slices of $\bar{\mathcal{P}}$, $\bar{\mathcal{D}}^t$ in Eq.(24), respectively.

Moreover, each $\left\|\bar{\mathbf{P}}_{(k)}\right\|_{\delta} + \frac{\mu^t}{2} \left\|\bar{\mathbf{P}}_{(k)} - \overline{\mathbf{D}}^t{}_{(k)}\right\|_F^2$ in Eq.(25) is independent of the others. Thus, the problem (Eq.(25)) can be divided into the following problems independently,

$$\bar{\mathbf{P}}^{t+1}_{(k)} = \min_{\bar{\mathbf{P}}_{(k)}} \left\|\bar{\mathbf{P}}_{(k)}\right\|_{\delta} + \frac{\mu^t}{2} \left\|\bar{\mathbf{P}}_{(k)} - \overline{\mathbf{D}}^t{}_{(k)}\right\|_F^2 \tag{26}$$

where $1 \leq k \leq K$.

Then, by substituting the function $F\left(\mathbf{Z}\right)$ in Eq.(18) with $\left\|\bar{\mathbf{P}}_{(k)}\right\|_{\delta}$, the variable $\mathbf{Z}$ and $\mathbf{A}$ with $\bar{\mathbf{P}}_{(k)}$, $\bar{\mathbf{D}}^t_{(k)}$, respectively, we can obtain

$$\sigma_{\mathbf{A}} = diag\left(\bar{\mathbf{D}}^t_{(k)}\right) \tag{27}$$

$$\omega^l = f'\left(\sigma^l\right) = \left(\frac{\left(1+\delta\right)\sigma^l}{\delta + \sigma^l}\right)' = \frac{\left(1+\delta\right)\delta}{\left(\delta + \sigma^l\right)^2} \tag{28}$$

where the function $f\left(\cdot\right)$ is the nonconvex penalty function Eq.(10).

Finally, we present our solution to the problem (Eq.(17)) in Algorithm.(2), which provides the updating rule for $\mathcal{P}^{t+1}$.

2) The update of $\mathbf{W}$

In the $(t+1)$-th iteration, by fixing $\mathbf{Y}^t$, we update $\mathbf{W}$ by solving the following sub-problem:

$$\begin{aligned} &\mathbf{W}^{t+1} = \underset{\mathbf{W}}{\operatorname{argmin}} \tfrac{\alpha}{2} \left\|\mathbf{Y}^t - \mathbf{W}\mathbf{Q}\right\|_F^2 + \tfrac{\beta}{2} \sum_{(i.i',n)\in\Delta} L\left(\left(\mathbf{w}_{i,:} - \mathbf{w}_{i',:}\right)\mathbf{q}_{:,n}\right) \\ &s.t.\ \mathbf{W} > 0, diag\left(\mathbf{W}\right) = 0 \end{aligned} \tag{29}$$

This sub-problem can be solved by stochastic gradient descent (SGD), and update $\mathbf{W}$ as follows:

$$\mathbf{w}^{t+1}_{i,:} = \mathbf{w}^{t+1}_{i,:} - \eta \left(\partial L\left(\mathbf{w}^t_{i,:}\right) + \frac{\alpha}{\beta\left|\Delta_i\right|} \left(\mathbf{w}^t_{i,:}\mathbf{Q} - \mathbf{y}^t_{i,:}\right)\mathbf{Q}^T \right) \tag{30}$$

$$\mathbf{w}^{t+1}_{i',:} = \mathbf{w}^{t+1}_{i',:} - \eta \left(\partial L\left(\mathbf{w}^t_{i',:}\right) + \frac{\alpha}{\beta\left|\Delta_{i'}\right|} \left(\mathbf{w}^t_{i',:}\mathbf{Q} - \mathbf{y}^t_{i',:}\right)\mathbf{Q}^T \right) \tag{31}$$

where $\left|\Delta_i\right|$ and $\left|\Delta_{i'}\right|$ represent the number of quadruples which contain the i-th OD pair and i'-th OD pair in Δ respectively, η is learning rate and $\mathbf{Q}$ is the $unfold$ of observed tensor $\mathcal{X}$. $\partial L\left(\mathbf{w}^t_{i,:}\right)$ and $\partial L\left(\mathbf{w}^t_{i',:}\right)$ are the local gradients of $\mathbf{w}^t_{i,:}$ and $\mathbf{w}^t_{i',:}$ in $L\left(\left(\mathbf{w}_{i,:} - \mathbf{w}_{i',:}\right)\mathbf{q}_{:,n}\right)$ respectively.

