Graph Spectral Regularized Tensor Completion for Traffic Data Imputation

Lei Deng, Xiao-Yang Liu, *Graduate Student Member, IEEE*, Haifeng Zheng, *Senior Member, IEEE*, Xinxin Feng, *Member, IEEE*, and Youjia Chen, *Member, IEEE*

***Abstract*— In intelligent transportation systems (ITS), incomplete traffic data due to sensor malfunctions and communication faults, seriously restricts the related applications of ITS. Recovering missing data from incomplete traffic data becomes an important issue for ITS. Existing works on traffic data imputation cannot achieve satisfactory accuracy due to inefficiently exploiting the underlying topological structure of the traffic data. In this paper, we model the topology of the road network as a graph and introduce graph Fourier transform (GFT) to process the traffic data. Then we utilize an algebraic framework termed as graph-tensor singular value decompositions (GT-SVD) to extract the hidden spatial information of traffic data. Furthermore, we propose a novel graph spectral regularized tensor completion algorithm based on GT-SVD and construct temporal regularized constraints to improve the recovery accuracy. The extensive experimental results on real traffic datasets demonstrate that the proposed algorithm outperforms the state-of-the-art methods under different missing patterns.**

***Index Terms*— ITS, tensor completion, graph Fourier transform, traffic data imputation.**

I. Introduction

With the increasing applications of intelligent transportation systems (ITS) in the modern life [1], [2], traffic data becomes important for data-driven traffic management [3]. It significantly helps monitor traffic operational state. However, due to sensor malfunctions and communication faults, large amounts of traffic data may lose in the processes of data collection and transmission [4], [5]. For example, it was reported that about half of the traffic data in Alberta, Canada was missing among seven years, and the data missing rate could even reach 90% sometime [6]. Data missing seriously affects the traffic control system to make appropriate traffic management strategies [7]. Therefore, reliable traffic data recovery is the basis of traffic management, such as traffic state monitoring and transportation planning [8].

Most works on traffic data imputation are developed by exploiting spatiotemporal correlations. For the spatial domain, since traffic information collecting devices or sensors are placed at particular geospatial locations of the roadways, spatial information can be exploited for traffic data imputation. However, the spatial structure of a large-scale transportation network is complex [9], bringing hurdles to traffic data imputation. An inevitable question is how to accurately represent the complex spatial structure of a transportation network. Previous works mainly represent the geometrical structure of a transportation network as a colored image [10], or convert the traffic states of sensing locations in a transportation network into a spatiotemporal matrix [11]. However, the topology of a transportation network cannot be represented efficiently in such ways. To address this issue, existing works model the transportation network as a graph [9], [12], where the nodes and the edges denote the sensors and the traffic flow directions, respectively. With this approach, the topology of a traffic network can be comprehensively represented by a graph.

Upon the graph model of traffic network, it is still challenging for efficient extracting spatial feature of traffic data [9]. A prevailing idea is to capture the structural feature by using the graph Laplacian matrix to construct spatial regularization [13], [15]. These methods achieved limited performance improvement due to inefficiently exploiting the spatial information. In this paper, we combine graph Fourier transform (GFT) [14] with tensor factorization to extract the topological feature of the transportation network more effectively.

For the temporal domain, a conventional method for capturing the temporal smoothness of traffic data is to construct the regularization using a Toeplitz matrix [15], [16]. However, these works commonly utilize the first-order Toeplitz matrix to depict the temporal smoothness, which is inflexible and imprecise. In this paper, we leverage the different smoothness of temporal dimensions and adopt the high-order and first-order Toeplitz matrices to capture the temporal correlations in "time interval" and "day" dimensions, respectively.

In this paper, we organize the traffic data as a tensor to make full use of spatiotemporal traffic information. We view the transportation network as a graph and utilize the graph Fourier transform (GFT) [14] to extract the spatial feature. In particular, inspired by t-SVD [17] that exploits the low-tubal-rank characteristic of traffic data, we combine GFT

Manuscript received 18 October 2020; revised 26 April 2021 and 26 June 2021; accepted 9 July 2021. Date of publication 3 August 2021; date of current version 9 August 2022. The work of Lei Deng and Haifeng Zheng was supported in part by the National Natural Science Foundation (NNSF) of China under Grant 61971139 and Grant 61571129. The work of Xinxin Feng was supported in part by NNSF of China under Grant 61601126. The Associate Editor for this article was Y. Kamarianakis. *(Corresponding author: Haifeng Zheng.)*

Lei Deng, Haifeng Zheng, Xinxin Feng, and Youjia Chen are with the Fujian Key Lab for Intelligent Processing and Wireless Transmission of Media Information, College of Physics and Information Engineering, Fuzhou University, Fuzhou 350116, China (e-mail: n191110002@fzu.edu.cn; zhenghf@fzu.edu.cn; fxx1116@fzu.edu.cn; youjia.chen@fzu.edu.cn).

Xiao-Yang Liu is with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: xl2427@columbia.edu).

Digital Object Identifier 10.1109/TITS.2021.3098637

and t-SVD for traffic tensor processing because the low-rank property still holds for the graph-tensor in the spectral domain. Then we utilize the graph-tensor singular value decomposition (GT-SVD) framework for mining the spatial correlations of traffic data. Furthermore, we propose a GT-SVD based graph spectral regularized tensor completion algorithm (GTC) for traffic data imputation. Meanwhile, the temporal regularization are also constructed to further improve the recovery performance.

The contributions of this paper are summarized as follows:

- We utilize a graph-tensor decomposition framework based on GFT and t-SVD (GT-SVD) for traffic data imputation. Compared to t-SVD [17] and the traditional multilinear algebraic framework [18], GT-SVD is more efficient in extracting spatial information for the signals with the topological structures. We also theoretically prove that GT-SVD holds similar algebraic properties with t-SVD, which allows us to analyze traffic data in a similar way as t-SVD. To the best of our knowledge, this is the first work that combines GFT with tensor factorization to efficiently extract the spatial information of traffic data.
- By analyzing the spatiotemporal correlations of road networks, we propose a GT-SVD based tensor completion algorithm (GTC) that utilizes the spatial information for traffic data recovery. We also develop regularization constraints by constructing Toeplitz matrices to further improve the recovery performance, which can impose the temporal smoothness constraints of traffic data more precisely. Instead of only using the first-order Toeplitz matrix [16], [19], we adopt the first-order and high-order Toeplitz matrices to capture the smoothness of "time interval" and "day" dimensions, respectively.
- We conduct extensive experiments to verify the effectiveness of the proposed GTC algorithm on real-world datasets under different missing patterns. The experimental results show that the proposed algorithm significantly outperforms the state-of-the-art traffic data imputation algorithms.

The remainder of the paper is organized as follows. In Section II, we describe the related works about traffic data imputation. In Section III, we introduce the notations and preliminaries. Sections IV and V give problem formulation and empirical study, respectively. Section VI presents the novel traffic data completion algorithm. In Section VII, we evaluate the performance of the proposed algorithm. Finally, the conclusion and future work are discussed in Section VIII.

II. Related Works

In the field of missing traffic data imputation, a variety of approaches have been developed, including K-Nearest Neighbors (KNN) [20], [21], neural network based approaches [22] and low-rank representation techniques [7].

K-Nearest Neighbors (KNN) is a modified form of mean value completion and it fills the missing entries with the mean values of K neighbors. For example, Tak *et al.* exploited the sectional KNN scheme for traffic data imputation [20]. Correlative KNN is also demonstrated to be effective for traffic state analysis [21]. However, these methods only consider the nearest neighbors of the missing entries and provide limited recovery accuracy.

With the development of machine learning and deep learning in recent years, more related approaches have been proposed for missing traffic data completion. Such approaches train a neural network model and recover the missing data through nonlinear mappings, which can achieve excellent performance in traffic data imputation. In particular, these imputation approaches utilize the historical traffic data for model training, and then directly recover the missing data once it is detected. For example, Yu *et al.* presented a graph convolutional generative autoencoder to address the real-time traffic speed estimation problem [23]. A hybrid spatiotemporal method that exploits the time series properties by a "prophet" model and extracts the spatial residuals information by an iterative random forest model, was proposed for traffic data imputation [24]. Zhong *et al.* [25] proposed several models to solve the traffic data imputation problem, such as genetically designed neural networks, regression models, factor models, and autoregressive integrated moving average (ARIMA) models. The genetically designed regression models are demonstrated to achieve better performance. In order to recover missing traffic data, Li *et al.* [26] proposed a fractionally strided 3D convolutional generative adversarial network for traffic data imputation. However, these methods require a large amount of historical data to train an efficient neural network model, implying that the imputation performance would be significantly degraded when historical data is not enough or even unavailable. Moreover, training an efficient neural network model is challenging because it requires high-performance computing capacity and large memory.

In recent years, by analyzing the low-rank property of data, matrix completion (MC) and tensor completion (TC) based algorithms have been proposed for data imputation [5], [7], [20], [27]. Traffic data is typically low-rank [28], which implies strong correlations in spatial and temporal domains. In other words, traffic data exhibits periodic similarity in time mode and strong coherence in adjacent sensors [8]. In the temporal domain, the traffic data of a certain sensor in adjacent moments, days, even weeks usually exhibits periodicity, which makes it possible to estimate the unknown data by exploiting the periodicity of traffic flow. Consequently, since traffic data is considered to be low-rank, it is reasonable to recover traffic data through matrix and tensor completion methods. For instance, Du *et al.* [29] proposed a multi-scale traffic data estimation algorithm based on the vehicular ad-hoc network (VANET) system to recover missing entries.

Compared with MC, TC is more applicable for multi-dimensional traffic data recovery. It is capable of mining the underlying correlations of higher-level data and improves recovery accuracy [30]. Currently, TC has been widely applied in data-driven ITS [7], [31], [32]. TC based approaches are mostly developed by using different tensor factorization strategies and exploiting the low-rank properties for traffic data imputation. Chen *et al.* [33] proposed a nonconvex truncated nuclear norm minimization based low-rank tensor completion model (LRTC-TNN) to

impute missing traffic data. CANDECOMP/PARAFAC (CP) decomposition, Tucker decomposition [18] and tensor SVD (t-SVD) [17] are commonly used approaches. For instance, Asif *et al.* [31] proposed a CP decomposition based method for traffic data recovery, called CP weighted optimization (CP-WOPT). Recently, an improved CP decomposition based model is applied for spatiotemporal traffic data imputation, called Bayesian Gaussian CANDECOMP/PARAFAC (BGCP) tensor decomposition model [34]. BGCP is developed by extending the Bayesian probabilistic matrix factorization model to higher-order tensors. Tucker decomposition is also commonly used for traffic data imputation [7], [28], [35]. For example, Tan *et al.* [7] proposed a tensor completion algorithm based on Tucker decomposition for traffic data estimation, which outperforms the traditional matrix-based algorithms. Similarly, Baggag *et al.* [35] also exploited Tucker decomposition and developed regularization to improve traffic data recovery accuracy. Chen *et al.* [32] proposed a Bayesian augmented tensor factorization (BATF) model, which tries to combine generic forms of domain knowledge from transportation systems. However, these works have not exploited the topological structure of traffic network, and cannot achieve satisfactory performance.

We notice that there are also several works focusing on developing graph-based traffic data imputation and forecasting algorithms [9], [13], [36]. Chen *et al.* [13] proposed a graph regularized local self-representation for missing traffic data imputation. The constructed nearest neighbor graph is used for characterizing the proximity relationship between samples, ensuring that the reconstructed samples do not deviate too much from their neighbors. Such a graph is developed based on KNN and cannot represent the real topology structure of a transportation network. Neural network based methods are also developed by exploiting the topological features of traffic data [9], [36]. In [36], the transition between network-wide traffic states at consecutive time steps is defined as a graph Markov process, namely graph Markov network. In [9], the graph wavelet is incorporated into a gated recurrent neural network for traffic forecasting. These methods combine graph structure with neural networks and mainly focus on traffic data prediction rather than data imputation. Different from the above works, our work incorporates the graph Fourier transform into tensor factorization for extracting the topological feature of a transportation network effectively. Besides, unlike neural network models that require a large amount of historical data for training, tensor-based methods do not need training data.

