Spatiotemporal Regularized Tucker Decomposition Approach for Traffic Data Imputation

Wenwu Gong, Zhejun Huang, and Lili Yang

***Abstract*—In intelligent transportation systems, traffic data imputation, estimating the missing value from partially observed data is an inevitable and challenging task. Previous studies have yet to fully consider traffic data's multidimensionality and spatiotemporal correlations, which are crucial for traffic data imputation, particularly in high-level missing scenarios. To address this problem, we propose a novel spatiotemporal regularized Tucker decomposition method. To begin with, we convert the traffic data into a third-order tensor. We approximate the traffic tensor under Tucker decomposition using the non-negative factor matrices and sparse core tensor. Notably, we do not need to specify or determine the Tucker rank through matrix nuclear-norm minimization. The l_1-norm of the core tensor characterizes the low rankness, while the manifold regularization and temporal constraint on factor matrices are employed to capture spatiotemporal correlations and improve imputation performance. We use an alternating proximal gradient method with guaranteed convergence to address the proposed model. Numerical experiments show that our proposal outperforms matrix-based and tensor-based baselines on real-world traffic datasets in various missing scenarios.**

***Index Terms*—Traffic data imputation, regularized Tucker decomposition, spatiotemporal constraints, alternating proximal gradient.**

I. Introduction

Traffic data (TD) analysis is vital for road traffic control with intelligent transportation systems (ITS) development and application. For example, the road loop sensors record traffic state, including traffic speed, flow, and occupancy rate; the cars equipped with GPS (internet traffic data) record subjects' movement from an origin to a destination, involving pair, time, and day modes. Both of them contain helpful information for traffic networks and route planning. Unfortunately, the missing data problem is inevitable due to communication malfunctions, transmission distortions, or adverse weather conditions [1]. Consequently, traffic data imputation (TDI) is unavoidable and urgently required in ITS.

Many imputation methods have been proposed to deal with the missing data problem, such as statistical-based methods [2] and deep learning-based [3]. However, these methods either need more interpretability or have low accuracy. Due to spatiotemporal correlation and large-scale structure [4], [5], low-rank tensor completion methods have been well developed. In particular, the low-rank tensor approximation (LRTA) model has been validated to be very effective in TDI [6], [7]. Additionally, the LRTA model has successfully discovered interpretable traffic patterns, as reported by Chen et al. in 2018 [1] and 2019 [8]. The primary idea behind TDI is to characterize spatiotemporal correlations, as highlighted in the previous studies [9], [10]. Therefore, combining low rankness (long-term trends) and local correlations (short-term patterns) in traffic data is crucial in solving the TDI problem.

A. Motivations

This paper aims to capture traffic patterns from partially observed TDs via a factorization model and then use them to estimate the missing value accurately. Because low rankness provides long-term trends for the traffic data, the LRTA-based optimization model (referred to as the Tucker decomposition in this paper) with spatiotemporal regularization is used for the TDI problem. The motivations of our proposed model are three folds:

Firstly, the multidimensional array of TDs contains rich information. For example, traffic speed data in adjacent sensors show similar patterns and present temporal correlation properties [11], [12]. The traffic matrix imputation method is 2-dimensional and cannot handle high missing rates or structure-missing scenarios, ignoring the data's multidimensional nature [8], [13]. Hence, reshaping the original traffic data into a high-order tensor to capture the traffic patterns is essential.

Secondly, it is challenging to minimize the tensor rank. On the one hand, flattening the tensor into a matrix and minimizing the unfolding matrix nuclear norm is computationally time-consuming. On the other hand, rank determination remains the main challenge in using low-rank tensor decomposition models for TDI.

In addition, most previous tensor-based imputation methods only focused on the long-term trends and temporal patterns of TD, which made handling high-level and structured missing scenarios difficult. The Tucker decomposition model preserves the multidimensional nature of the TD and extracts the hidden patterns [14] in a subspace. Thus, Tucker decomposition combined with spatiotemporal constraints in the subspace captures the traffic long term and reflects the spatial and temporal correlations for the TDI problem.

B. Contributions

Though low-rank tensor completion is a hot topic for TDI, the problem is still open and needs to be better addressed. One of the main challenges is developing a low-rank Tucker model without a predefined rank that can accurately capture

W. Gong (ORCID: 0000-0002-8019-0582) is a Ph.D. student at the Department of Statistics and Data Science, Southern University of Science and Technology, Shenzhen, 518055, China.

Zhejun Huang is with the Department of Statistics and Data Science, Southern University of Science and Technology, Shenzhen, 518055, China.

Lili Yang is with the Department of Statistics and Data Science, Southern University of Science and Technology, Shenzhen, 518055, China.

long-term trends. Considering the short-term patterns of the TD, another challenge remains to encode the spatiotemporal correlations and enhance the imputation performance.

This paper proposes an innovative enhanced low-rank Tucker decomposition model called Spatiotemporal Regularized Tucker Decomposition (STRTD) for the TDI problem. We summarize the main contributions as follows:

1) To better capture long-term traffic trends, we transform the matrix-based data into a 3rd-order tensor, which provides richer spatial and temporal information.
2) We propose STRTD to characterize traffic patterns in TDs. The proposed model promotes long-term trends by contrasting the Tucker core tensor's sparsity and non-negative factor matrices without a predefined rank. Additionally, STRTD employs manifold regularization and temporal constraint to characterize the short-term patterns and enhance the model performance.
3) The STRTD model is an additive, non-convex, non-smooth optimization problem. We use the alternating proximal gradient (APG) method to transform the objective function into multiple solvable subproblems and iterate alternately to find the critical point.
4) We verify the importance of spatiotemporal constraints in STRTD on two real-world TDs. A comprehensive comparative study with baselines is also conducted to demonstrate the effectiveness of STRTD. With the free-hyperparameters tuning, we show that STRTD performs better in real-world traffic imputation problems under different types of missing scenarios.

We organize the rest of the paper as follows. Section II discusses the related work on TDI. Section III introduces the notations and defines the TDI problem. Section IV proposes the spatiotemporal constraints and model framework. In Section V, we present an algorithm that guarantees convergence. We evaluate the performance of our proposal on extensive experiments and compare them with some baseline approaches in Section VI. The last section concludes this paper and presents future work.

II. Related Work

Numerous time series imputation methods have been developed in the last two decades, especially for TDI [15]. From the model-building perspective, these methods can be divided into machine- and low-rank learning-based. Because TD has spatial similarity and temporal variation characteristics, many studies have proven that the low rankness assumption combined with the spatiotemporal information method performs better than other existing methods for TDI [5], [10]. So, the low-rank tensor learning methods for TDI are discussed in detail in the following section.