Moreover, we project the matrix $\mathbf{W}^{t+1}$ on to the non-negativity constraint by $\mathbf{W}^{t+1} = max\left(\mathbf{W}^{t+1}, 0\right)$, and let $diag\left(\mathbf{W}\right) = 0$.

As the loss function L is in the term $L\left(x\right) = \max\left(0, 1-x\right)^2$, we have

(a) if $\left(\mathbf{w}^t_{i,:} - \mathbf{w}^t_{i',:}\right)\mathbf{q}_{:,n} < 1$,

$$\partial L\left(\mathbf{w}^t_{i,:}\right) = 2\left(\left(\mathbf{w}^t_{i,:} - \mathbf{w}^t_{i',:}\right)\mathbf{q}_{:,n} - 1\right)\mathbf{q}^T_{:,n} \tag{32}$$

$$\partial L\left(\mathbf{w}^t_{i',:}\right) = -2\left(\left(\mathbf{w}^t_{i,:} - \mathbf{w}^t_{i',:}\right)\mathbf{q}_{:,n} - 1\right)\mathbf{q}^T_{:,n} \tag{33}$$

(b) if $\left(\mathbf{w}^t_{i,:} - \mathbf{w}^t_{i',:}\right)\mathbf{q}_{:,n} > 1$,

$$\partial L\left(\mathbf{w}^t_{i,:}\right) = \partial L\left(\mathbf{w}^t_{i',:}\right) = 0 \tag{34}$$

Algorithm 2 The solution to subproblem(17)

Input: Matrix $\mathbf{Y}^t \in \mathbb{R}^{I\times JK}$, Matrix $\mathbf{Z}^t \in \mathbb{R}^{I\times JK}$, ϵ and μ^t.
Output: $\mathcal{P}^{t+1} \in \mathbb{R}^{I\times I\times K}$.

```
fold the matrix Y^t ∈ R^{I×JK} into 𝒴^t ∈ R^{I×I×K}.
fold the matrix Z^t ∈ R^{I×JK} into 𝒵^t ∈ R^{I×I×K}.
Compute the auxiliary tensor 𝒟 = 𝒴^t − 𝒵^t/μ^t.
Compute D̄ = fft(𝒟, [ ], 3).
Compute each front slice of Ū, S̄ and V̄ by:
for k = 1, ··· , K do
    [U, S, V] = SVD(D̄_(k)).
    σ_A = diag(S).
    l = 1.
    repeat
        Compute the σ^l by Eq.(20).
        l = l + 1.
    until (σ^l − σ^{l−1} < ε)
    σ* = σ^l.
    Ū_(k) = U, S̄_(k) = diag{σ*}, V̄_(k) = V.
end for
Compute 𝒰 = ifft(Ū, [ ], 3).
Compute 𝒮 = ifft(S̄, [ ], 3).
Compute 𝒱 = ifft(V̄, [ ], 3).
Compute 𝒫^{t+1} = 𝒰 * 𝒮 * 𝒱^T
```

3) The update of auxiliary variable

Fixing $\mathcal{P}^{t+1}$, $\mathbf{Z}^t$ and $\mathbf{W}^{t+1}$, we update $\mathbf{Y}^{t+1}$ by solving the following sub-problem

$$\mathbf{Y}^{t+1} = \underset{\mathbf{Y}}{\operatorname{argmin}} \frac{\alpha}{2}\left\|\mathbf{Y} - \mathbf{W}\mathbf{Q}\right\|_F^2 + \frac{\mu}{2}\left\|unfold\left(\mathcal{P}\right) - \mathbf{Y} + \frac{\mathbf{Z}}{\mu}\right\|_F^2 \tag{35}$$