Motivated by the mentioned studies, considering the natural topological structure of a transportation network and the multi-dimension characteristics of traffic data, a customized tensor factorization strategy (GT-SVD) is designed for capturing the spatial correlations of the graph-based traffic data.

III. Notations and Preliminaries

We begin by introducing the basic notations and definitions used in this paper. We use calligraphy letters like $\mathcal{A}$, to denote a third-order tensor, boldface capital letters like $\boldsymbol{A}$ for a matrix, and boldface lowercase letters like $\boldsymbol{a}$ for a vector. The basis of graph signal processing (GSP) is introduced firstly since it will be frequently used for traffic data analysis. In particular, let $G = (V, E, \boldsymbol{A})$ be a graph with N nodes, where V is the set of nodes, E is the set of edges and $\boldsymbol{A}$ is the adjacency matrix of size $N \times N$. Laplacian matrix $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ is used to describe the connection information of nodes, where $\boldsymbol{D} = \text{diag}(d_1, d_2, \cdots, d_N)$ is the degree matrix, and $d_i = \sum_j \boldsymbol{A}_{ij}$. Suppose that the vector $\boldsymbol{f} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_N\}^\top$ is a graph signal, where $\boldsymbol{f}_i$ is attached to the i-th node. The frequency component of $\boldsymbol{f}$ is defined by graph Fourier transform (GFT):

Definition 1 ([GFT [14]): Given a graph signal $\boldsymbol{f} \in \mathbb{R}^N$ and its corresponding Laplacian matrix $\boldsymbol{L}$, the GFT of $\boldsymbol{f}$ is defined as

$$\widetilde{\boldsymbol{f}} = GFT\{\boldsymbol{f}\} = \boldsymbol{U}^* \boldsymbol{f}. \tag{1}$$

The inverse graph Fourier transform (IGFT) is

$$\boldsymbol{f} = IGFT\{\widetilde{\boldsymbol{f}}\} = \boldsymbol{U}\widetilde{\boldsymbol{f}}, \tag{2}$$

where $\boldsymbol{U}^*$ is the Hermitian of $\boldsymbol{U}$, which can be obtained by the eigendecomposition $\boldsymbol{L} = \boldsymbol{U}\Lambda\boldsymbol{U}^*$.

Definition 2 (Graph Convolution [14]): The graph convolution between two graph signals $\boldsymbol{f}$ and $\boldsymbol{g}$ is defined by

$$\boldsymbol{f} \star \boldsymbol{g} = \boldsymbol{U}(\boldsymbol{U}^* \boldsymbol{f} \odot \boldsymbol{U}^* \boldsymbol{g}), \tag{3}$$

where $\star$ denotes the graph convolution and $\odot$ is the Hadamard (element-wise) product.

For convenient discussion, we call a vectorized graph signal (i.e., $\boldsymbol{f} \in \mathbb{R}^{N\times 1}$) as a graph-vector. We extend the notion of GFT to the tensor case. For a tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, its (i, j, k)-th entry is represented as $\mathcal{A}(i, j, k)$, and notation $\mathcal{A}(:, :, i)$, $\mathcal{A}(:, i, :)$, $\mathcal{A}(i, :, :)$ are used to indicate the i-th frontal, lateral, horizontal slice of the tensor, respectively. For convenience, the i-th frontal slice is represented as $\mathcal{A}^{(i)}$, i.e., $\mathcal{A}^{(i)} = \mathcal{A}(:, :, i)$. Similar to the cases of vector and matrix, tensor Frobenius norm is $\|\mathcal{A}\|_F = \sqrt{\sum_{ijk} |\mathcal{A}(i, j, k)|^2}$. For ease of description, we call a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$ as a graph-tensor, whose tubes (fibers) are graph-vectors of size $1 \times 1 \times n_3$. Note that all the tubes (graph-vectors) of $\mathcal{A}$ have the same graph structure.

We take traffic data as an example for explaining the graph-tensor model. Suppose that there is a road network with n_3 sensors (nodes) and each sensor probes the traffic volume n_1 times per day within n_2 days. The traffic data can be modeled as a tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, where each tube $\mathcal{A}(i, j, :)$ is a graph-vector at the i-th time interval on the j-th day. The constructed graph-tensor model is shown in Fig. 1. The graph structure of all tubes (graph-vectors) can be described by the same Laplacian matrix. Different from previous works [7], [32], where the constructed traffic tensor arranges the sensors with an order that strictly corresponds to the spatial order. Hence the spatial correlations can be utilized for tensor analysis. In the graph-tensor, the order of sensors can be arranged without following the actual spatial order, but a corresponding Laplacian matrix is required to describe the spatial information. This advantage makes it more straightforward and easier to construct a traffic tensor.

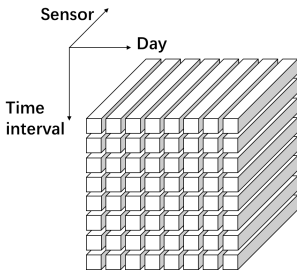


Fig. 1. An illustration of the graph-tensor model.

Based on the above introduction, GFT can be applied for tensor analysis. For a third-order graph-tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its GFT can be calculated by

$$\mathcal{A} \times_3 \boldsymbol{U}^* = \text{fold}((\boldsymbol{U}^* \otimes \boldsymbol{I}) \times \text{unfold}(\mathcal{A})), \tag{4}$$

where $\times_3$ is the mode-3 product, $\otimes$ is the Kronecker product, $\boldsymbol{I}$ is the identity matrix of size $n_3 \times n_3$, fold($\mathcal{A}$) is defined as

$$\text{fold}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} \\ \mathcal{A}^{(2)} \\ \vdots \\ \mathcal{A}^{(n_3)} \end{bmatrix}, \tag{5}$$

and unfold(fold($\mathcal{A}$)) $= \mathcal{A}$. Note that $\mathcal{A} \times_3 \boldsymbol{U}^*$ is a tensor defined in the graph spectral domain. We define $\widetilde{\mathcal{A}} = \text{GFT}(\mathcal{A}) = \mathcal{A} \times_3 \boldsymbol{U}^* \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ as the GFT representation of $\mathcal{A}$, and the inverse operation is $\mathcal{A} = \text{IGFT}(\widetilde{\mathcal{A}}) = \widetilde{\mathcal{A}} \times_3 \boldsymbol{U}$. In fact, $\widetilde{\mathcal{A}}$ can be equivalently computed by performing GFT to all the tubes, i.e., $\text{GFT}(\mathcal{A}) = \boldsymbol{U}^* \text{vec}(\mathcal{A}(i, j, :)), i = 1, 2, \cdots, n_1, j = 1, 2, \cdots, n_2$, where vec$(\cdot)$ denotes the vectorization operation that maps a tensor tube to a vector. Note that although this work focuses on one-dimensional transform, multi-dimensional transform is also demonstrated to be a promising direction [37], [38].

Furthermore, we denote the block diagonal matrix $\overline{\boldsymbol{A}} \in \mathbb{C}^{n_1 n_3 \times n_2 n_3}$ as

$$\overline{\boldsymbol{A}} = \text{bdiag}(\widetilde{\mathcal{A}}) = \begin{bmatrix} \widetilde{\mathcal{A}}^{(1)} & & & \\ & \widetilde{\mathcal{A}}^{(2)} & & \\ & & \ddots & \\ & & & \widetilde{\mathcal{A}}^{(n_3)} \end{bmatrix}, \tag{6}$$

and the inverse operation is $\widetilde{\mathcal{A}} = \text{ibdiag}(\overline{\boldsymbol{A}})$. Note that given any one of $\widetilde{\mathcal{A}}$, $\mathcal{A}$ and $\overline{\boldsymbol{A}}$, we can obtain the others.

Since the traditional tensor-product [17] is unapplicable for graph-tensor, we adopt the graph-tensor product.

Definition 3 (Graph-Tensor Product [40]): The graph-tensor product between $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is defined as

$$\mathcal{C} = \mathcal{A} *_g \mathcal{B} = (\mathcal{A} \times_3 \boldsymbol{U}^*) \triangle (\mathcal{B} \times_3 \boldsymbol{U}^*) \times_3 \boldsymbol{U}, \tag{7}$$

where $*_g$ denotes the graph-tensor product, and $\mathcal{C} \in \mathbb{R}^{n_1 \times n_4 \times n_3}$. $\triangle$ denotes the facewise product [41], which can be calculated by independently multiplying the frontal slices of two tensors. In fact, the (i, j)-th tube $\mathcal{C}(i, j, :)$ can be obtained by $\mathcal{C}(i, j, :) = \sum_{k=1}^{n_2} \mathcal{A}(i, k, :) \star \mathcal{B}(k, j, :)$, where $\star$ denotes the graph convolution between two tensor tubes (as defined in Definition 2).

Algorithm 1 Graph-Tensor Singular Value Decomposition (GT-SVD) [40]

Input: Original graph-tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, Laplacian matrix $L \in \mathbb{R}^{n_3 \times n_3}$.
Output: $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$, $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$.
$\widetilde{\mathcal{T}} = \text{GFT}(\mathcal{T})$.
for $k = 1 : n_3$ **do**
  $[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}] = \text{svd}(\widetilde{\mathcal{T}}^{(k)})$.
  $\widetilde{\mathcal{U}}^{(k)} = \boldsymbol{U}, \widetilde{\mathcal{S}}^{(k)} = \boldsymbol{S}, \widetilde{\mathcal{V}}^{(k)} = \boldsymbol{V}$.
end
$\mathcal{U} = \text{IGFT}(\widetilde{\mathcal{U}}), \mathcal{S} = \text{IGFT}(\widetilde{\mathcal{S}}), \mathcal{V} = \text{IGFT}(\widetilde{\mathcal{V}})$.

Definition 4 (Graph-Tensor Conjugate Transpose [40]): The conjugate transpose of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is $\mathcal{A}^\top \in \mathbb{R}^{n_2 \times n_1 \times n_3}$ whose frontal slices are conjugate transposed.

The transportation network can be regarded as a graph, which makes it easy to apply GFT to process the traffic data. By following previous works [39], [40], we specify GFT as the linear transform in tensor-SVD $*_L$ factorization [40] and obtain a graph-tensor singular value decomposition (GT-SVD) framework to capture the spatial information of traffic data. GT-SVD is an extension of t-SVD [17] from the Fourier domain to the graph spectral domain.

Definition 5 (GT-SVD [40]): The GT-SVD of a graph-tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is given by

$$\mathcal{A} = \mathcal{U} *_g \mathcal{S} *_g \mathcal{V}^\top, \tag{8}$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors [17], and $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor [17].

The algorithm of GT-SVD is given in Alg. 1. The difference between GT-SVD and t-SVD is that GT-SVD introduces GFT to replace the DFT operation along the third dimension. This modification helps us capture the spatial information of the graph-tensor while maintaining the properties of t-SVD. Although this work focuses on applying GT-SVD for traffic data processing, similar methods can also be applied for other data with topological structure.

Now we further give the definitions of graph-tensor tubal-rank and multi-rank.

Definition 6 (Graph-Tensor Tubal-Rank and Multi-Rank): The tubal-rank of $\mathcal{A}$, denoted as $\text{rank}_g(\mathcal{A})$, is defined as the number of non-zero singular tubes of $\mathcal{S}$. The graph-tensor multi-rank of tensor $\mathcal{A}$ is denoted as $\text{rank}_m(\mathcal{A}) = \boldsymbol{k}$, where $\boldsymbol{k}$ is a vector and $\boldsymbol{k}_i = \text{rank}(\overline{\boldsymbol{A}}^{(i)}), i = 1, \cdots, n_3$. Then $\text{rank}_g(\mathcal{A}) = \max(\boldsymbol{k}_1, \cdots, \boldsymbol{k}_{n_3})$.