A. Low rankness

The low-rank property, which depicts the inherent correlations in real-world datasets, is an essential and significant assumption in the completion problem. Candes et al. [16] proposed the trace norm to estimate the missing matrix data regarding low-rank minimization. Liu et al. [17] extended the matrix case to the tensor and proposed a tensor nuclear norm for the image inpainting problem. Ran et al. [18] applied the low-rank tensor nuclear norm minimization method to recover the spatiotemporal traffic flow.

To avoid using the computationally expensive singular value decomposition (SVD) in unfolding matrix norm minimization, Tan et al. [19] proposed a Tucker decomposition model based on the truncated singular values of each factor matrix to exploit the low rankness in TD. Furthermore, Yokota et al. [20] showed that the rank increment strategy is sufficient when a lower m-rank approximation initializes the tensor than its target m-rank in Tucker-based completion applications.

A significant difference between these two approximation methods is how they make decisions about low rankness. Compared with rank minimization and its relaxation, on the one hand, the low-rank decomposition model can preserve the tensor structure and avoid the high-cost unfolding matrix SVD [21]. On the other hand, nuclear norm minimization cannot impose spatiotemporal constraints directly on traffic data [12]. The low-rank decomposition model is more appropriate for the TDI task [9], [22], [23].

B. Factorization Model

Low-rank tensor decomposition, a high-order matrix factorization extension, has received increasing attention in spatiotemporal traffic data analysis. On the one hand, considering the sensory traffic matrix data, many papers applied the spatiotemporal Hankel operator to capture the low-rank structure by transforming the original incomplete matrix to a 4th-order tensor [13], [24]. This transformer captures spatiotemporal information in a data-driven manner. However, it is time-consuming and parameter-sensitive. On the other hand, many papers used tensor decomposition models for TDI. For example, Tan et al. [4], [19] proposed a Tucker-based model to estimate missing traffic speeds, and the model results experimentally verified the accuracy of traffic data completion under Tucker-based decomposition. Chen et al. [22] proposed a 3rd-order Bayesian augmented CP decomposition model for traffic data analysis, combining domain knowledge to enhance the imputation performance. However, estimating the exact rank of CP decomposition in practice takes much work. Furthermore, the spatiotemporal constraints can be directly imposed on traffic data in the decomposition model [12]. So, this paper focuses on the Tucker decomposition model for spatiotemporal traffic data imputation.

C. Regularized Tucker Decomposition

Many studies have reported that using a low-rank Tucker decomposition model is insufficient for cases where the missing ratio is high [21], [25], [26]. One of the most popular methods is to add regularization to the Tucker decomposition. Rose et al. [9] proposed a unified low-rank tensor learning framework considering local similarity by constructing a Laplacian regularizer for multivariate data analysis.

Considering the spatiotemporal correlations in traffic data, the constraint-based methods, such as smoothness [25], manifold regularization [12], and temporal regularization [5], have

TABLE I
NOTATIONS

$\mathcal{X}, \mathbf{U}, \alpha$	A tensor, matrix and real value, respectively.
$\mathbb{R}_+^{I_1 \times I_2 \times \cdots \times I_N}$	Set of N-th order non-negative array.
$\mathbf{U} \geq 0$	non-negative matrix $\mathbf{U}$, i.e., $u_{ij} \geq 0, \forall i, j$.
$\mathcal{P}_+(\mathbf{U})$	Operator yielding a non-negative matrix of $u_{ij} = \max(u_{ij}, 0), \forall i, j$.
$\mathcal{S}_\mu(x)$	Shrinkage operator with μ in component-wise.
$\Omega, \bar{\Omega}$	Observed index set and its complement.
$\mathcal{X}_\Omega$	Observed entries supported on the observed index.
$\mathcal{H}$	Tensorization operator.
$\times_n$	Mode-n product.
$\otimes, \odot$	Kronecker product and Hadamard product.
$\|*\|_F$	Frobenius norm.
$\mathbf{X}_{(n)}$	Mode-n unfolding of tensor $\mathcal{X}$.
tr	Matrix trace operator.

been well studied. For example, Pan et al. [21] proposed a sparse enhanced Tucker decomposition model to exploit inherent long-term and short-term information in spatiotemporal traffic data imputation tasks. Besides, Zhang et al. [26] introduced the Toeplitz matrix in the TT models to improve the TD imputation performance. Most approaches require predefined tensor ranks and are designed purely based on spatial or temporal correlation, resulting in low performance in data imputation, especially for structured missing scenarios.

Most papers still need to fully consider the long and short-term patterns simultaneously, i.e., low rankness and the spatiotemporal correlations. The proposed STRTD addresses these properties in a tensor object by reshaping the traffic matrix data in a 3rd-order spatiotemporal tensor form. Then, STRTD exploits the long-term traffic trends using a low-rank Tucker model without a predefined rank and captures the short-term patterns with given spatiotemporal priors, including manifold regularization and temporal constraint. Our experiment results demonstrate that STRTD performs better in two real-world TDs.

III. PRELIMINARIES

We review some related concepts of Tucker decomposition as follows and present all notations used in this paper in Tab. I. Please refer to [6] for more information on the preliminaries.

A. Notations

A tensor is a multidimensional array where the order of the tensor is the number of dimensions, also called the mode. Throughout this paper, we use calligraphy font for tensors, such as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, whose element is denoted as $x_{i_1, i_2, \cdots, i_n}$. The bold uppercase letters for matrices, such as $\mathbf{U} \in \mathbb{R}^{I_1 \times I_2}$, bold lowercase letters for vectors, such as $\mathbf{a} \in \mathbb{R}^{I_1}$, and lower case for scalars, such as α, β.

We denote the mode-n unfolding (i.e., matricization) of an N-order tensor $\mathcal{X}$ by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times \prod_{j \neq n} I_j}$. Based on the matrix Kronecker product $\otimes$, we can represent the Tucker decomposition $\mathcal{X} = \mathcal{G} \times_{n=1}^N \mathbf{U}_n$ by $\mathbf{X}_{(n)} = \mathbf{U}_n \mathbf{G}_{(n)} \mathbf{V}_n^{\mathrm{T}}$, $\mathbf{V}_n = (\mathbf{U}_N \otimes \cdots \otimes \mathbf{U}_{n+1} \otimes \mathbf{U}_{n-1} \otimes \cdots \otimes \mathbf{U}_1)$ and the superscript 'T' represent matrix transpose. We can verify that $\operatorname{vec}(\mathcal{X}) = (\mathbf{U}_{\mathrm{N}} \otimes \cdots \otimes \mathbf{U}_{\mathrm{n}} \otimes \cdots \otimes \mathbf{U}_1) \operatorname{vec}(\mathcal{G}) = \otimes_{\mathrm{n=N}}^1 \mathbf{U}_{\mathrm{n}} \operatorname{vec}(\mathcal{G})$.