Firstly, we $unfold$ the tensor $\mathcal{P}^{t+1}$ and obtain the matrix $\mathbf{P}^{t+1}$. Then, we have a closed form solution to the problem in (Eq.35):

$$\mathbf{Y}^{t+1} = \max\left(\left(\left(\alpha\mathbf{W}^{t+1}\mathbf{Q} + \mu^t\left(\mathbf{P}^{t+1} + \frac{\mathbf{Z}^t}{\mu^t}\right)\right) / \left(\alpha + \mu^t\right), 0\right) \tag{36}$$

Next, we update the dual variables $\mathbf{Z}^{t+1}$ and the trade-off parameter μ^{t+1} as follows:

$$\mathbf{Z}^{t+1} = \mathbf{Z}^t + \mu^t\left(\mathbf{P}^{t+1} - \mathbf{Y}^{t+1}\right) \tag{37}$$

$$\mu^{t+1} = \lambda \mu^t \tag{38}$$

Finally, the complete solution to problem (15) can be shown in Algorithm 3, which is named **OTC**. In the Algorithm, ξ identifies the stoping condition of the iteration solution, that is, the iteration operations (in step 4-step 9) execute until the distance of estimated tensors of two iteration rounds is smaller than ξ.

Algorithm 3 OTC

Input: The observed tensor $\mathcal{X}$

Output: The estimated complete tensor $\mathcal{P}^{t+1}$.

1: Initial parameters: $\mathbf{W}^t > 0$, $diag\left(\mathbf{W}^t\right) = 0$, μ^t, $\mathbf{Z}^t = 0$, $\lambda > 1$, $t = 1$
2: $\mathcal{P}^t = \mathcal{X}$, $\mathbf{Y}^t = unfold\left(\mathcal{P}^t\right)$, $\mathbf{Q} = unfold\left(\mathcal{X}\right)$
3: **repeat**
4: Given μ^t, $\mathbf{Z}^t$, and $\mathbf{Y}^t$, update $\mathcal{P}^{t+1}$ by Algorithm.(2).
5: Given $\mathbf{Y}^t$, update $\mathbf{W}^{t+1}$ by Eq.(30) and Eq.(31).
6: Given $\mathbf{Z}^t$, $\mathbf{W}^{t+1}$, and $\mathcal{P}^{t+1}$ update $\mathbf{Y}^{t+1}$ by Eq.(35).
7: Given $\mathbf{Y}^{t+1}$, $\mathcal{P}^{t+1}$ update $\mathbf{Z}^{t+1}$ by Eq.(37).
8: Update μ^{t+1} by Eq.(38).
9: $t = t + 1$.
10: **until** $(\frac{\left\|\mathcal{P}^{t+1} - \mathcal{P}^t\right\|_F^2}{\left\|\mathcal{P}^t\right\|_F^2} \leq \xi)$

VIII. PERFORMANCE

Data set. We conduct extensive experiments using four public network monitoring traces to evaluate the performance of the proposed techniques, including two network traffic traces (Abilene [15], GÈANT [16]), a network latency trace Harvard226 [17]) and a throughput trace WS-DREAM [18]).

The Abilene network consists of 12 nodes thus 144 OD pairs, with the monitoring data collected every 5 minutes in 168 days. The GÈANT network consists of 23 nodes thus 529 OD pairs, with the monitoring samples taken every 15 minutes in 112 days. In Harvard226, application-level RTTs are measured between 226 Azureus clients every five minutes in 72 hours. The WS-Dream records the throughput between 142 users and 4,500 Web services over 64 consecutive time slices, at an interval of 15 minutes.

Instead of considering all entries, we care more about the top-k entries. We define the following three metrics to evaluate the accuracy of the top-k entries retrieved:

Definition 13: $NDCG@k$: In order to evaluate the accuracy of order-preservation among all the algorithms, we denote $\mathbf{x}$ as a vector of the ground-true monitoring data in a time slot and $\mathbf{p}$ as a vector of the estimated monitoring data in a time slot. π denotes a permutation by sorting $\mathbf{x}$ in the decreasing order, and $\hat{\pi}$ denotes a permutation by sorting $\mathbf{p}$ in the decreasing order. $\hat{\pi}_i$ denotes the position of the i-th element in the permutation $\hat{\pi}$. Let positive integer k be a truncation threshold, the Discounted Cumulative Gains ($DCG@k$) score and its normalized variant ($NDCG@k$) are defined respectively as $DCG@k\left(\mathbf{x}, \hat{\pi}\right) = \sum_{i=1}^{k} \frac{\hat{\pi}_i}{\log(i+1)}$ and $NDCG@k\left(\mathbf{x}, \hat{\pi}\right) = \frac{DCG@(\mathbf{x}, \hat{\pi})}{DCG@(\mathbf{x}, \pi)}$.