We also introduce some lemmas based on graph-tensor, which are useful for simplifying the graph-tensor completion model.

Lemma 1: Suppose that $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ are two graph-tensors. Let $\mathcal{F} = \mathcal{A} *_g \mathcal{B}$, then the following properties hold.

(i) $\|\mathcal{A}\|_F^2 = \|\overline{\boldsymbol{A}}\|_F^2$,

(ii) $\mathcal{F} = \mathcal{A} *_g \mathcal{B}$ and $\overline{\boldsymbol{F}} = \overline{\boldsymbol{A}}\,\overline{\boldsymbol{B}}$ are equivalent to each other.

Here the notation $*_g$ represents the graph-tensor product (Definition 3). The proof of properties (i) and (ii) of Lemma 1 can be easily obtained since GFT exhibits similar properties with DFT. More specifically, using the Parseval relation for GFT, we have

$$\sum_{i=1}^{N} |\widetilde{f}(i)|^2 = \sum_{i=1}^{N} |f(i)|^2, \tag{9}$$

where $\widetilde{f} = \text{GFT}\{f\}$, we have property (i). Moreover, we can write $\mathcal{A} *_g \mathcal{B} = (\mathcal{A} \times_3 \boldsymbol{U}^*) \triangle (\mathcal{B} \times_3 \boldsymbol{U}^*) \times_3 \boldsymbol{U}$, which implies property (ii).

Lemma 2: Suppose that $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, and $\mathcal{C} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ are three graph-tensors. Then the following properties hold.

(i) If $\text{rank}_g(\mathcal{A}) = \hat{k}$, where $\hat{k} < min(n_1, n_2)$, then $\mathcal{A}$ can be expressed as a graph-tensor product $\mathcal{A} = \mathcal{G} *_g \mathcal{H}$, where $\mathcal{G} \in \mathbb{R}^{n_1 \times \hat{k} \times n_3}$ and $\mathcal{H} \in \mathbb{R}^{\hat{k} \times n_2 \times n_3}$, and $\text{rank}_g(\mathcal{G}) = \text{rank}_g(\mathcal{H}) = \hat{k}$.

(ii) $\text{rank}_g\left(\mathcal{B} *_g \mathcal{C}\right) \leq \min\left(\text{rank}_g(\mathcal{B}), \text{rank}_g(\mathcal{C})\right)$.

Note that rank_g denotes the graph-tensor tubal-rank (Definition 6). The proof of Lemma 2 can be easily obtained by following [43], which is given in Appendix A.

IV. Problem Statement

In the traffic information system, traffic volume is considered to be an important indicator for estimating traffic conditions. However, traffic data collection might be negatively affected by the sensor faults and disconnections of the network, which makes the traffic data incomplete. We aim to estimate the unknown traffic data by tensor completion.

A dataset that collects the traffic volume of n_3 sensors at n_1 time intervals within n_2 days can be viewed as a third-order graph-tensor (time interval $\times$ day $\times$ sensors, with a size of $n_1 \times n_2 \times n_3$). The traffic data of the road network is organized as a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, whose (i, j, k)-th element denotes the traffic volume measurement of the k-th sensor at the i-th time interval on the j-th day. The incomplete traffic data is represented as a tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with a number of missing entries that need to be estimated. The target of our work is to estimate the missing traffic data based on the measured entries.

For a tensor with partially missing entries, tensor completion aims to recover the missing entries from the incomplete tensor. To this end, we define the tensor completion objective function as follows

$$\min_{\mathcal{X}} \text{rank}_g(\mathcal{X}), \quad \text{s.t. } \mathcal{P}_\Omega(\mathcal{X} - \mathcal{M}) = 0, \tag{10}$$

where $\text{rank}_g(\mathcal{X})$ denotes the tubal-rank of $\mathcal{X}$, $\mathcal{P}_\Omega$ is the sampling operator and Ω is the set of observed entries.

V. Empirical Study

To better reveal the spatiotemporal correlations of traffic data, we conduct analysis using real-world highway traffic datasets.

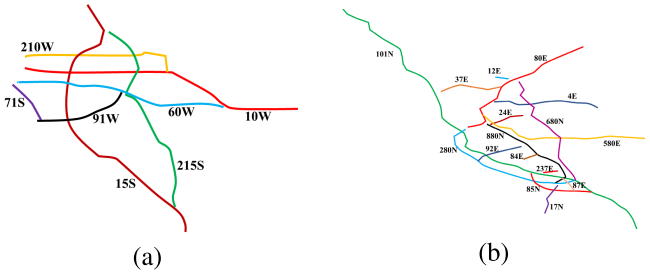


Fig. 2. Illustrations of the highways of the datasets. (a) The road network of the PeMS08 dataset. (b) The road network of the PeMS04 dataset.

A. Datasets

PeMS04 and PeMS08 datasets [12] from Caltrans Performance Measurement System (PeMS) are adopted in our experiments. PeMS collects the traffic volume, average speed, and average occupancy of the highway that connects the major cities in California. In particular, more than 39,000 sensors are deployed on the two sides of the highways. Each sensor collects the one-way traffic information on its side every 30 seconds. In this work, we focus on traffic volume recovery.

PeMS08: This dataset collects the traffic data of 170 sensors on 8 roads in San Bernardino Bay Area (see Fig. 2(a)). The data within 62 days (from July 1st, 2016 to August 31st, 2016) is selected for analysis and experiment.

PeMS04: This dataset collects the traffic data of 307 sensors on 29 roads in San Francisco Bay Area (see Fig. 2(b)). The time span of PeMS04 is from January 1st, 2018 to February 28th, 2018.

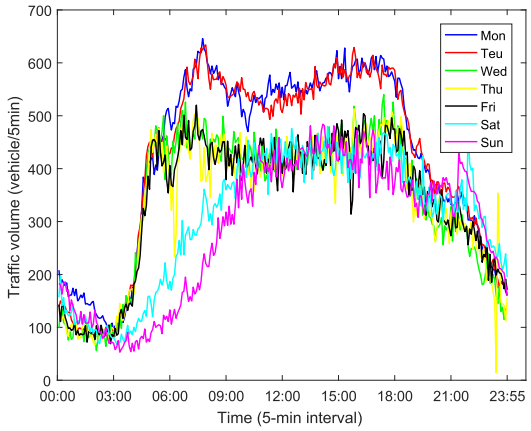
Recovering the data of PeMS04 is more difficult than PeMS08 since the road network of PeMS04 is more complicated and contains more sensors. The origin data of the PeMS08 and PeMS04 datasets are aggregated into every 5-minute interval. Thus, each sensor collects 288 data points a day. Note that the connections of sensors are directed because of the one-way traffic. Hence a directed graph can be used to represent the spatial structure of sensors. More details of the datasets can be found in [12].

B. Spatiotemporal Correlations of Traffic Data

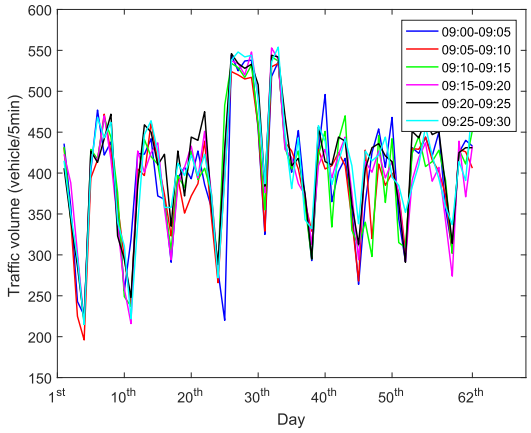
As mentioned above, exploiting the spatial and temporal relations among data is the core idea for traffic data imputation. To this end, we first analyze the characteristics of traffic volume data in the PeMS08 dataset.

For the temporal domain, the traffic conditions of adjacent days or adjacent intervals always exhibit similarities. To show the temporal characteristics more intuitively, we take the traffic volume of a randomly chosen sensor as an example to analyze the traffic conditions.

We first investigate the correlations of traffic volume on adjacent days. The traffic volume of a randomly chosen sensor within a week (from August 1st, 2016 to August 7th, 2016) is selected for visualization. As shown in Fig. 3(a), the traffic volume on the weekend is less than that on weekdays. This phenomenon also fits the traveling habits. Moreover, the traffic volume on different days also exhibits a similar tendency. For example, the traffic volume reaches the valley in the early



(a)



(b)

Fig. 3. Traffic volume of the selected sensor. (a) Traffic volume within a week. (b) Traffic volume at adjacent time intervals (09:00-09:30) within 62 days.

morning and evening and reaches the peak at noon, respectively. We also investigate the correlations of traffic volume at adjacent time intervals. Fig. 3(b) shows the traffic volume of the selected sensor at 6 sequential time intervals (09:00-09:30) within 62 days (from July 1st, 2016 to August 31st, 2016), where each line represents the traffic volume of the selected sensor at a specific time interval on different days. It can be seen that the traffic volume on days also exhibits strong periodicity. The strong periodicity and similarities in the temporal dimensions indicate that the missing traffic data can be inferred by exploiting these correlations. We also notice that the traffic volume on adjacent days (Fig. 3(b)) is much more varying than that at adjacent time intervals (Fig. 3(a)), which implies that the traffic volume on "time interval" and "day" dimensions exhibit different smoothness. This phenomenon inspires us to utilize Toeplitz matrices with different orders to capture the smoothness of "time interval" and "day", respectively.

For the spatial domain, the traffic volume of neighboring sensors that spreads along with upstream and downstream exhibits strong correlations due to the inherent connections of sensors. As mentioned above, a directed graph is used to describe the connection of sensors. In order to study the spatial

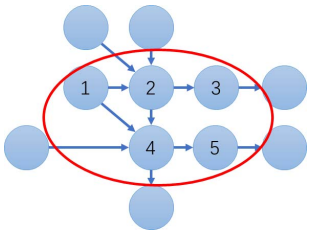


Fig. 4. The selected sub-graph of PeMS08. The arrows denote the directions of traffic.

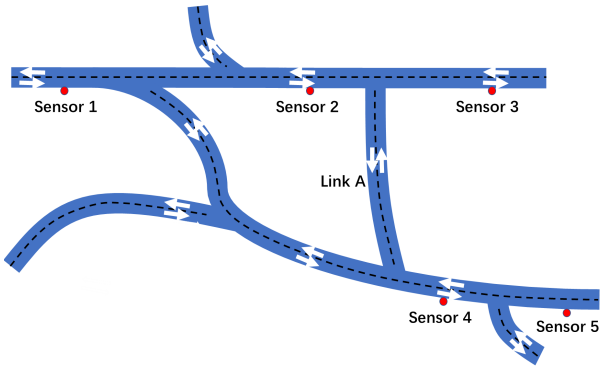


Fig. 5. An illustration of the transportation network corresponding to the selected sub-graph. The arrows denote the traffic flow directions, and the red points denote the sensors. Each sensor collects the one-way traffic information on its side.

correlations of sensors, we pick up a sub-graph of PeMS08 for analysis. As shown in Fig. 4, five connected sensors are selected to demonstrate the spatial correlations. As shown in Fig. 4, we use a directed graph to represent the spatial sensor correlations, where the arrow denotes the traffic flow direction from an original sensor (node) to a destination one. For example, a vehicle can run from sensor 2 to sensor 3, whereas it cannot run from sensor 3 to sensor 2 directly because of the one-way traffic. As a result, the upstream-downstream relationship between different sensors can be captured by the constructed directed graph and the corresponding GFT in our work. A simplified illustration of the transportation network corresponding to the sub-graph is also presented in Fig. 5. We first introduce the Pearson correlation coefficient to describe the correlation strength between two sensors. Pearson coefficient is a normally used metric to quantify the correlation of sequences [42], which is defined as follows

$$p_{ab} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{a_i - \overline{a}}{\sigma_a}\right)\left(\frac{b_i - \overline{b}}{\sigma_b}\right). \tag{11}$$