Finally, for a given tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and observed index set Ω, we define $\mathcal{X}_\Omega$ as a projector that keeps the nonzero terms and leaves the other values as zero values, i.e.,

$$\mathcal{X}_\Omega := \begin{cases} x_{i_1 i_2 \ldots i_n}, & \text{if } (i_1, i_2, \ldots, i_n) \in \Omega \\ 0, & \text{otherwise.} \end{cases}$$

B. Problem Definition

TD is typically collected from M sensors over J days with I time points. The missing multivariate time series is denoted as $\mathbf{Y}_\Omega \in \mathbb{R}^{M \times IJ}$ with the observed index set Ω, as shown in Fig. 1(a). Chen et al. [5] showed that a low-rank tensor can effectively capture long-term trends in TD and estimate the traffic matrix. Furthermore, the TD tends to be similar along the nearby sensors and correlates at adjacent time points, reflecting short-term patterns [27]. So, this paper introduces the tensorization operator [20] $\mathcal{H}$ to stack one-day traffic sensory data and reshape the TD into a 3rd-order tensor. Then, an enhanced low-rank Tucker decomposition combined with the spatiotemporal constraints model (STRTD) is proposed to capture the long and short-term patterns in TD. Conversely, the inverse operator $\hat{\mathbf{Y}} = \mathcal{H}^{-1}(\hat{\mathcal{X}})$ converts the reconstructed tensor into the original traffic matrix and then estimates the missing values.

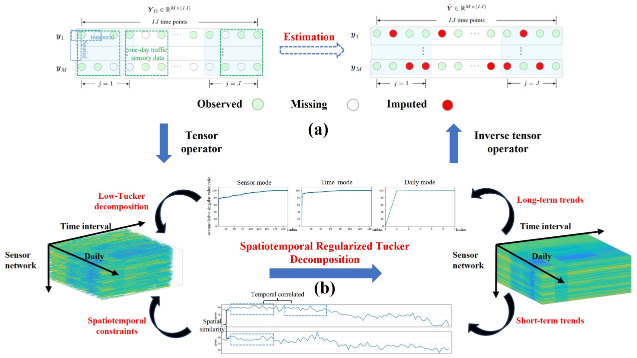

Fig. 1. The proposed STRTD framework for the TDI problem. (a) Matrix representation for TD. (b) Low-rank Tucker imputation based on the 3rd-order traffic tensor.

Mathematically, we illustrate the proposed framework by minimizing the following objective function:

$$\begin{aligned} & \underset{\mathcal{G}; \{\mathbf{U}_n\}}{\operatorname{minimize}} \ \alpha \|\mathcal{G}\|_1 + \beta \ g(\mathbf{U}_n) \\ & s.t. \ \mathcal{X} = \mathcal{G} \times_{n=1}^N \mathbf{U}_n, \ \mathbf{U}_n \geq 0, \quad \mathcal{X}_\Omega = \mathcal{H}(\mathbf{Y}_\Omega), \end{aligned} \tag{1}$$

$g(\cdot)$ is the user-defined spatiotemporal constraint, and α, β are tradeoff parameters to compromise the low rankness and regularization role. Under different missing scenarios, we update $\mathcal{X}$ by the rule (2)

$$\hat{\mathcal{X}} = \mathcal{X}_\Omega + \{\hat{\mathcal{G}} \times_1 \hat{\mathbf{U}}_1 \times \cdots \times_N \hat{\mathbf{U}}_N\}_{\bar{\Omega}}. \tag{2}$$

The main idea of our framework is to propose and study low-rank Tucker approximation for traffic tensor and estimate the traffic matrix by $\hat{\mathbf{Y}} = \mathcal{H}^{-1}(\hat{\mathcal{X}})$.

IV. PROPOSED MODEL

This section describes the formulation of TDI using spatial and temporal constraints in a regularized low-rank Tucker decomposition model. The proposed method involves the sparsity of the Tucker core tensor, non-negative factor matrices, manifold regularization, and temporal constraint.

A. Spatiotemporal Constraints

As mentioned, TD often reflects short-term patterns along the spatial and temporal modes. On the one hand, the similarity between rows of the traffic matrix characterizes the spatial pattern, and the difference operator models the temporal variation [11]. On the other hand, the short-term patterns can be captured by using factor priors in the subspace under Tucker decomposition [28], [29]. In this paper, we address the spatiotemporal correlations relying on the manifold regularization and temporal constraint matrix on factor matrices, which leads to better performance for the TDI problem.

1) Manifold regularization: deals with non-linear data dimension reduction [14], which is used to search the geometric structure of the graph. Since the TD is in a low-dimensional spatial subspace, the similarity between the two sensors also exists in the spatial mode [12]. We first select p nearest neighbors in traffic sensors and use the kernel weighting to determine a similarity matrix, defined as (3)

$$w_{ij} = e^{-\left(\|y_i - y_j\|^2\right)/\sigma^2}, \tag{3}$$

where y_i and y_j are the neighbor nodes along the spatial mode, $\sigma^2 = 1$ denotes the uniform divergence.

Given the matrix $\mathbf{W} \in \mathbb{R}^{I_1 \times I_1} \geq 0$ for traffic spatial mode, we can use (4), the manifold regularization term, to capture the spatial similarity in the subspace.

$$\sum_{i=1}^{I_1} \sum_{j=1}^{I_1} w_{ij} \left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2 = \operatorname{tr}\left(\mathbf{U}^{\mathrm{T}} \mathbf{L} \mathbf{U}\right), \ \mathbf{L} = \mathbf{D} - \mathbf{W}, \tag{4}$$

where $\mathbf{u}_i$ is the column vector of $\mathbf{U}^{\mathrm{T}}$, $\mathbf{D} \in \mathbb{R}^{I_1 \times I_1}$ is a diagonal matrix and $d_{ii} = \sum_{j=1}^{I_1} w_{ij}$, $i = 1, \ldots, I_1$.