As the score of $NDCG@k$ is bounded by $[0, 1]$, $NDCG@k$ is maximized when the permutation of the top-k ground-truth data is equal to the permutation of the top-k estimated entries.

Definition 14: $Precision@k$: In order to evaluate the accuracy of order-preservation among all the algorithms, we denote $\mathbf{x}$ as a vector of the ground-true monitoring data in a time slot, and let the positive integer k be a truncation threshold. We define the precision as $Precision@k = \frac{|\Phi(k) \cap \Psi(k)|}{|\Psi(k)|}$, where $\Phi\left(k\right)$ represents the indexes (i, j) set of estimated top-k entries and $\Psi\left(k\right)$ represents the indexes (i', j') set of ground-truth top-k entries.

As the score of $Precision@k$ is bounded by $[0, 1]$, the $Precision@k$ is maximized when the set $\Phi\left(k\right)$ is equal to $\Psi\left(k\right)$, which means the indexes (i, j) of the top-k entries in the ground-truth data and estimated data are equal.

Definition 15: The $TopNMAE@k$: In order to evaluate the accuracy of the estimated monitoring data among all the algorithms, we denote $\mathbf{x}^k = \left\{x_1^k, \cdots, x_k^k\right\}$ as a vector of the top-k ground-true monitoring data in a time slot, and denote $\mathbf{p}^k = \left\{p_1^k, \cdots, p_k^k\right\}$ as a vector of the estimated values of $x_1^k, \cdots, x_k^k$. Then we have $TopNMAE@k = \frac{\sum_{i=1}^{k} \left|x_i^k - p_i^k\right|}{\sum_{i=1}^{k} \left|x_i^k\right|}$.

We empirically set $k = 20$ and $k = 50$ and the resulted $NDCG@20$, $Precision@20$, $TopNMAE@20$, $NDCG@50$, $Precision@50$ and $TopNMAE@50$,are the average values of all time slots monitored.

A. Convergence behavior

As shown in Algorithm.3, our Order-preserved Tensor Completion algorithm (termed OTC) involves an iteration process that updates the five parameters iteratively. To evaluate the convergence behavior, we use the metric Root Mean Square Error defined as $RMSE = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \left(x_{ijk} - p_{ijk}\right)^2}{I \times J \times K}$, where x_{ijk} and p_{ijk} are the raw and estimated data values. Fig.5 shows that OTC converges quickly in a few iterations for all four monitoring data traces.

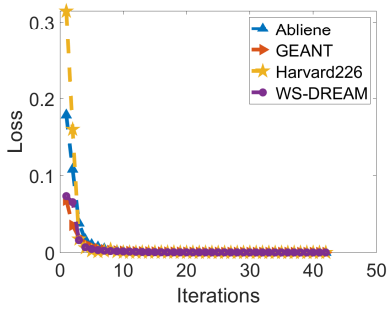

Fig. 5. The convergence behavior under different data traces.

B. Performance comparison

Besides our proposed OTC, we implement other four tensor completion algorithms based on different tensor decomposition techniques to estimate the top-k largest monitoring data. LTC [9] is the localized tensor completion algorithm based on

CP factorization, DTC [23] is a discrete tensor completion based on CP factorization, Tucker is the tucker based tensor completion proposed in [36], and T-SVD is the t-SVD based tensor completion proposed in [31].

Besides above four algorithms, we implement the OTC-NoOrder algorithm which adopts the OTC algorithm without the order-preserved constraint.

In our proposed model, we set the $\delta = 0.01$, $\alpha = 1$, and $\beta = 0.5$ in problem (Eq.(14)).