Note that p_{ab} denotes the Pearson coefficient of sequences $\boldsymbol{a}$ and $\boldsymbol{b}$; n is the length of sequence; $\overline{a}$ and a_i denote the average value and the i-th entry of $\boldsymbol{a}$, respectively; $\overline{b}$ and b_i are also defined similarly; σ denotes standard deviation. We calculate the Pearson coefficients by using the traffic volume of the five sensors in the sub-graph on July 1st, 2016. The results are presented in Table I. Obviously, sensor 3 and sensor 5 exhibit the weakest correlation (the lowest Pearson coefficient) since they are not connected. The Pearson coefficients between the other sensors are all above 0.86, which implies the high correlations of the connected sensors. We note that the

TABLE I
PEARSON COEFFICIENTS OF TRAFFIC VOLUME DATA BETWEEN DIFFERENT SENSORS

	sensor 1	sensor 2	sensor 3	sensor 4	sensor 5
sensor 1	1	0.92	0.92	0.97	0.91
sensor 2	0.92	1	0.86	0.93	0.93
sensor 3	0.92	0.86	1	0.93	0.83
sensor 4	0.97	0.93	0.93	1	0.93
sensor 5	0.91	0.93	0.83	0.93	1

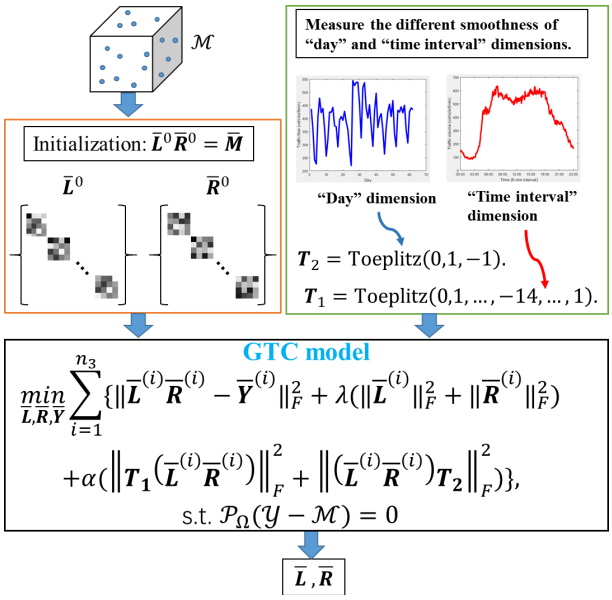


Fig. 6. The overall framework of the proposed algorithm for traffic data imputation.

correlations of traffic volume collected by adjacent sensors would be generally higher than the non-adjacent sensors. In the real transportation scenario, the traffic conditions might not strictly follow this rule occasionally because they would be influenced by the drivers' preferences.

By analyzing the traffic volume above, it is obvious that traffic data exhibits strong correlations and similarities in both spatial and temporal domains. These features make it possible to use mathematical methods to impute the missing traffic data.

VI. THE PROPOSED GRAPH SPECTRAL REGULARIZED TENSOR COMPLETION ALGORITHM

In this section, we elaborate on the proposed GT-SVD based tensor completion algorithm (GTC) with graph spectral regularization. The overall framework of the proposed algorithm is shown in Fig. 6. In particular, given an incomplete graph-tensor of traffic data $\mathcal{M}$ and the initial rank, we decompose $\mathcal{M}$ as two components in the graph-spectral domain. That is, $\overline{L}^0\overline{R}^0 = \overline{M}$, where $\overline{L}^0$ and $\overline{R}^0$ are block-diagonal matrices and can act as the initializations of $\overline{L}$ and $\overline{R}$ for solving the GTC model. Meanwhile, we measure the different smoothness of traffic data on the "day" and "time interval" dimensions and develop Toeplitz matrices T_1, T_2 as regularization for further performance improvement. Finally, the estimated components $\overline{L}$ and $\overline{R}$ can be obtained by solving the GTC model. Then the traffic data can be recovered by $\mathcal{X}_{\text{est}} = \text{IGFT}(\text{ibdiag}(\overline{L}\,\overline{R}))$. The details of solving procedure will be presented in Alg. 2.

Algorithm 2 GTC for Traffic Data Completion

Input: $\mathcal{M} \in \mathbb{R}^{n_1\times n_2\times n_3}$: the incomplete tensor data,
Ω: the measured subset,
T_1, T_2: temporal regularized matrices,
α, λ: adjusting parameters,
$r^0 \in \mathbb{R}^{n_3}$: the initial multi-rank,
Maxiter: max iteration,
ϵ: the tolerance factor.

Initialize: $\overline{L}^0, \overline{R}^0, \overline{Y}^0$.
while $k \le Maxiter$ **do**
  Fix $\overline{Y}^k$ and $\overline{R}^k$, update $\overline{L}$ by
  $$\overline{L}^{k+1} \leftarrow \arg\min_{\overline{L}} \text{Loss}(\overline{Y}^k, \Omega, \overline{R}^k, \overline{L}, T_1, T_2, \alpha, \lambda).$$
  Fix $\overline{Y}^k$ and $\overline{L}^{k+1}$, update $\overline{R}$ by
  $$\overline{R}^{k+1} \leftarrow \arg\min_{\overline{R}} \text{Loss}(\overline{Y}^k, \Omega, \overline{L}^{k+1}, \overline{R}, T_2, T_1, \alpha, \lambda).$$
  Fix $\overline{L}^{k+1}$ and $\overline{R}^{k+1}$, update $\overline{Y}$ by
  $$\overline{Y}^{k+1} \leftarrow \overline{L}^{k+1}\overline{R}^{k+1} + \mathcal{P}_\Omega(\overline{M} - \overline{L}^{k+1}\overline{R}^{k+1}).$$
  Perform rank adjusting strategy to update the estimated rank r^k and the sizes of $\overline{L}^{k+1}$ and $\overline{R}^{k+1}$.
  Check convergence conditions: $||\overline{L}^{k+1} - \overline{L}^k||_\infty < \epsilon$, $||\overline{R}^{k+1} - \overline{R}^k||_\infty < \epsilon$, $||\overline{Y}^{k+1} - \overline{Y}^k||_\infty < \epsilon$.
  $k \leftarrow k + 1$.
end
Output: $\mathcal{X}_{\text{est}} = \text{IGFT}(\text{ibdiag}(\overline{L}^{k+1}\overline{R}^{k+1}))$.

A. Low-Rank Graph-Tensor Factorization

It is known that the problem of rank minimization is NP-hard. To solve (10), structural assumptions about the graph-tensor $\mathcal{X}$ are required. We assume that traffic tensor can be well-approximated by the graph-tensor product $\mathcal{L} *_g \mathcal{R}$, where $\mathcal{L} \in \mathbb{R}^{n_1\times\hat{r}\times n_3}$ and $\mathcal{R} \in \mathbb{R}^{\hat{r}\times n_2\times n_3}$ are two smaller tensors, and $\hat{r}$ is much smaller than $\min(n_1, n_2)$.

We use the PeMS08 dataset to verify the above assumptions. We compute the GFT of the constructed graph-tensor and plot the spectral domain singular value distribution in Fig. 7, where the first 20 largest singular values for each frontal slice are presented. Obviously, the singular values decline and approximate to zero fast, demonstrating that the traffic data is low-rank in the spectral domain.

According to Lemma 2, graph-tensor tubal-rank exhibits similar properties as the matrix rank. Hence the low-rank decomposition strategy can be utilized to solve the large-scale tensor completion in an efficient way [43]. Then the graph-tensor $\mathcal{M} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is factorized into two smaller tensors $\mathcal{L} \in \mathbb{R}^{n_1\times\hat{r}\times n_3}$ and $\mathcal{R} \in \mathbb{R}^{\hat{r}\times n_2\times n_3}$ with tubal-rank up to $\hat{r}$, which satisfies $\mathcal{M} = \mathcal{L} *_g \mathcal{R}$, and $\text{rank}_g(\mathcal{L}) = \text{rank}_g(\mathcal{R}) = \hat{r}$.

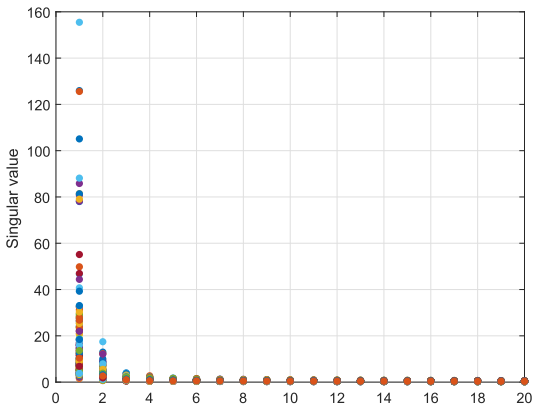


Fig. 7. Spectral domain singular value distribution of PeMS08. Each color denotes the singular values for one frontal slice in the spectral domain.

It would be more effective to update and store two tensors with smaller sizes. Meanwhile, following the previous works on matrix and tensor completion [15], [19], [27], we have the following tensor factorization formulation:

$$\min_{\mathcal{L},\mathcal{R}} \left\| \mathcal{P}_{\Omega}(\mathcal{L} *_g \mathcal{R} - \mathcal{M}) \right\|_F^2. \tag{12}$$

In (12), $\mathcal{P}_{\Omega}$ has constraints on the product of $\mathcal{L}$ and $\mathcal{R}$, hence it would be challenging to update $\mathcal{L}$ and $\mathcal{R}$. We further approximate the minimization problem (12) by introducing an intermediate variable $\mathcal{Y}$ as follows

$$\begin{aligned} &\min_{\mathcal{L},\mathcal{R},\mathcal{Y}} \left\| \mathcal{L} *_g \mathcal{R} - \mathcal{Y} \right\|_F^2 + \left(\|\mathcal{L}\|_F^2 + \|\mathcal{R}\|_F^2 \right), \\ &\text{s.t. } \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{M}. \end{aligned} \tag{13}$$

Consequently, we convert the constraint on the product of $\mathcal{L}$ and $\mathcal{R}$ into $\mathcal{Y}$. Obviously, it is easier to project $\mathcal{Y}$ onto Ω than (12). We add the term $\left(\|\mathcal{L}\|_F^2 + \|\mathcal{R}\|_F^2 \right)$ in order to make the optimization function more convex.

We suppose that $\text{rank}_m(\mathcal{Y}) = \boldsymbol{r}$ and $\text{rank}_g(\mathcal{Y}) = \hat{r}$, where $\boldsymbol{r}_i = \text{rank}(\overline{\boldsymbol{Y}}^{(i)}), i = 1, \cdots, n_3$, and $\hat{r} = \max(\boldsymbol{r}_1, \cdots, \boldsymbol{r}_{n_3})$. Then $\overline{\boldsymbol{Y}}^{(i)}$ can be written as the product of two matrices $\hat{\boldsymbol{L}}^{(i)} \in \mathbb{C}^{n_1 \times \boldsymbol{r}_i}$ and $\hat{\boldsymbol{R}}^{(i)} \in \mathbb{C}^{\boldsymbol{r}_i \times n_2}$, where $\hat{\boldsymbol{L}}^{(i)}$ and $\hat{\boldsymbol{R}}^{(i)}$ are respectively the i-th block-diagonal matrices of $\hat{\boldsymbol{L}} \in \mathbb{C}^{n_1 n_3 \times (\sum_{i=1}^{n_3} \boldsymbol{r}_i)}$ and $\hat{\boldsymbol{R}} \in \mathbb{C}^{(\sum_{i=1}^{n_3} \boldsymbol{r}_i) \times n_2 n_3}$. We set $\overline{\boldsymbol{L}}^{(i)} = [\hat{\boldsymbol{L}}^{(i)}; \boldsymbol{0}] \in \mathbb{C}^{n_1 \times \hat{r}}$ and $\overline{\boldsymbol{R}}^{(i)} = [\hat{\boldsymbol{R}}^{(i)}; \boldsymbol{0}] \in \mathbb{C}^{\hat{r} \times n_2}$. Thus, $\overline{\boldsymbol{L}}\,\overline{\boldsymbol{R}} = \hat{\boldsymbol{L}}\hat{\boldsymbol{R}}$. According to Lemma 1, (13) can be rewritten as follows

$$\begin{aligned} &\min_{\overline{\boldsymbol{L}},\overline{\boldsymbol{R}},\overline{\boldsymbol{Y}}} \sum_{i=1}^{n_3} \left\{ \left\| \overline{\boldsymbol{L}}^{(i)} \overline{\boldsymbol{R}}^{(i)} - \overline{\boldsymbol{Y}}^{(i)} \right\|_F^2 + \lambda \left(\left\| \overline{\boldsymbol{L}}^{(i)} \right\|_F^2 + \left\| \overline{\boldsymbol{R}}^{(i)} \right\|_F^2 \right) \right\}, \\ &\text{s.t. } \mathcal{P}_{\Omega}(\mathcal{Y} - \mathcal{M}) = 0. \end{aligned} \tag{14}$$

where λ is a parameter for the adjustable trade-off between a precise fit to the measured data and the low-rank approximation.