2) Temporal constraint: it is to capture the correlations between adjacent time points in the time dimensional [12]. Considering the non-stationary in the temporal dimension, the original traffic data is often correlated at adjacent time points. For adjacent $j-1$th and jth time points in traffic matrix $\mathbf{Y}$, we consider the Toeplitz operator $\mathbf{T}$ defined on the traffic tensor $\mathcal{X}$ to capture temporal variation, i.e., $\|\mathbf{Y}_{\cdot j} - \mathbf{Y}_{\cdot j-1}\|_F^2 = \|\mathcal{X} \times_n \mathbf{T}\|_F^2$. Note that

$$\begin{aligned} \|\mathcal{X} \times_n \mathbf{T}\|_F^2 &= \|\mathcal{G} \times_1 \mathbf{U}_1 \cdots \times_n (\mathbf{T}\mathbf{U}_n) \times_{n+1} \cdots \times_N \mathbf{U}_N\|_F^2 \\ &= \left\|(\mathbf{T}\mathbf{U}_n)\left(\mathcal{G} \times_{p=1, p\neq n}^{N} \mathbf{U}_p\right)\right\|_F^2 \\ &\leq \|\mathbf{T}\mathbf{U}_n\|_F^2 \left\|\mathcal{G} \times_{p=1, p\neq n}^{N} \mathbf{U}_p\right\|_F^2 \\ &\leq \text{const.} \left\|\mathbf{T}\mathbf{U}_n\right\|_F^2 . \end{aligned} \tag{5}$$

Consequently, we use $\|\mathbf{T}\mathbf{U}\|_F^2$ to characterize the temporal correlation of traffic tensor in our proposal.

B. Spatiotemporal Regularized Tucker Decomposition Model

Let $\mathcal{X}^0 \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be the missing traffic tensor. Based on the model (1) and the aforementioned spatiotemporal constraints, we consider the following optimization problem

$$\begin{aligned} &\underset{\mathcal{G};\{\mathbf{U}_n\};\mathcal{X}}{\operatorname{minimize}} \ \mathbb{F}(\mathcal{G}, \{\mathbf{U}_n\}, \mathcal{X}) \\ &\quad \triangleq \{\frac{1}{2}\left\|\mathcal{X} - \mathcal{G} \times_{n=1}^{N} \mathbf{U}_n\right\|_F^2 + \alpha\|\mathcal{G}\|_1 + \\ &\quad \sum_{n=1}^{K} \frac{\beta_n}{2} \operatorname{tr}\left(\mathbf{U}_n^{\mathrm{T}} \mathbf{L}_n \mathbf{U}_n\right) + \sum_{n=K+1}^{N} \frac{\beta_n}{2} \|\mathbf{T}_n \mathbf{U}_n\|_F^2\} \\ &\quad s.t. \ \mathbf{U}_n \in \mathbb{R}_+^{I_n \times I_n}, n = 1, \ldots, N \text{ and } \mathcal{X}_\Omega = \mathcal{X}_\Omega^0, \end{aligned} \tag{6}$$

where α, β_n are positive penalty parameters, K represents the numbers of spatial modes, $\mathbf{L}_n$ captures the spatial similarity, and $\mathbf{T}_n$ encodes the temporal variation. By imposing non-negativity constraints on the factor matrix, the Tucker core tensor becomes sparser [30] and leads to a more intuitive explanation of traffic patterns [31]. We name the model in (6) as the Spatiotemporal Regularized Tucker Decomposition (STRTD) method, simultaneously exploiting the long and short-term characteristics of sensory traffic matrix data.

Remark: The Tucker components' constraint assures that our proposal is well-defined. On the one hand, if all penalty parameters and non-negative vanish, there are product combinations $\{\lambda_{n+1}\}$ such that $\{\lambda_1\mathcal{G}, \lambda_2\mathbf{U}_1, \cdots, \lambda_{n+1}\mathbf{U}_n\}$ does not change the value of (6). Hence, the low-rank Tucker approximation may not admit a solution. On the other hand, the spatiotemporal constraints imply the gradients of (6) are Lipschitz continuous and have bounded Lipschitz constant under proximal linear operators (See ***Proposition 1*** and ***Proposition 2***), which guarantee the solution set is nonempty.

V. SOLVING STRTD MODEL

To solve the complicated optimization problems (6), we use the alternating proximal gradient (APG) method to transform the objective function into multiple solvable subproblems and iterate alternately to find the critical point. Furthermore, we present the convergence results for our proposed STRTD model.

A. Optimization for the STRTD Model

The STRTD is the regularized block multi-convex optimization problem [32], where we can use the prox-linear operator to solve that. The details are shown in the Appendix A.

Firstly, we transform the (6) in mode-n unfolding, and the factor matrices subproblems are given as the following three types.

- Basic non-negative matrix factorization.

$$\underset{\mathbf{U}_n \geq 0}{\operatorname{minimize}} \ \ell(\mathbf{U}_n) = \frac{1}{2}\left\|\mathbf{X}_{(n)} - \mathbf{U}_n \mathbf{G}_{(n)} \mathbf{V}_n^{\mathrm{T}}\right\|_{\mathrm{F}}^2 \tag{7}$$

where $\mathbf{V}_n = \otimes_{p=N, p\neq n}^{1} \mathbf{U}_p$.

- Manifold regularization on factor matrix.

$$\begin{aligned} \underset{\mathbf{U}_n \geq 0}{\operatorname{minimize}} \ \ell(\mathbf{U}_n) = &\frac{1}{2}\left\|\mathbf{X}_{(n)} - \mathbf{U}_n \mathbf{G}_{(n)} \mathbf{V}_n^{\mathrm{T}}\right\|_{\mathrm{F}}^2 \\ &+ \frac{\beta_n}{2} \operatorname{tr}\left(\mathbf{U}_n^{\mathrm{T}} \mathbf{L}_n \mathbf{U}_n\right) \end{aligned} \tag{8}$$

where $\mathbf{L}_n = \mathbf{D}_n - \mathbf{W}_n$ represents the Laplacian matrix.

- Temporal constraint on factor matrix.

$$\begin{aligned}\underset{\mathbf{U}_n \geq 0}{\operatorname{minimize}}\ \ell(\mathbf{U}_n) =& \frac{1}{2}\left\|\mathbf{X}_{(n)} - \mathbf{U}_n \mathbf{G}_{(n)} \mathbf{V}_n^{\mathrm{T}}\right\|_{\mathrm{F}}^2 \\ &+ \frac{\beta_n}{2}\left\|\mathbf{T}_n \mathbf{U}_n\right\|_{\mathrm{F}}^2\end{aligned} \tag{9}$$

where $\mathbf{T}_n$ is a self-defined temporal constraint matrix.