1) The accuracy of order-preservation

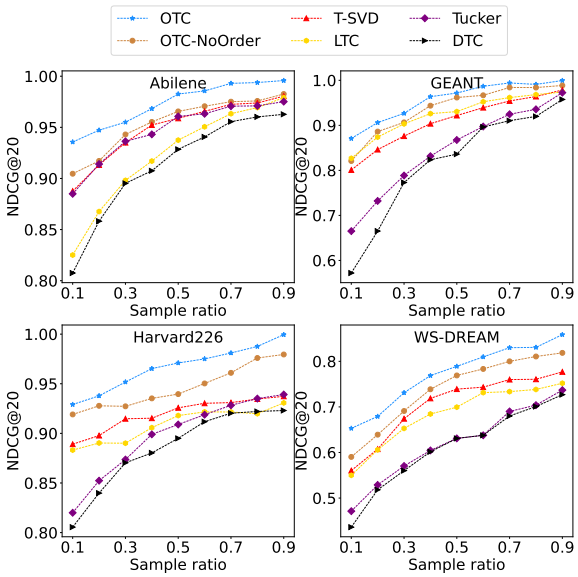

Fig. 6. The $NDCG@20$ of different sample ratio among different data traces.

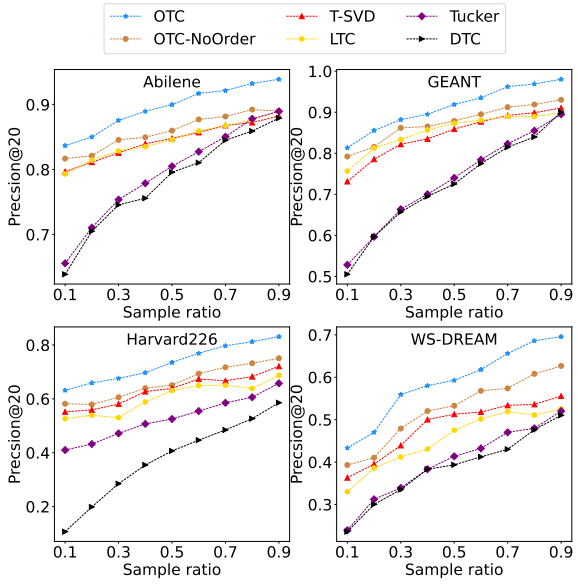

Fig. 7. The $Precsion@20$ of different sample ratio among different data traces.

We evaluate the accuracy of order-preservation among all the algorithms using the metrics $NDCG@20$, $Precsion@20$, $NDCG@50$ and $Precsion@50$. Higher $NDCG@20$, $Precsion@20$, $NDCG@50$ and $Precsion@50$ mean more accurate order-preservation.

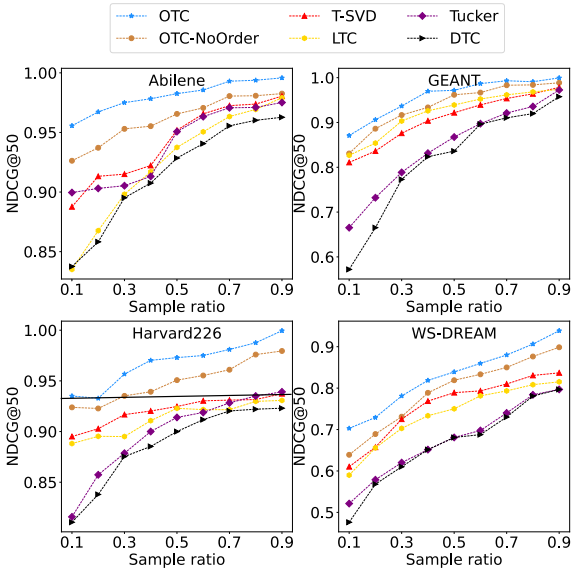

Fig. 8. The $NDCG@50$ of different sample ratio among different data traces.