B. Constructing Temporal Regularization

As mentioned before, traffic data exhibits temporal smoothness. In order to capture the temporal smoothness, Toeplitz matrices are adopted to construct temporal constraints. Here we refer the Toeplitz matrix $\boldsymbol{T} = \text{Toeplitz}(0, 1, -1)$, i.e.,

$$\text{Toeplitz}(0, 1, -1) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & \cdots \\ 0 & 1 & -1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & -1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{15}$$

to a first-order Toeplitz matrix, and refer $\boldsymbol{T} = \text{Toeplitz}(0, 1, -2, 1)$, i.e.,

$$\text{Toeplitz}(0, 1, -2, 1) = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & \cdots \\ 0 & 1 & -2 & 1 & 0 & \cdots \\ 0 & 0 & 1 & -2 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{16}$$

to a second-order Toeplitz matrix and so on. We note that the traffic tensor is transformed into the graph spectral domain by executing GFT along the "sensor" dimension, indicating that the smoothness on "time interval" and "day" dimensions remain unchanged and still can be depicted by Toeplitz matrices in the graph spectral domain. We choose Toeplitz matrices $\boldsymbol{T}_1 \in \mathbb{R}^{n_1 \times n_1}$ and $\boldsymbol{T}_2 \in \mathbb{R}^{n_2 \times n_2}$ with different orders to capture the temporal correlations of "time interval" and "day" dimensions respectively, whose temporal smoothness is significantly different. Then (14) can be further written as

$$\begin{aligned} &\min_{\overline{\boldsymbol{L}},\overline{\boldsymbol{R}},\overline{\boldsymbol{Y}}} \sum_{i=1}^{n_3} \left\{ \left\| \overline{\boldsymbol{L}}^{(i)} \overline{\boldsymbol{R}}^{(i)} - \overline{\boldsymbol{Y}}^{(i)} \right\|_F^2 + \lambda \left(\left\| \overline{\boldsymbol{L}}^{(i)} \right\|_F^2 + \left\| \overline{\boldsymbol{R}}^{(i)} \right\|_F^2 \right) \right. \\ &\qquad \left. + \alpha \left(\left\| \boldsymbol{T}_1 (\overline{\boldsymbol{L}}^{(i)} \overline{\boldsymbol{R}}^{(i)}) \right\|_F^2 + \left\| (\overline{\boldsymbol{L}}^{(i)} \overline{\boldsymbol{R}}^{(i)}) \boldsymbol{T}_2 \right\|_F^2 \right) \right\}, \\ &\text{s.t. } \mathcal{P}_{\Omega}(\mathcal{Y} - \mathcal{M}) = 0, \end{aligned} \tag{17}$$

where α and λ are balancing parameters for corresponding terms. We call the above problem in (17) as GT-SVD based tensor completion (GTC).

C. Solving GTC for Traffic Data Completion

To solve the complicated optimization problem in GTC, we extend sparsity regularized matrix factorization (SRMF) [19] to the graph spectral domain. At the same time, the spatiotemporal constraint matrices of SRMF are also replaced by two temporal constraint matrices $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$. Firstly, alternating constrained least squares is performed to factorize a matrix subject to sparse constraints. In particular, we fix two variables and update the left one. For example, assuming that $\overline{\boldsymbol{R}}$ and $\overline{\boldsymbol{Y}}$ are fixed, from (17), we define the corresponding loss function as

$$\begin{aligned} &\text{Loss}(\overline{\boldsymbol{Y}}, \Omega, \overline{\boldsymbol{R}}, \overline{\boldsymbol{L}}, \boldsymbol{T}_1, \boldsymbol{T}_2, \alpha, \lambda) \\ &\quad = \sum_{i=1}^{n_3} \left\{ \left\| \overline{\boldsymbol{L}}^{(i)} \overline{\boldsymbol{R}}^{(i)} - \overline{\boldsymbol{Y}}^{(i)} \right\|_F^2 + \lambda \left\| \overline{\boldsymbol{L}}^{(i)} \right\|_F^2 \right. \\ &\qquad \left. + \alpha \left(\left\| \boldsymbol{T}_1 (\overline{\boldsymbol{L}}^{(i)} \overline{\boldsymbol{R}}^{(i)}) \right\|_F^2 + \left\| (\overline{\boldsymbol{L}}^{(i)} \overline{\boldsymbol{R}}^{(i)}) \boldsymbol{T}_2 \right\|_F^2 \right) \right\}, \\ &\qquad \text{s.t. } \mathcal{P}_{\Omega}(\mathcal{Y} - \mathcal{M}) = 0. \end{aligned} \tag{18}$$

Then $\overline{\boldsymbol{L}}$ can be updated by

$$\overline{\boldsymbol{L}} \leftarrow \arg\min_{\overline{\boldsymbol{L}}} \text{Loss}(\overline{\boldsymbol{Y}}, \Omega, \overline{\boldsymbol{R}}, \overline{\boldsymbol{L}}, \boldsymbol{T}_1, \boldsymbol{T}_2, \alpha, \lambda), \tag{19}$$

TABLE II
DETAILS OF DATASETS

Dataset	PeMS04	PeMS08
Data type	Traffic volume	Traffic volume
Time span	1/1/2018-2/28/2018	7/1/2016-8/31/2016
Days	59	62
Time interval	5 min	5 min
Scale	307 sensors	170 sensors
Tensor size	288×59×307	288×62×170

where (19) can be solved by many approaches such as the invMinL2 function in MATLAB and the "myinverse" function from SRMF.

Moreover, the sizes of tensor $\mathcal{L}$ and $\mathcal{R}$ depend on the estimated rank of each block-diagonal matrix of $\overline{\boldsymbol{Y}}$. However, it is difficult to accurately estimate the rank of traffic tensor data. An overestimated rank will increase the complexities in tensor factorization and reconstruction while an underestimated rank can only achieve limited recovery accuracy. Therefore, a rank adjusting strategy [44] is adopted to estimate the rank of each block-diagonal matrix $\overline{\boldsymbol{Y}}^{(i)}$ efficiently.

Once the convergence conditions are satisfied or the number of iterations reaches the maximum, the processes of updating $\overline{\boldsymbol{L}}$ and $\overline{\boldsymbol{R}}$ terminates. Then we can obtain the estimated data via $\overline{\boldsymbol{X}}_{\text{est}} = \overline{\boldsymbol{L}}\,\overline{\boldsymbol{R}}$ in the graph spectral domain. Thus, the final estimated traffic tensor data $\mathcal{X}_{\text{est}}$ can be computed by $\mathcal{X}_{\text{est}} = \text{IGFT}(\text{ibdiag}(\overline{\boldsymbol{X}}_{\text{est}}))$. The detailed optimization procedure is given in Alg. 2. We note that the proposed GTC algorithm supports parallel computing for real-time implementation. For example, the block-diagonal matrices of $\overline{\boldsymbol{L}}$ ($\overline{\boldsymbol{R}}$) are updated independently in the graph spectral domain, implying that they can be computed simultaneously. Moreover, GFT and IGFT operations can also be conducted parallelly for each third-dimensional tubes.

VII. PERFORMANCE EVALUATION

To verify the recovery performance of the proposed GTC algorithm, both PeMS04 and PeMS08 traffic volume datasets are used for experiments. The details of the datasets are presented in Table II. The sampling rate S_p is varying from 0.1 to 0.9 for comprehensive performance evaluation.

A. Performance Metric

To evaluate the estimation performance fairly, we use the relative error as a metric which is defined as

$$\text{relative error} = \frac{\|\mathcal{X} - \mathcal{X}_{\text{est}}\|_F}{\|\mathcal{X}\|_F}. \tag{20}$$

Note that $\|\bullet\|_F$ denotes the tensor Frobenius norm; $\mathcal{X}$ and $\mathcal{X}_{\text{est}}$ are the original tensor and the estimated tensor, respectively.

Moreover, we use the mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE) and R^2 score as the metric to evaluate the prediction performance with different imputation algorithms,

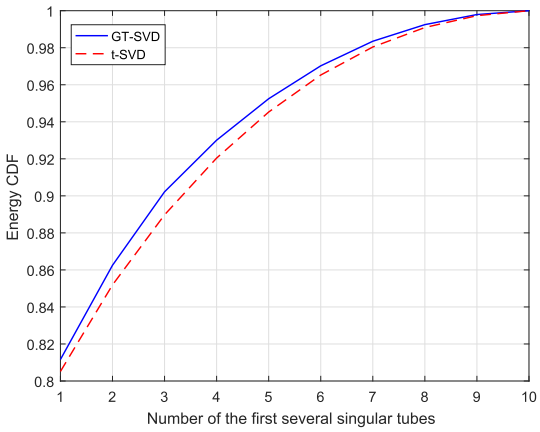


Fig. 8. The energy CDFs of singular tubes captured by t-SVD and GT-SVD.

which are defined as

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|, \tag{21}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}, \tag{22}$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right|, \tag{23}$$

$$R^2 = 1 - \frac{\sum_i(\hat{y}_i - y_i)^2}{\sum_i(\overline{y} - y_i)^2}, \tag{24}$$

where $\hat{y}_i$ is the predicted value, y_i is the ground truth and $\overline{y}$ is the mean value of y.

B. Experiment Setup

To evaluate the performance of the proposed method, we compare the GTC algorithm with the state-of-the-art algorithms, i.e., Bayesian augmented tensor factorization (BATF) [32], fixed-point continuation with approximate SVD (FPCA) [31], TenALS [45], t-SVD based alternating direction method of multipliers (ADMM) [46] and high accuracy low-rank tensor completion (HaLRTC) [47]. BATF is an augmented tensor decomposition model that incorporates generic forms of domain knowledge from transportation systems. It is efficient for missing traffic data imputation. FPCA is capable to solve rank minimization problems of the large-scale matrix by using a homotopic approach with approximate SVD [31]. TenALS is a novel CP decomposition based method that refines the estimations of singular vectors iteratively through alternating minimization. T-SVD based ADMM exploits t-SVD to characterize the information and structure complexity of multilinear data. HaLRTC is able to solve convex function with high accuracy by applying ADMM. Our source codes are available online.[1]

C. Effectiveness Evaluation for GT-SVD

We evaluate the effectiveness of GT-SVD on capturing the low-tubal-rank property by using PeMS08 dataset.

[1] https://github.com/summerdenglei/GTC

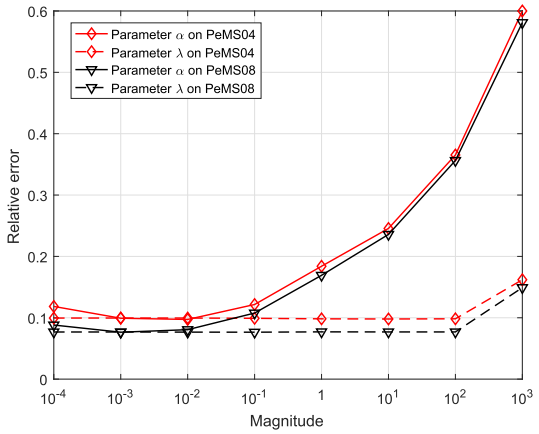


Fig. 9. Sensitivities with respect to α and λ on different datasets at sampling rate $= 0.3$.