Proposition 1: Subproblems (7) - (9) are differentiable and convex. The gradients $\nabla_{\mathbf{U}_n}\ell(\mathbf{U}_n)$ are both Lipschitz continuous with the following Lipschitz constant.

$$L_{\mathbf{U}_n} = \begin{cases} \left\|\mathbf{G}_{(n)}\mathbf{V}_n^{\mathrm{T}}\mathbf{V}\mathbf{G}_{(n)}^{\mathrm{T}}\right\|_2 + \beta_n \left\|\mathbf{L}_n\right\|_2, & \text{Manifold} \\ \left\|\mathbf{G}_{(n)}\mathbf{V}_n^{\mathrm{T}}\mathbf{V}\mathbf{G}_{(n)}^{\mathrm{T}}\right\|_2 + \beta_n \left\|\mathbf{T}_n^{\mathrm{T}}\mathbf{T}_n\right\|_2, & \text{Temporal} \\ \left\|\mathbf{G}_{(n)}\mathbf{V}_n^{\mathrm{T}}\mathbf{V}\mathbf{G}_{(n)}^{\mathrm{T}}\right\|_2, & \text{otherwise.} \end{cases}$$

Then, (10) presents the prox-linear operator to solve factor matrices subproblems. The appendix contains the detailed proof of *Proposition 1* and obtains updated rule (14).

$$\hat{\mathbf{U}}_n = \underset{\mathbf{U}_n \geq 0}{\operatorname{argmin}} \left\langle \nabla_{\mathbf{U}_n}\ell(\tilde{\mathbf{U}}_n), \mathbf{U}_n - \tilde{\mathbf{U}}_n \right\rangle + \frac{L_{\mathbf{U}_n}}{2}\|\mathbf{U}_n - \tilde{\mathbf{U}}_n\|_F^2, \tag{10}$$

where $\tilde{\mathbf{U}}_n$ denotes the extrapolated point.

Secondly, we update the subproblem $\mathcal{G}$ using the vectorization optimization problem (11)

$$\begin{aligned}\underset{\mathcal{G}}{\operatorname{minimize}}\ & \frac{1}{2}\left\|\operatorname{vec}(\mathcal{X}) - \otimes_{\mathrm{n=N}}^{1}\mathbf{U}_{\mathrm{n}}\operatorname{vec}(\mathcal{G})\right\|_{\mathrm{F}}^2 + \alpha\|\operatorname{vec}(\mathcal{G})\|_1 \\ &= f(\mathcal{G}) + \alpha\|\operatorname{vec}(\mathcal{G})\|_1.\end{aligned} \tag{11}$$

Proposition 2: $\nabla_{\mathcal{G}} f(\mathcal{G})$ is Lipschitz continuous with the bounded Lipschitz constant $L_{\mathcal{G}} = \left\|\otimes_{n=N}^{1}\mathbf{U}_n^{\top}\mathbf{U}_n\right\|_2 = \prod_{n=1}^{N}\left\|\mathbf{U}_n^{\mathrm{T}}\mathbf{U}_n\right\|_2$.

Guided by *Proposition 2*, we denote the core tensor prox-linear function as (12)

$$\hat{\mathcal{G}} = \underset{\mathcal{G}}{\operatorname{argmin}} \left\langle \nabla_{\mathcal{G}} f(\tilde{\mathcal{G}}), \mathcal{G} - \tilde{\mathcal{G}} \right\rangle + \frac{L_{\mathcal{G}}}{2}\|\mathcal{G} - \tilde{\mathcal{G}}\|_F^2 + \alpha\|\mathcal{G}\|_1, \tag{12}$$

where $\tilde{\mathcal{G}}$ denotes the extrapolated point. Using the soft-thresholding operator [33], the core tensor updating rule is shown as (13), and the Appendix presents the detailed proof.

Thirdly, considering the spatiotemporal priors are constrained on factor matrices $\{\mathbf{U}_n\}$, our proposed algorithm applies the order of $\mathcal{G}, \mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_N$ for algorithm design. Suppose the current iteration is k-th step, we update the core tensor $\mathcal{G}^k$

$$\mathcal{G}^{k+1} = \mathcal{S}_{\frac{\alpha}{L_{\mathcal{G}}^k}}\left(\tilde{\mathcal{G}}^k - \frac{1}{L_{\mathcal{G}}^k}\nabla_{\mathcal{G}} f\left(\tilde{\mathcal{G}}^k\right)\right), \tag{13}$$

where $\tilde{\mathcal{G}}^k$ is given by (15) and $\mathcal{S}_\mu(\mathcal{G})$ is a soft-thresholding operator. Also, the factor matrices $\{\mathbf{U}_n^k\}$ is updated by

$$\mathbf{U}_n^{k+1} = \mathcal{P}_+\left(\tilde{\mathbf{U}}_n^k - \frac{1}{L_{\mathbf{U}_n}^k}\nabla_{\mathbf{U}_n}\ell\left(\tilde{\mathbf{U}}_n^k\right)\right), \tag{14}$$

where $\tilde{\mathbf{U}}_n^k$ is given by (16) and $\mathcal{P}_+(\mathbf{U})$ is a mapping function that projects the negative entries of $\mathbf{U}$ into zeros.

Technically, we propose an initial strategy where the $\{\mathbf{U}_n\}$ is generated randomly and then processed by normalization [30]. We conclude that it can reduce the low-rank approximation errors in our experiments. Furthermore, we speed up Algorithm 1 with (15) and (16), which is updated by a parameterized iterative shrinkage-thresholding scheme [34], given by $t^k = \frac{0.8+\sqrt{4(t^{k-1})^2+0.8}}{2}$, $t^0 = 1$.

$$\tilde{\mathcal{G}}^k = \mathcal{G}^k + \omega_k\left(\mathcal{G}^k - \mathcal{G}^{k-1}\right), \text{ for } k \geq 1. \tag{15}$$

$$\tilde{\mathbf{U}}_n^k = \mathbf{U}_n^k + \omega_k\left(\mathbf{U}_n^k - \mathbf{U}_n^{k-1}\right), \text{ for } k \geq 1. \tag{16}$$

$$\omega_k = \min\{\frac{t^{k-1}-1}{t^k}, 0.999\sqrt{\frac{L_{\mathcal{G}}^{k-1}}{L_{\mathcal{G}}^k}}(\text{or } \sqrt{\frac{L_{\mathbf{U}}^{k-1}}{L_{\mathbf{U}}^k}})\}, \tag{17}$$