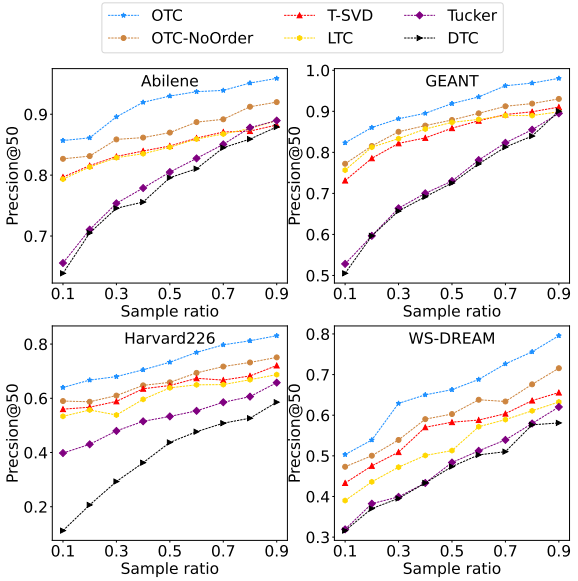

Fig. 9. The $Precsion@50$ of different sample ratio among different data traces.

As shown in Fig.6, Fig.7, Fig.8 and Fig.9 show that, $NDCG@20$, $Precsion@20$, $NDCG@50$ and $Precsion@50$ increase with the increase of sample ratio with more data observed.

Compared with other algorithms, our proposed OTC achieves the best performance by estimating the missing data with more accurate order. Although OTC-NoOrder achieves better performance than the other algorithms, its performance is still lower than that of OTC, which demonstrates that the order-preserved constraint in our OTC is effective.

As shown in Fig.8, $NDCG@50$ of our proposed OTC is over 0.93 evaluated under the Harvard226 data trace with the sampling ratio of 10%, while the other baselines can not achieve 0.93 for $NDCG@50$ even when the sampling ratio is over 80%. This demonstrates that our proposed OTC can make the permutation of estimated data more similar to the permutation of ground-truth data.

2) The accuracy of estimated top-k largest monitoring data

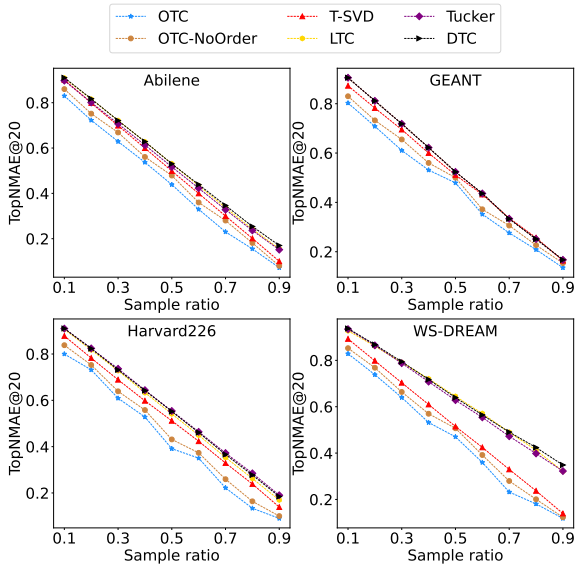

Fig. 10. The $TopNMAE@20$ of different sample ratio among different data traces.

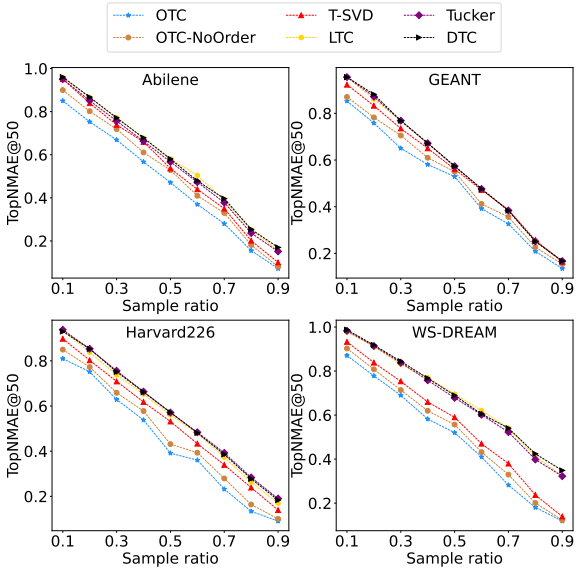

Fig. 11. The $TopNMAE@50$ of different sample ratio among different data traces.