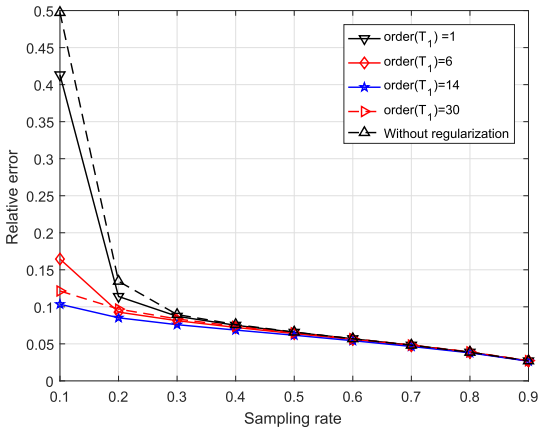


Fig. 10. Performance comparison for choosing Toeplitz matrices with different orders on PeMS08.

Experiments are carried out to investigate the energy distributions of singular tubes by GT-SVD and t-SVD. In particular, a tensor can be decomposed into three components: $\mathcal{U}$, $\mathcal{S}$ and $\mathcal{V}$ by GT-SVD or t-SVD, and the number of non-zero singular tubes of $\mathcal{S}$ are defined as the tubal-rank. The energy of each non-zero singular tube is defined as its ℓ_2 norm. Normally, most energy of the tensor data can be captured by the first several singular tubes of the f-diagonal tensor $\mathcal{S}$. We use the traffic data from PeMS08 within 10 time intervals on 10 days and construct a graph-tensor of size $10 \times 10 \times 170$ for experiments. Fig. 8 plots the energy cumulative distribution functions (CDFs) of singular tubes of $\mathcal{S}$ by GT-SVD and t-SVD respectively. We can observe that more energy of the traffic graph-tensor can be captured by GT-SVD than t-SVD with the same number of the first several singular tubes. The results also imply that the low-rank structure of traffic data can be better captured by GT-SVD than t-SVD.

D. Performance Impacted by Parameter Settings

Moreover, there are several input parameters (i.e., $\boldsymbol{r}^0$, α and λ) are considered in our algorithm. $\boldsymbol{r}^0$ is the initial multi-rank; α and λ allow turnable tradeoffs between the regularization and the importance of rank. These parameters significantly impact recovery performance. Then we conduct experiments to investigate the performance impacted by these parameter settings.

In the real-world scenario, it might be difficult to find the optimal parameters for a real dataset. Hence it is necessary to investigate the impacts of parameter settings and develop an algorithm that is not so sensitive to the values of parameters. In fact, it is difficult to find a single parameter setting that is optimal for all datasets. But we can find rough parameter settings that would not be too far away from the optimal performance.

We now discuss the parameter settings of the initial multi-rank $\boldsymbol{r}^0$, α and λ. The initial multi-rank $\boldsymbol{r}^0$ can be simply set to $\boldsymbol{r}^0 = [1, \cdots, 1] \in \mathbb{R}^{n_3}$ since the rank adjusting strategy can estimate the true rank. Meanwhile, we also conduct experiments to explore the sensitivities of the parameters α and λ to GTC. Fig. 9 presents the performances of GTC under different parameter magnitude at a low random sampling rate (i.e., $S_p = 0.3$). We can observe that the relative errors with different magnitudes of λ are hardly changed on both PeMS08 and PeMS04 datasets (except $\lambda > 10^2$). For parameter α, the best performance can be achieved when $10^{-3} \leq \alpha \leq 10^{-2}$ on both PeMS08 and PeMS04 datasets. We also conduct similar experiments at high sampling rates (i.e., $S_p > 0.5$) and obtain the same conclusion. Then we set $\alpha = 0.001$ and $\lambda = 10$ in our experiments. The experimental results demonstrate that such a parameter setting is generally optimal for different datasets.

E. Performance Impacted by Regularization

As mentioned before, we utilize Toeplitz matrices to represent the temporal smoothness of traffic data and further construct regularization to improve recovery performance. We investigate the impact of regularization using PeMS08. We first evaluate the performances with and without regularization. The experiments without regularization are carried out by replacing the Toeplitz matrices $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ by zero matrices of the same sizes. As presented in Fig. 10, the recovery performance without regularization is not satisfactory.

Furthermore, since traffic data exhibits different temporal smoothness on adjacent days and at adjacent time intervals, it is reasonable to adopt Toeplitz matrices with different orders to capture the temporal smoothness more flexibly and precisely. Then we compare the traffic data recovery errors by varying the order of the Toeplitz matrix $\boldsymbol{T}_1$ using PeMS08. The results are also shown in Fig. 10. There are two key observations from Fig. 10. Firstly, the regularization significantly improves the recovery performance comparing to the situation without regularization. Thus, it is necessary to exploit temporal regularization for performance improvement. On the other hand, we observe that the order of the Toeplitz matrix $\boldsymbol{T}_1$ also impacts recovery accuracy. A 14th-order Toeplitz matrix (i.e., order $(\boldsymbol{T}_1) = 14$) can describe the temporal smoothness of "time interval" more precisely and achieve the best performance. Similarly, we also conduct experiments to find the optimal order for $\boldsymbol{T}_2$ in a same way. The experimental results demonstrate that a first-order Toeplitz matrix would

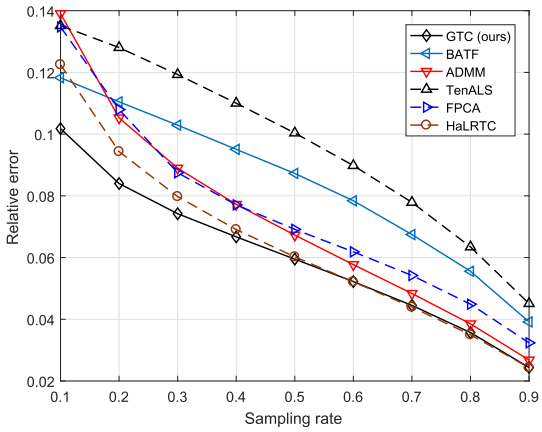


Fig. 11. Performance comparison under random missing pattern on PeMS08.

be proper to capture the smoothness of "day" (i.e., order $(\boldsymbol{T}_2) = 1$).

F. Performance Comparison Under Different Missing Patterns

In real-world scenarios, data missing may not strictly obey uniform and random distribution but structural missing because of sensor malfunctions and communication faults [4], [15]. Therefore, to evaluate the real scenario, we investigate the performance of GTC under three different missing patterns: random missing pattern, temporal missing pattern and spatial missing pattern.

- Random Missing Pattern: Mostly, traffic data is lost uniformly and randomly. Thus we drop the elements in the traffic tensor randomly.
- Temporal Missing Pattern: Alternatively, due to the collection devices being broken or disconnections of the wireless network, traffic data might be lost during a period of time. Consequently, we randomly select sensors (i.e., frontal slices) and drop the data at some successive time intervals from each selected sensor [4], [15].
- Spatial Missing Pattern: Traffic data on some successive sensors might be missing due to the lack of power. We simulate this situation by dropping the data of multiple successive sensors along each freeway [15].

Firstly, we verify our algorithm on the PeMS08 dataset under the random missing pattern. As shown in Fig. 11, we present the recovery errors of different algorithms. Obviously, GTC achieves similar performance with ADMM and HaLRTC at high sampling rates. But thanks to the full utilization of the topological structure of traffic data and the flexible temporal regularization, GTC is able to achieve much better performance at low sampling rates.

For clear illustration, we randomly select a frontal slice of the tensor data for visualization. Fig. 12 shows the schematic diagrams of the recovery result on PeMS08 at 20% randomly missing rate. It can be seen that the recovery data well approximates the original data, while the original structure and characteristics of traffic data are maintained. The results demonstrate that the proposed algorithm achieves efficient recovery performance under the random missing pattern.

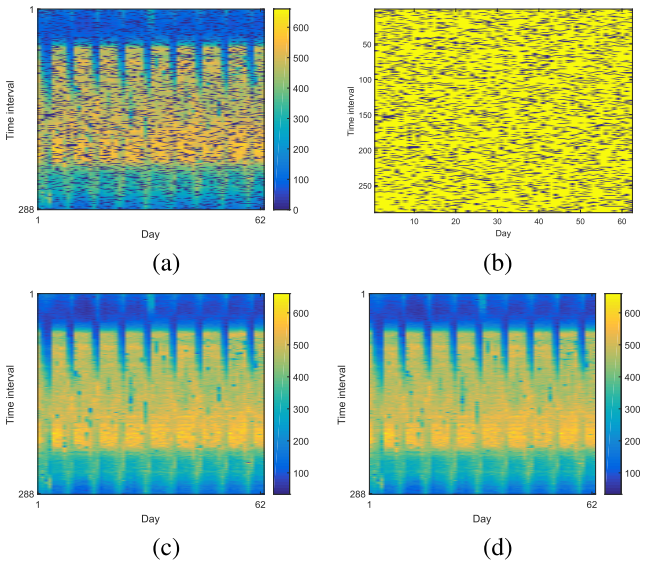


Fig. 12. The recovery results of GTC on PeMS08 at 20% random missing rate. The traffic volume is measured by the color of the grids. (a) The traffic data at 20% random missing rate; the blue grids indicate the positions of missing entries. (b) The locations of the missing entries; the yellow grids indicate the positions of observed entries, and the blue indicates the positions of missing entries. (c) Original data. (d) Recovered data.

Furthermore, we evaluate the proposed algorithm using the PeMS04 dataset. Fig. 13 plots the relative errors of different recovery algorithms at different missing rates on PeMS04. For the random missing pattern, ADMM and FPCA achieve similar performance while HaLRTC achieves close performance to GTC at high sampling rates. Since our work pays more attention to utilize the topological structure and temporal smoothness of the traffic data, the proposed algorithm outperforms other algorithms. In addition, we notice that the recovery performances of PeMS04 are worse than PeMS08. This is because PeMS04 contains more roadways and sensors, which leads to a more complicated network topology than PeMS08. A complicated topology structure implies more unpredictable behaviors and variations in the daily transportation operation, bringing difficulties for feature extraction. In particular, it worsens the low-rank structure of the traffic tensor since our work exploits GT-SVD to capture the spatial correlations and the low-rank property of traffic data.

We also investigate the recovery performance of the proposed algorithm under the temporal missing pattern on both PeMS08 and PeMS04 datasets. We randomly select 50% frontal slices from the traffic tensor and drop the data at some successive time intervals. The sampling rate of each selected frontal slice varies from 0.2 to 0.8. The experimental results are presented in TABLES. III and IV. Obviously, the performances of FPCA are significantly degraded comparing to the random missing pattern. FPCA is a matrix-based completion algorithm, which is vulnerable to structural missing. ADMM and HaLRTC achieve better performances than BATF and TenALS when the sampling rate $S_p > 0.2$. The GTC algorithm outperforms the other algorithms at all sampling rates since these algorithms cannot make full use of the

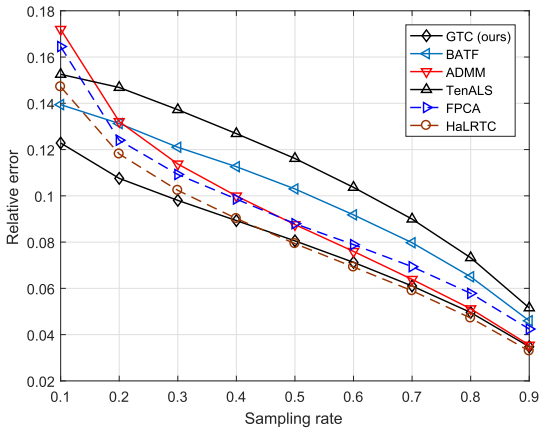


Fig. 13. Performance comparison under random missing pattern on PeMS04.