Furthermore, we use the control rule [30] to re-update tensor $\mathcal{X}^k$ at the end of iteration k

$${\mathcal{X}^{k+1}}_{\Omega} = {\mathcal{X}^0}_{\Omega} + \gamma({\mathcal{X}^k}_{\Omega} - {\mathcal{Z}^k}_{\Omega}), \quad {\mathcal{X}^{k+1}}_{\bar{\Omega}} = {\mathcal{Z}^k}_{\bar{\Omega}}, \tag{18}$$

where $\mathcal{Z}^k = \mathcal{G}^k \times_{n=1}^{N} \mathbf{U}_n^k$, $\bar{\Omega}$ is the complement set of Ω, and $0 \leq \gamma \leq 1$ is a user defined hyper-parameter. We ensure that the value of $\mathbb{F}\left(\mathcal{G}^k, \{\mathbf{U}_n^k\}\right)$ decreases before re-updating the $\tilde{\mathcal{G}}$, $\{\tilde{\mathbf{U}}_n\}$ and calculate the complete tensor $\hat{\mathcal{X}} = \mathcal{X}_{\Omega}^0 + {\mathcal{Z}^k}_{\bar{\Omega}}$ as the imputed result when (19) is satisfied.

$$\left\|\Omega \odot (\mathcal{Z}^k - \mathcal{X}^0)\right\|_F \left\|\Omega \odot \mathcal{X}^0\right\|_F^{-1} < \text{tol}, \text{ for some } k. \tag{19}$$

Algorithm 1 APG-based solver for the STRTD model

```
Input: Missing traffic tensor $\mathcal{X}^0 \in \mathbb{R}_+^{I_1 \times I_2 \times \cdots \times I_N}$, $\Omega$ containing indices of observed entries, and the parameters $\alpha \geq 0$, $\beta_n \geq 0$, to = $1e^{-4}$, and $K = 300$.
Output: Reconstructed tensor $\hat{\mathcal{X}}$.
construct positive semi-definite similarity matrix $\mathbf{W}_n$ and temporal constraint matrix $\mathbf{T}_n$;
initialize $\mathcal{G}^0, \mathbf{U}_n^0 \in \mathbb{R}_+^{I_n \times I_n}$ $(1 \leq n \leq N)$;
for $k = 1$ to $K$ do
    Optimize $\mathcal{G}$ according to (13);
    for $n = 1$ to $N$ do
        Optimize $\mathbf{U}_n$ using (14);
    end for
    Update $\mathcal{Z}^k$ using (18);
    Whenever $\mathbb{F}\left(\mathcal{G}^k, \mathbf{U}_k\right) < \mathbb{F}\left(\mathcal{G}^{k-1}, \mathbf{U}_{k-1}\right)$, we re-update $\tilde{\mathcal{G}}$, $\{\tilde{\mathbf{U}}_n\}$ using (15) and (16) until stopping conditions (19) are satisfied.
end for
return $\hat{\mathcal{X}}_{\Omega} = {\mathcal{X}^0}_{\Omega}$, $\hat{\mathcal{X}}_{\bar{\Omega}} = {\mathcal{Z}^k}_{\bar{\Omega}}$.
```

Algorithm 1 is an APG-based updating procedure with closed-form solutions for the proposed (6) problem, which improves the algorithm's efficiency.

B. Convergence Analysis

Since the STRTD model is a non-convex problem, we demonstrate the convergence properties of the algorithm using cyclic block coordinate descent [32]. The detailed analysis is shown in Appendix B.

Theorem 1. *Let $\Theta^k = \{\mathcal{G}^k, \{\mathbf{U}_n^k\}\}$ be the sequence generated by Algorithm 1, then we assure that Θ^k converges to a critical point $\hat{\Theta} = \{\hat{\mathcal{G}}, \{\hat{\mathbf{U}}_n\}\}$.*

The proof of **Theorem 1** is based on the results given by [33]. For simplicity, we give a proof framework here and omit the details. Firstly, we establish a square summable result, i.e., $\sum_{k=1}^{\infty} \left\|\Theta^{k-1} - \Theta^k\right\|_F^2 < \infty$. Next, we can prove $\hat{\Theta}$ is a stationary point by verifying the first-order optimality conditions. Finally, the Kurdyka–Lojasiewicz (KL) inequality of $\mathbb{F}$ guarantees that Θ^k converges globally to a critical point.

C. Computational Complexity Analysis

Throughout this section, we denote the input tensor as $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and the core tensor as $\mathcal{G} \in \mathbb{R}^{I_1 \times \dots \times I_N}$. Considering the proposed Algorithm 1, gradient computing is the most time-consuming. Moreover, Lipschitz constants calculation is negligible since the components can be obtained during the gradients' computation. Assuming that the STRTD converges in the K iterations, we can roughly summarize the per-iteration complexity time complexity of the STRTD algorithm as

$$\mathcal{O}\left((N+1)\sum_{n=1}^{N}\left(\prod_{i=1}^{n} I_i\right)\left(\prod_{j=n}^{N} I_j\right)\right), \tag{20}$$

where the per-iteration cost is relevant to the tensor sizes $\prod_{i=1}^{n} I_i$, the proposed algorithm is theoretically efficient [33]. The detailed analysis is shown in Appendix C.

VI. EXPERIMENTS

In this section, we conduct experiments on two TDs to compare STRTD with baselines in different missing scenarios. All experiments are performed using MATLAB 2023a on a Windows 10 64-bit operating system on a workstation equipped with an Intel(R) Xeon(R) W-2123 CPU with 3.60 GHz and 64 GB RAM. Note that our Matlab codes are available on request.

A. Traffic Datasets

We use the following two TDs for our experiment and form them as 3rd-order tensors for traffic data imputation problems.

- (**G**): Guangzhou urban traffic speed dataset. The original data is of size 214 × 8784 in the form of a multivariate time series matrix. We select seven days for our model training and reshape it into 3-rd order tensor of size $214 \times 144 \times 7$, i.e., (sensors, time, day).
- (**A**): Internet traffic flow dataset in Abilene. Dataset **A** includes 11 OD pairs, recording traffic flow every 5 minutes from December 8, 2003, to December 14, 2003. We consider a 3rd-order tensor of size $121 \times 288 \times 7$, where the first dimension corresponds to 121 OD pairs, the second to the time interval, and the last to 7 days.

To analyze these datasets' spatiotemporal characteristics, we first calculate the spatial correlations [4] between various pairs of rows in traffic matrix $\mathbf{Y}$. Fig. 2 (a) depicts the cumulative distribution function (CDF) of the correlation coefficient. It indicates that over 50% of the sensors in two TDs exhibit strong correlations. This observation reveals that the sensor network in datasets **G** and **A** has strong spatial correlations. Fig. 2(b) shows the CDF of the traffic data with the increment rates (IRs) [12]. More than 50% of the data's IRs vary between 0.1 and 2, indicating temporal variations in the data. These results imply that the proposed spatiotemporal constraints are essential for our TDI problems.