We evaluate the accuracy of the estimated monitoring data using all the algorithms with the metrics $TopNMAE@20$ and $TopNMAE@50$, while the lower $TopNMAE@20$ and $TopNMAE@50$, mean a higher accuracy of the estimated top-k largest monitoring data.

As shown in Fig.10 and Fig.11, $TopNMAE@20$ and $TopNMAE@50$ decrease with the increase of the sample ratio. With more measurement samples, the tensor completion model can more accurately capture the hidden structure of the monitoring data to improve the estimation accuracy of the top-k largest monitoring data.

In these figures, we can find that, our proposed OTC and OTC-NoOrder achieve the lowest $TopNMAE@20$ and $TopNMAE@50$ in the cases of all sample ratios and all data traces. This demonstrates that our proposed OTC and OTC-NoOrder can more accurately capture the low-rank property of the monitoring data to increase the estimation performance of missing data.

In summary, compared with the state of the art tensor completion algorithms, our proposed OTC can provide more accurate missing data estimation and retrieval of top-k large entries, due to the integration of tighter low-rank constraint and order information in our model.

IX. Conclusion

To estimate the missing monitor data while preserving the data entries' order in the data set, this paper proposes a novel order-preserved tensor completion model that integrates both the low rank property and the order information into a joint learning problem for the estimation of missing data in network monitoring. In this model, we not only propose a novel nonconvex function to directly approximate the tensor rank, but also propose an order-preserved constraint under the linear self-recovery method that can be easily integrated with the tensor completion model. We also propose an ADMM-based algorithm to reformulate the proposed tensor completion problem into convex subproblems over each individual variable and solve them iteratively. We have conducted extensive experiments using four real data sets, including two network traffic data sets, a network delay data set, a network throughput data set. The experimental results show that our proposed approach outperforms five state-of-the-art methods both on the accuracy of estimating the missing data and on the accuracy in retrieving the largest top-k entries from the estimated data.

Acknowledgment

The work is supported by the National Science Foundation for Distinguished Young Scholars (No.62025201), the National Natural Science Foundation of China under Grant Nos.62102138, 61972144 and 61976087, the China National Postdoctoral Program for Innovative Talents (No.BX20200120), the China Postdoctoral Science Foundation (No.2020M682556), and the Hunan Provincial Natural Science Foundation of China (No.2021JJ40115).

References

[1] R. Zhu, B. Liu, D. Niu, Z. Li, and H. V. Zhao, "Network latency estimation for personal devices: A matrix completion approach," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 724–737, 2016.

[2] K. Xie, L. Wang, X. Wang, G. Xie, G. Zhang, D. Xie, and J. Wen, "Sequential and adaptive sampling for matrix completion in network monitoring systems," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 2443–2451.

[3] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices (extended version)," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 662–676, 2011.

[4] G. Gürsun and M. Crovella, "On traffic matrix completion in the internet," in *Proceedings of the 2012 Internet Measurement Conference*, 2012, pp. 399–412.

[5] Y.-C. Chen, L. Qiu, Y. Zhang, G. Xue, and Z. Hu, "Robust network compressive sensing," in *Proceedings of the 20th annual international conference on Mobile computing and networking*, 2014, pp. 545–556.

[6] K. Xie, Y. Chen, X. Wang, G. Xie, J. Cao, and J. Wen, "Accurate and fast recovery of network monitoring data: A gpu accelerated matrix completion," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 958–971, 2020.

[7] Y. Zhang, M. Roughan, C. Lund, and D. L. Donoho, "Estimating point-to-point and point-to-multipoint traffic matrices: An information-theoretic approach," *IEEE/ACM Transactions on networking*, vol. 13, no. 5, pp. 947–960, 2005.

[8] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *Proceedings of the joint international conference on Measurement and modeling of computer systems*, 2004, pp. 61–72.

[9] K. Xie, X. Wang, X. Wang, Y. Chen, G. Xie, Y. Ouyang, J. Wen, J. Cao, and D. Zhang, "Accurate recovery of missing network measurement data with localized tensor completion," *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2222–2235, 2019.

[10] 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*, vol. 35, no. 1, pp. 208–220, 2012.

[11] ——, "Tensor completion for estimating missing values in visual data," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, pp. 2114–2121.