TABLE III

RELATIVE ERRORS AT DIFFERENT SAMPLING RATES UNDER TEMPORAL MISSING PATTERN ON PEMS08

Algorithm\Sampling rate	20%	40%	60%	80%
GTC	**0.054**	**0.040**	**0.025**	**0.017**
BATF	0.081	0.067	0.055	0.039
ADMM	0.077	0.053	0.043	0.029
TenALS	0.092	0.079	0.065	0.048
FPCA	0.310	0.239	0.156	0.074
HaLRTC	0.075	0.049	0.037	0.024

TABLE IV

RELATIVE ERRORS AT DIFFERENT SAMPLING RATES UNDER TEMPORAL MISSING PATTERN ON PEMS04

Algorithm\Sampling rate	20%	40%	60%	80%
GTC	**0.072**	**0.054**	**0.040**	**0.024**
BATF	0.094	0.078	0.064	0.044
ADMM	0.099	0.070	0.053	0.036
TenALS	0.104	0.087	0.080	0.052
FPCA	0.584	0.518	0.437	0.296
HaLRTC	0.095	0.064	0.049	0.039

temporal correlations. Thus, it can be concluded that GTC is more robust against temporal missing comparing to the other algorithms. For a better explanation, we also provide examples of the traffic volume imputation results on a specific sensor comparing with the ground truth data. Fig. 14 shows an imputation example of traffic volume on the PeMS08 dataset under the temporal missing pattern. We can see that the recovery data of GTC well approximates the ground truth data at a high temporal missing rate (e.g., 80%).

We further evaluate the performance of the proposed algorithm under the spatial missing pattern on both PeMS08 and PeMS04. More specifically, we drop the data of some sequential sensors along each freeway and vary the sampling rate from 0.1 to 0.9. Note that the datasets contain the traffic data of some highways where only several traffic sensors (e.g., less than 4 sensors) are deployed. We remain the data of these sensors, which implies that the actual sampling rates would be a little higher than presented in Figs. 15 and 16. As shown in Figs. 15 and 16, the recovery errors of FPCA are still high. TenALS can achieve excellent performance at very low sampling rates ($S_p \leq 0.1$) on PeMS04, but it cannot keep

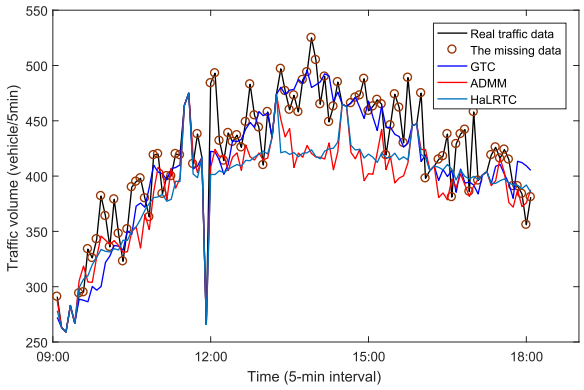


Fig. 14. An illustration for traffic volume imputation on the PeMS08 dataset under the temporal missing pattern at sampling rate $S_p = 0.2$.

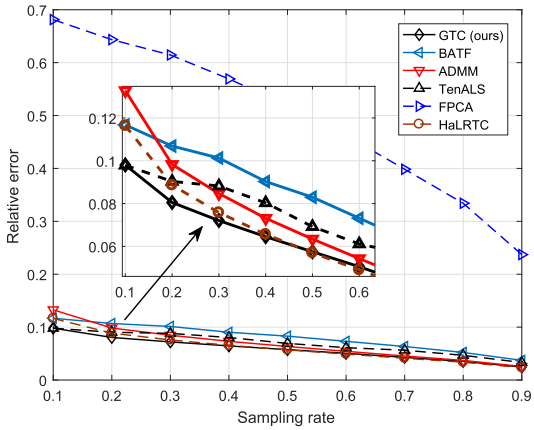


Fig. 15. Performance comparison under spatial missing pattern on PeMS08.

in the lead as the sampling rate increases. HaLRTC achieves very close performance with GTC at high sampling rates (i.e., $S_p \geq 0.5$). Overall, the proposed GTC algorithm still holds the lead at the other sampling rates since it fully exploits the spatiotemporal correlations.

G. Traffic Data Prediction Performance Impacted by Different Imputation Algorithms

In the ITS, traffic data prediction is also an important function for traffic management. Traffic flow prediction relies on the complete historical data and prediction performance will degrade significantly when the training data is incomplete [7]. Thus a common way for traffic prediction with missing values is to impute the missing values firstly and then apply prediction models on the imputed data [48]. Although existing methods like GRU-D [48] can predict traffic with missing data, it is still desired to investigate the prediction performance impacted by different imputation algorithms.

We carry out experiments on the PeMS08 dataset at a 40% random missing rate as an example. We assume that 40% of entries are abandoned randomly from the traffic data. In the training and testing processes, the length of the input time series is set to be 10, and the predicted data length is set to be 1. In addition, we use 70% traffic data for model training, 20% data for validation, and 10% data for test.

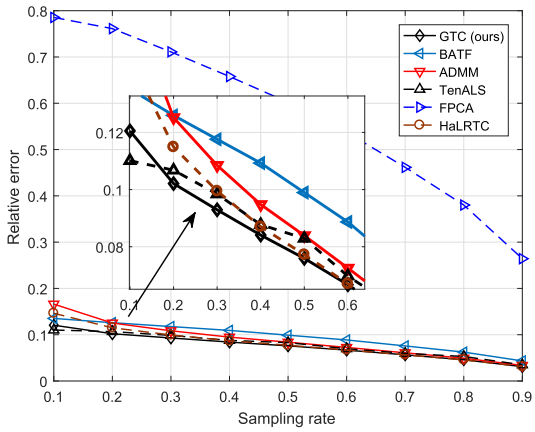


Fig. 16. Performance comparison under spatial missing pattern on PeMS04.

TABLE V

PREDICTION PERFORMANCE COMPARISON OF DIFFERENT METHODS ON PEMS08 DATASET AT SAMPLING RATE $S_p = 0.6$

	MAE	RMSE	MAPE	R^2
GTC with GRU	**15.23**	**22.85**	**10.22%**	**0.9754**
ADMM with GRU	15.65	23.37	10.67%	0.9743
BATF with GRU	16.86	25.65	11.67%	0.9690
GRU-D	20.07	28.98	17.06%	0.9612
TenALS with GRU	31.11	42.46	33.52%	0.9152

Two methods are considered for evaluating the prediction performance:

- GRU-D: The GRU-D model would be trained and tested by using incomplete traffic data.
- Imputation algorithms with GRU: We firstly recover the missing traffic data by using the imputation algorithms (e.g., GTC, ADMM, BATF and TenALS), respectively. Then we apply the traditional gated recurrent unit (GRU) model for traffic data prediction by using the recovered traffic data.

The experimental results are presented in TABLE. V. There are several key observations: 1) Comparing the prediction performance of GRU-D and "GTC with GRU", we can see that "GTC with GRU" improves the traffic data prediction performance significantly over the GRU model, owing to the excellent imputation performance of GTC; 2) Comparing the prediction performance of GRU as the prediction model with different imputation algorithms, we can observe that the prediction accuracy would be impacted by the imputation accuracy. The more accurate the imputation performance is, the better prediction performance can be achieved (the corresponding imputation performance of different algorithms is presented in Fig. 11 when the sampling rate $S_p = 0.6$); 3) The prediction performance of "TenALS with GRU" is worse than GRU-D, implying that the simple GRU model cannot achieve satisfactory prediction accuracy if the imputation results are not good. Overall, the GTC model can provide more accurate imputation performance than the competitive algorithms, which further improves the traffic data prediction performance.

VIII. CONCLUSION

In this paper, we propose to apply a novel graph-tensor factorization framework (GT-SVD) for mining the spatial information of traffic data effectively. A GT-SVD based graph spectral regularized tensor completion algorithm (GTC) was further developed for missing traffic data imputation. Since traffic data shows different smoothness on "time interval" and "day" dimensions, flexible temporal regularization with different orders of Toeplitz matrices was constructed for capturing the temporal smoothness precisely. The experimental results on real-world datasets demonstrate that our algorithm can exploit the spatiotemporal correlations of traffic data more fully and outperforms the state-of-the-art methods under different missing patterns.

There are still some avenues for our future research. For example, it is still challenging to achieve real-time traffic data estimation for the proposed iterative optimization algorithm. In contrast, a deep learning based approach is able to perform real-time traffic data imputation. A promising direction is to combine these two kinds of approaches by unfolding the iterative algorithm into a deep neural network, which considers both the advantages of iterative algorithms and deep learning (e.g., mathematical interpretability and real-time performance). Alternatively, it is also expected to achieve reliable real-time performance by adopting high-performance GPU primitives for graph-tensor operations [49].

APPENDIX A
PROOF OF LEMMA 2

Proof: Suppose $\text{rank}_m(\mathcal{A}) = \boldsymbol{k}$ and $\hat{k} = \max(\boldsymbol{k}_1, \cdots, \boldsymbol{k}_{n_3})$, we can decompose $\overline{\boldsymbol{A}}^{(i)}$ into matrix product $\overline{\boldsymbol{A}}^{(i)} = \hat{\boldsymbol{G}}^{(i)} \hat{\boldsymbol{H}}^{(i)}$ since whose rank is $\boldsymbol{k}_i$, where $\hat{\boldsymbol{G}}^{(i)} \in \mathbb{C}^{n_1 \times \boldsymbol{k}_i}$ and $\hat{\boldsymbol{H}}^{(i)} \in \mathbb{C}^{\boldsymbol{k}_i \times n_2}$ denote the i-th block-diagonal matrices of $\hat{\boldsymbol{G}} \in \mathbb{C}^{n_1 n_2 \times (\sum_{i=1}^{n_3} \boldsymbol{k}_i)}$ and $\hat{\boldsymbol{H}} \in \mathbb{C}^{(\sum_{i=1}^{n_3} \boldsymbol{k}_i) \times n_2 n_3}$ respectively, and $\text{rank}(\hat{\boldsymbol{G}}^{(i)}) = \text{rank}(\hat{\boldsymbol{H}}^{(i)}) = \boldsymbol{k}_i$. We set $\overline{\boldsymbol{G}}^{(i)} = [\hat{\boldsymbol{G}}^{(i)}, \boldsymbol{0}] \in \mathbb{C}^{n_1 \times \hat{k}}$ and $\overline{\boldsymbol{H}}^{(i)} = [\hat{\boldsymbol{H}}^{(i)}, \boldsymbol{0}] \in \mathbb{C}^{\hat{k} \times n_2}$, where $\overline{\boldsymbol{G}}^{(i)}$ and $\overline{\boldsymbol{H}}^{(i)}$ are the i-th block-diagonal matrices of $\overline{\boldsymbol{G}} \in \mathbb{C}^{n_1 n_3 \times \hat{k} n_3}$ and $\overline{\boldsymbol{H}} \in \mathbb{C}^{\hat{k} n_3 \times n_2 n_3}$, respectively. Thus, we can write $\overline{\boldsymbol{F}} = \hat{\boldsymbol{G}} \hat{\boldsymbol{H}} = \overline{\boldsymbol{G}}\,\overline{\boldsymbol{H}}$. According to Lemma 1, We know that $\overline{\boldsymbol{E}} = \overline{\boldsymbol{X}}\,\overline{\boldsymbol{Y}}$ is equivalent to $\mathcal{E} = \mathcal{X} *_g \mathcal{Y}$. Then we have $\mathcal{F} = \mathcal{G} *_g \mathcal{H}$, where $\mathcal{G} \in \mathbb{R}^{n_1 \times \hat{k} \times n_3}$ and $\mathcal{H} \in \mathbb{R}^{\hat{k} \times n_2 \times n_3}$ are two graph-tensors with smaller sizes and $\text{rank}_g(\mathcal{G}) = \text{rank}_g(\mathcal{H}) = \hat{k}$.