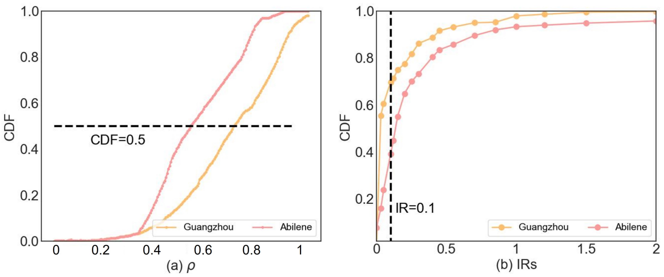

Fig. 2. Interpretation of the spatiotemporal characteristics in our TDs.

B. Experimental Settings

1) Missing scenario: For a thorough verification of the STRTD to TDI problem, we take into account three missing scenarios, i.e., random missing (RM), no-random missing (NM), and black-out missing (BM). Generally, RM means that missing data is uniformly distributed, and NM is conducted by randomly selecting sensors and discarding consecutive hours. At the same time, BM refers to all sensors not working for a certain period of time. According to the mechanisms, we mask the observed index set Ω and use the partial observations for the model training.

2) Baseline models: For comparison, we select six state-of-the-art spatiotemporal traffic data imputation methods: stTT [26], LATC [5], LR-SETD [21], BGCP [22], tSVD [35] and TAS-LR [12], to demonstrate the robustness and efficiency of our proposal. The baselines are shown in Tab. II, in which the TAS-LR is a matrix-based approach, LATC is the matricization method, and others are the tensor decomposition method.

TABLE II
COMPARISON OF BASELINE MODELS

Baselines	Spatiotemporal constraints			Structures
	Low rankness	Spatial	Temporal	
STRTD	✓	✓	✓	3rd tensor
stTT [26]	✓	✓	✓	3rd tensor
LATC [5]	✓		✓	3rd tensor
LR-SETD [21]	✓		✓	3rd tensor
BGCP [22]	✓		✓	3rd tensor
tSVD [35]	✓			3rd tensor
TAS-LR [12]	✓	✓	✓	Matrix

✓denotes the mentioned method has considered the constraint.

3) Model performance: To measure the imputation performance, we adopt two criteria, including mean absolute percentage error (MAPE) and normalized mean absolute error (NMAE):

$$\begin{aligned}\text{MAPE} &= \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right| \times 100,\\ \text{NMAE} &= \frac{\sum_{i=1}^{n}|y_i-\hat{y}_i|}{\sum_{i=1}^{n}|y_i|}\end{aligned} \tag{21}$$

where y_i and $\hat{y}_i$ are actual values and imputed values, respectively.

C. Implementation Details

Parameters setting: Two parameters α and β_n need to be tuned in our STRTD model. Hyperparameter α adjusts the strength of the sparsity term, i.e., the low-rank tensor approximation, and β_n characterizes spatiotemporal regularization. In all our experiments, we easily set the core tensor size to be the same as the traffic tensor and set $\alpha = 1$, which does not need to predefine the Tucker ranks. We calculate the maximum SVD value of spatiotemporal constraint matrices to deliver β_n, i.e., $\beta_n = \frac{1}{2*0.1*\sigma(\mathbf{L}\ or\ \mathbf{T}\mathbf{T}^{\top})}$. Additionally, we evaluate the performance of different strategies to varying sample ratios (SRs) under RM scenarios, with SRs ranging from 0.9 to 0.1, 0.07, and 0.05. Fig. 3 shows that setting the parameter γ to 0.2 reduces imputation error in high-level missing scenarios. In addition, the proposed initialization strategy reduces low-rank approximation errors. For better model comparison, the termination condition for all experiments is set to (19), where tol = 10^{-4} and the maximum number of iterations is 300. Furthermore, the parameters of baselines are optimally assigned or automatically chosen as described in the reference papers.

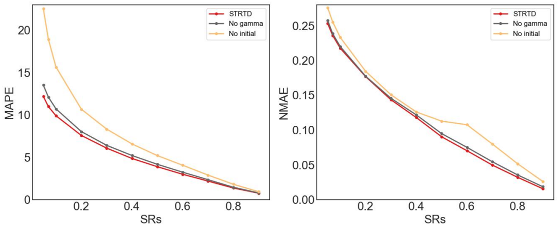

Fig. 3. Model performance over strategies for **G** (left) and **A** (right), respectively.

Ablation studies: To illustrate TDs' long and short-term patterns, we first discuss the tensor structure in our STRTD model under different RM ratios. In our analysis, we denote the mentioned 3rd-order traffic tensor as **M1**. Following the method proposed in [13], we reshape **G** into a $10 \times 205 \times 7 \times 1002$ tensor and **A** into a $11 \times 11 \times 288 \times 7$ tensor, represented by **M2**. Fig. 5 (a)-(b) shows the model performance; it can be seen that the 3rd-order tensor structure covers richer spatial and temporal information. To further verify the validity of the spatiotemporal regularizations, we discuss the effect of the spatial and temporal constraints of STRTD. Fig. 5(c)-(d) compares the influence of spatiotemporal constraints for datasets **G** and **A**, respectively. We can observe that spatial and temporal regularizations enhance the traffic data imputation performance. However, the temporal constraint plays a more critical role compared to the spatial constraint.

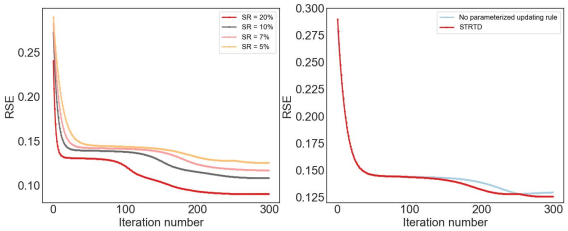

Fig. 4. The curves of the RSE values relative to the iterations under the RM scenarios on **G** dataset for different SRs.

Convergence behaviors: We have proven that Algorithm 1 sequences converge to a critical point theoretically in **Theorem 1**. Here, Fig. 4 (left) shows the curves of the relative square error (RSE) values versus the iteration number of the proposed STRTD on the **G** dataset to show the numerical convergence. Along with the iteration increases, the RSE decreases and stabilizes after approximately 200 iterations, indicating the numerical convergence of the algorithm. Furthermore, we test the RSE values of RM under SR = 0.05%, Fig. 4 (right) shows that the parameterized updating rule can speed up the convergence of Algorithm 1.

D. Results

This section will compare the STRTD method with other baselines mentioned in Tab. III.