[12] 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 Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 148–172, 2013.

[13] C. J. Hillar and L.-H. Lim, "Most tensor problems are np-hard," *Journal of the ACM (JACM)*, vol. 60, no. 6, pp. 1–39, 2013.

[14] Z. Zhang and S. Aeron, "Exact tensor completion using t-svd," *IEEE Transactions on Signal Processing*, vol. 65, no. 6, pp. 1511–1526, 2017.

[15] "The abilene observatory data collections. http://abilene. internet2.edu /observatory/data-collections.html."

[16] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 83–86, 2006.

[17] J. Ledlie, P. Gardner, and M. I. Seltzer, "Network coordinates in the wild." in *USENIX NSDI 2007*.

[18] Z. Zheng and M. R. Lyu, "Ws-dream: A distributed reliability assessment mechanism for web services," in *2008 IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN)*. IEEE, 2008, pp. 392–397.

[19] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE signal processing magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[20] S. Aja-Fernández, R. de Luis Garcia, D. Tao, and X. Li, *Tensors in image processing and computer vision*. Springer Science & Business Media, 2009.

[21] H. Tan, G. Feng, J. Feng, W. Wang, Y.-J. Zhang, and F. Li, "A tensor-based method for missing traffic data completion," *Transportation Research Part C: Emerging Technologies*, vol. 28, pp. 15–27, 2013.

[22] K. Xie, L. Wang, X. Wang, G. Xie, J. Wen, G. Zhang, J. Cao, and D. Zhang, "Accurate recovery of internet traffic data: A sequential tensor completion approach," *IEEE/ACM Transactions on Networking*, vol. 26, no. 2, pp. 793–806, 2018.

[23] K. Xie, J. Tian, X. Wang, G. Xie, J. Wen, and D. Zhang, "Efficiently inferring top-k elephant flows based on discrete tensor completion," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2170–2178.

[24] K. Shin, B. Hooi, J. Kim, and C. Faloutsos, "Densealert: Incremental dense-subtensor detection in tensor streams," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1057–1066.

[25] H. Kasai, W. Kellerer, and M. Kleinsteuber, "Network volume anomaly detection and identification in large-scale networks based on online time-structured traffic tensor tracking," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 636–650, 2016.

[26] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, and D. Zhang, "Graph based tensor recovery for accurate internet anomaly detection," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1502–1510.

[27] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, J. Cao, and D. Zhang, "Fast tensor factorization for accurate internet anomaly detection," *IEEE/ACM transactions on networking*, vol. 25, no. 6, pp. 3794–3807, 2017.

[28] E. d. S. e Silva and A. Streit, "Network anomaly detection based on tensor decomposition," in *18th Mediterranean Communication and Computer Networking Conference*, 2020.

[29] B. Romera-Paredes and M. Pontil, "A new convex relaxation for tensor completion," *Advances in Neural Information Processing Systems*, vol. 26, pp. 2967–2975, 2013.

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

[31] L. Deng, H. Zheng, X.-Y. Liu, X. Feng, and Z. D. Chen, "Network latency estimation with leverage sampling for personal devices: An adaptive tensor completion approach," *IEEE/ACM Transactions on Networking*, vol. 28, no. 6, pp. 2797–2808, 2020.

[32] X.-Y. Liu, S. Aeron, V. Aggarwal, X. Wang, and M.-Y. Wu, "Adaptive sampling of rf fingerprints for fine-grained indoor localization," *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2411–2423, 2015.

[33] X. Ning and G. Karypis, "Slim: Sparse linear methods for top-n recommender systems," in *2011 IEEE 11th International Conference on Data Mining*. IEEE, 2011, pp. 497–506.

[34] C. Lu, C. Zhu, C. Xu, S. Yan, and Z. Lin, "Generalized singular value thresholding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.

[35] P. D. Tao and L. T. H. An, "Convex analysis approach to dc programming: theory, algorithms and applications," *Acta mathematica vietnamica*, vol. 22, no. 1, pp. 289–355, 1997.

[36] X. Zhang, D. Wang, Z. Zhou, and Y. Ma, "Robust low-rank tensor recovery with rectification and alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 238–255, 2020.