For property (ii), suppose $\text{rank}_m(\mathcal{B}) = \boldsymbol{b}$ and $\text{rank}_g(\mathcal{B}) = \hat{b}$, where $\boldsymbol{b}_i = \text{rank}(\overline{\boldsymbol{B}}^{(i)}), i = 1, \cdots, n_3$, and $\hat{b} = \max(\boldsymbol{b}_1, \cdots, \boldsymbol{b}_{n_3})$. Let $\mathcal{Z} = \mathcal{B} *_g \mathcal{C}$. Similarly, assume that $\text{rank}_m(\mathcal{C}) = \boldsymbol{c}$, $\text{rank}_g(\mathcal{C}) = \hat{c}$; $\text{rank}_m(\mathcal{Z}) = \boldsymbol{z}$, $\text{rank}_g(\mathcal{Z}) = \hat{z}$. Note that given two matrices $\boldsymbol{M} \in \mathbb{C}^{n_5 \times n_6}$ and $\boldsymbol{N} \in \mathbb{C}^{n_6 \times n_7}$, we have $\text{rank}(\boldsymbol{MN}) \leq \min(\text{rank}(\boldsymbol{M}), \text{rank}(\boldsymbol{N}))$. Then we have $\boldsymbol{z}_i = \text{rank}(\overline{\boldsymbol{Z}}^{(i)}) = \text{rank}(\overline{\boldsymbol{B}}^{(i)}\overline{\boldsymbol{C}}^{(i)}) \leq \min(\text{rank}(\overline{\boldsymbol{B}}^{(i)}), \text{rank}(\overline{\boldsymbol{C}}^{(i)})) = \min(\boldsymbol{b}_i, \boldsymbol{c}_i)$. Furthermore, we can write $\hat{z} = \max(\boldsymbol{z}_1, \cdots, \boldsymbol{z}_{n_3}) \leq \min(\hat{b}, \hat{c})$. Then we can obtain $\text{rank}_g\left(\mathcal{B} *_g \mathcal{C}\right) \leq \min\left(\text{rank}_g\left(\mathcal{B}\right), \text{rank}_g\left(\mathcal{C}\right)\right)$. ■

REFERENCES

[1] M. Zhu, X.-Y. Liu, and X. Wang, "An online ride-sharing path-planning strategy for public vehicle systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 616–627, Feb. 2019.

[2] M. Zhu *et al.*, "Public vehicles for future urban transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3344–3353, Dec. 2016.

[3] B. Płaczek, "Selective data collection in vehicular networks for traffic control applications," *Transp. Res. C, Emerg. Technol.*, vol. 23, pp. 14–28, Aug. 2012.

[4] L. Qu, L. Li, Y. Zhang, and J. Hu, "PPCA-based missing data imputation for traffic flow volume: A systematical approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 512–522, Sep. 2009.

[5] X.-Y. Liu and X. Wang, "LS-decomposition for robust recovery of sensory big data," *IEEE Trans. Big Data*, vol. 4, no. 4, pp. 542–555, Dec. 2018.

[6] B. Smith, W. Scherer, and J. Conklin, "Exploring imputation techniques for missing data in transportation management systems," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1836, no. 1, pp. 132–142, Jan. 2003.

[7] H. Tan *et al.*, "A tensor-based method for missing traffic data completion," *Transp. Res. C, Emerg. Technol.*, vol. 28, pp. 15–27, Mar. 2013.

[8] I. Laña, I. Olabarrieta, M. Velez, and J. Del Ser, "On the imputation of missing data for road traffic forecasting: New insights and novel techniques," *Transp. Res. C, Emerg. Technol.*, vol. 90, pp. 18–33, May 2018.

[9] Z. Cui, R. Ke, Z. Pu, X. Ma, and Y. Wang, "Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction," *Transp. Res. C, Emerg. Technol.*, vol. 115, Jun. 2020, Art. no. 102620.

[10] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, Apr. 2017.

[11] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, p. 1501, Jun. 2017.

[12] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 922–929.

[13] X. Chen, Y. Cai, Q. Ye, L. Chen, and Z. Li, "Graph regularized local self-representation for missing value imputation with applications to on-road traffic sensor data," *Neurocomputing*, vol. 303, pp. 47–59, Aug. 2018.

[14] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2012.

[15] Y. Wang, Y. Zhang, X.-L. Piao, H. Liu, and K. Zhang, "Traffic data reconstruction via adaptive spatial-temporal correlations," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1531–1543, Aug. 2018.

[16] L. Kong *et al.*, "Data loss and reconstruction in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2818–2828, Nov. 2014.

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

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

[19] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices (extended version)," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 662–676, Jun. 2012.

[20] S. Tak, S. Woo, and H. Yeo, "Data-driven imputation method for traffic data in sectional units of road links," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1762–1771, Jun. 2016.

[21] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 62, pp. 21–34, Jun. 2016.

[22] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 9, 2020, doi: 10.1109/TITS.2020.2997352.

[23] J. J. Q. Yu and J. Gu, "Real-time traffic speed estimation with graph convolutional generative autoencoder," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3940–3951, Oct. 2019.

[24] H. Li, M. Li, X. Lin, F. He, and Y. Wang, "A spatiotemporal approach for traffic data imputation with complicated missing patterns," *Transp. Res. C, Emerg. Technol.*, vol. 119, Oct. 2020, Art. no. 102730.

[25] M. Zhong, P. Lingras, and S. Sharma, "Estimation of missing traffic counts using factor, genetic, neural, and regression techniques," *Transp. Res. C, Emerg. Technol.*, vol. 12, no. 2, pp. 139–166, Apr. 2004.

[26] Z. Li, H. Zheng, and X. Feng, "3D convolutional generative adversarial networks for missing traffic data completion," in *Proc. 10th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2018, pp. 1–6.

[27] X.-Y. Liu, S. Aeron, V. Aggarwal, and X. Wang, "Low-tubal-rank tensor completion using alternating minimization," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1714–1737, Mar. 2020.

[28] X. Chen, Z. He, and J. Wang, "Spatial-temporal traffic speed patterns discovery and incomplete data recovery via SVD-combined tensor decomposition," *Transp. Res. C, Emerg. Technol.*, vol. 86, pp. 59–77, Jan. 2018.

[29] R. Du, C. Chen, B. Yang, and X. Guan, "VANET based traffic estimation: A matrix completion approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 30–35.

[30] 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 Trans. Netw.*, vol. 28, no. 6, pp. 2797–2808, Dec. 2020.

[31] M. T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet, "Matrix and tensor based methods for missing data estimation in large traffic networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1816–1825, Jul. 2016.

[32] X. Chen, Z. He, Y. Chen, Y. Lu, and J. Wang, "Missing traffic data imputation and pattern discovery with a Bayesian augmented tensor factorization model," *Transp. Res. C, Emerg. Technol.*, vol. 104, pp. 66–77, Jul. 2019.

[33] X. Chen, J. Yang, and L. Sun, "A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation," *Transp. Res. C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102673.

[34] X. Chen, Z. He, and L. Sun, "A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation," *Transp. Res. C, Emerg. Technol.*, vol. 98, pp. 73–84, Jan. 2019.

[35] A. Baggag *et al.*, "Learning spatiotemporal latent factors of traffic via regularized tensor factorization: Imputing missing values and forecasting," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2573–2587, Jun. 2021.

[36] Z. Cui, L. Lin, Z. Pu, and Y. Wang, "Graph Markov network for traffic forecasting with missing data," *Transp. Res. Part C: Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102671.

[37] X.-Y. Liu and X. Wang, "Fourth-order tensors with multidimensional discrete transforms," 2017, *arXiv:1705.01576*. [Online]. Available: http://arxiv.org/abs/1705.01576

[38] F. Grassi, A. Loukas, N. Perraudin, and B. Ricaud, "A time-vertex signal processing framework: Scalable processing and meaningful representations for time-series on graphs," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 817–829, Feb. 2018.

[39] Q. Sun, M. Yan, D. Donoho, and S. Boyd, "Convolutional imputation of matrix networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 11, 2018, pp. 7663–7672.

[40] E. Kernfeld, M. Kilmer, and S. Aeron, "Tensor–tensor products with invertible linear transforms," *Linear Algebra Appl.*, vol. 485, pp. 545–570, Nov. 2015.

[41] E. Newman, L. Horesh, H. Avron, and M. Kilmer, "Stable tensor neural networks for rapid deep learning," 2018, *arXiv:1811.06569*. [Online]. Available: http://arxiv.org/abs/1811.06569

[42] F. Rempe, G. Huber, and K. Bogenberger, "Spatio-temporal congestion patterns in urban traffic networks," *Transp. Res. Procedia*, vol. 15, pp. 513–524, Jan. 2016.

[43] P. Zhou, C. Lu, Z. Lin, and C. Zhang, "Tensor factorization for low-rank tensor completion," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1152–1163, Mar. 2018.

[44] Y. Xu, R. Hao, W. Yin, and Z. Su, "Parallel matrix factorization for low-rank tensor completion," *Inverse Problems Imag.*, vol. 9, no. 2, pp. 208–220, Dec. 2013.

[45] P. Jain and S. Oh, "Provable tensor factorization with missing data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2, Jun. 2014, pp. 1431–1439.

[46] 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. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, vol. 27, no. 3, pp. 3842–3849.

[47] 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.

[48] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Sci. Rep.*, vol. 8, no. 1, pp. 1–12, Apr. 2018.

[49] T. Zhang, W. Kan, and X.-Y. Liu, "High performance GPU primitives for graph-tensor learning operations," *J. Parallel Distrib. Comput.*, vol. 148, pp. 125–137, Feb. 2021.

Lei Deng received the B.Eng. degree in communication engineering from Liaoning Technical University, China, in 2017. He is currently pursuing the Ph.D. degree with Fuzhou University, Fuzhou, China. His research interests include tensor theory and the Internet of Things.

Xiao-Yang Liu (Graduate Student Member, IEEE) received the B.Eng. degree in computer science from the Huazhong University of Science and Technology, China, in 2010, the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University, China, in 2017, and the M.S. degree in electrical engineering from Columbia University in 2018, where he is currently pursuing the second Ph.D. degree with the Department of Electrical Engineering. His research interests include tensor theory, deep learning, non-convex optimization, big data analysis, and the Internet of Things.

Haifeng Zheng (Senior Member, IEEE) received the B.Eng. and M.S. degrees in communication engineering from Fuzhou University, Fuzhou, China, and the Ph.D. degree in communication and information system from Shanghai Jiao Tong University, Shanghai, China. He was a Visiting Scholar with the State University of New York at Buffalo from October 2015 to September 2016. He is currently a Professor with the College of Physics and Information Engineering, Fuzhou University. His research interests include tensor theory, machine learning, the Internet of Things, and wireless networks.

Xinxin Feng (Member, IEEE) received the B.S. and M.S. degrees in communication engineering from the Nanjing University of Science and Technology, China, in 2006 and 2008, respectively, and the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University, China, in 2015. She is currently an Associate Professor with the College of Physics and Information Engineering, Fuzhou University, Fuzhou, China. Her research interests include machine learning, big data analysis, and network economies.

Youjia Chen (Member, IEEE) received the B.S. and M.S. degrees in communication engineering from Nanjing University, Nanjing, China, in 2005 and 2008, respectively, and the Ph.D. degree in wireless engineering from The University of Sydney, Australia, in 2017. From 2008 to 2009, she worked at Alcatel-Lucent Shanghai Bell. She worked with the College of Photonic and Electronic Engineering, Fujian Normal University, China, in August 2009. In 2018, she joined the College of Physics and Information Engineering, Fuzhou University, China. Her current research interests include network traffic flow modeling, wireless caching, and machine learning.