1) Overall performance among baseline models: The ablation studies show that spatial and temporal constraints can enhance the model performance for the TDI problem. To show the superiority of the STRTD, Tab. III shows the overall performance of baseline models on the datasets **G** and **A** under various missing scenarios. The best error indicator values are bolded. From these quantitative comparisons, the low-rank tensor imputation methods outperform matrix-based ones. In addition, the STRTD can impute the TDs with fewer observed data more accurately. Specifically speaking, the proposed method achieves the lowest MAPE and NMAE values. Compared with other baselines, the STRTD model performs better for each RM case. Reconstructing the NM and BM scenarios is more challenging than the RM scenarios, but the proposed method consistently performs well. These results show that combining short-term traffic patterns with long-term trends benefits STRTD by utilizing low-rankness and spatiotemporal constraints.

We calculate the MAPE values for the **G** and NMAE values for the **A** under different RM scenarios (SR changes from 0.90 to 0.05) in Fig. 6. The results show that STRTD has the lowest value, even for the highly missing. Especially when the missing rate is 95%, imputing with MAPE and NMAE values results in improvements higher than 3%.

2) Imputation examples with different missing scenarios: Here, we show some STRTD imputation examples with different missing scenarios on the Guangzhou (**G**) dataset. For the RM scenario, Fig. 7 shows the same signal trends under

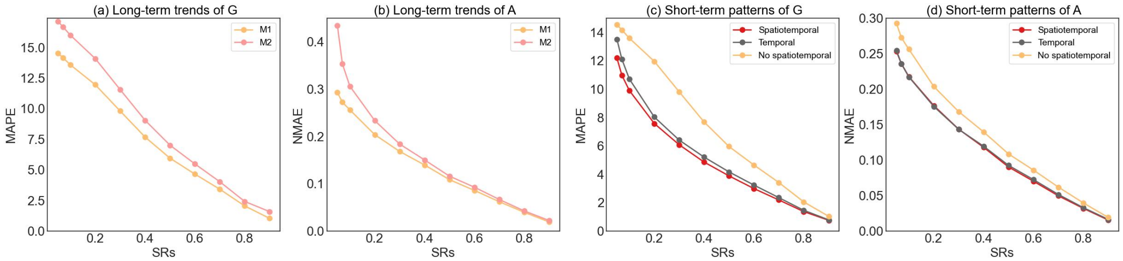

Fig. 5. Results of the ablation studies. (a)-(b) Interpretation of the multidimensionality of dataset **G** and **A**. (c)-(d) Illustration of the influence of spatiotemporal constraints for **G** and **A**, respectively.

TABLE III
Performance comparison of STRTD and other baselines for RM, NM, and BM scenarios

Data	Missing scenario	STRTD	tSVD	LATC	LR-SETD	BGCP	stTT	TAS-LR
G (MAPE)	RM-30%	**2.19**	2.23	5.94	5.32	6.91	10.96	10.12
	RM-70%	**6.08**	6.58	6.93	6.89	7.88	10.95	11.62
	RM-90%	**9.90**	11.17	10.05	10.44	10.35	11.28	15.11
	RM-95%	**12.19**	13.55	12.60	16.22	12.25	12.84	17.69
	NM-30%	**10.81**	12.93	74.98	13.48	15.62	11.88	12.31
	NM-70%	**12.48**	50.19	88.01	21.24	27.31	15.15	19.14
	NM-90%	**18.77**	85.51	87.66	57.33	32.73	21.97	52.01
	BM-30%	**13.56**	52.01	45.66	28.02	40.35	34.31	32.65
A (NMAE)	RM-30%	**0.0497**	0.0501	0.1229	0.1159	0.1196	0.2120	0.3092
	RM-70%	**0.1435**	0.1753	0.1488	0.1935	0.1527	0.2251	0.3178
	RM-90%	**0.2175**	0.2274	0.2328	0.2356	0.2361	0.3764	0.3407
	RM-95%	**0.2532**	0.2752	0.4809	0.2601	0.3636	0.5124	0.3576
	NM-30%	**0.2777**	-	-	0.6093	-	0.2869	0.3214
	NM-70%	0.4067	-	-	0.7401	-	**0.3725**	0.5791
	NM-90%	**0.7241**	-	-	0.8013	-	0.7303	0.8176
	BM-30%	0.4418	-	-	**0.2509**	-	0.3011	0.8417
Time (Seconds)		31	17	140	35	1922	**14**	75

The best results are highlighted in bold fonts, and - denotes that the algorithm is not applicable.

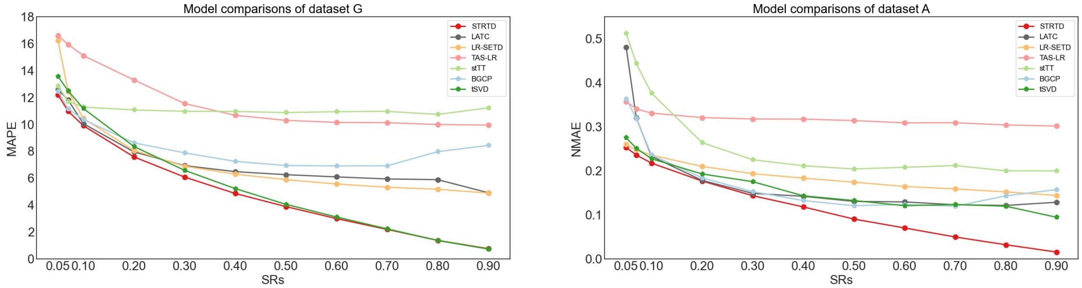

Fig. 6. MAPE and NMAE values for different sample ratios under RM scenarios for datasets **G** (left) and **A** (right), respectively.

different SRs (see the number of purple dots), indicating that the STRTD can accurately impute partial observations. Also, the residuals in Fig. 8 reveal that the STRTD can successfully reconstruct the TD precisely, even for the extreme case (i.e., 90% RM). To further validate the superiority of our STRTD, we plot the structural missing scenarios (NM and BM) result in Fig. 9 and Fig. 10. In this case, accurate imputation and traffic trend learning are achievable even in severe missing scenarios with STRTD.

VII. Conclusion

Traffic data imputation (TDI) is inevitable and challenging in data-driven intelligent transportation systems (ITS). This paper treats the TDI as a low-rank Tucker decomposition problem. The proposed STRTD exploits the long-term trends using a low-rank Tucker model and captures the short-term patterns with manifold regularization and temporal constraint. Through extensive experiments on two real-world TDs, our results show that the proposed STRTD beats other baselines for TDI with different RM scenarios and performs well on NM and BM missing scenarios (see Tab. III and Fig. 6).

There are three potential prospects for future work. First, our proposal ignores the exact rank and uses the sparse core tensor and non-negative factor matrix terms to promote low rankness. A potential approach is to use another low-

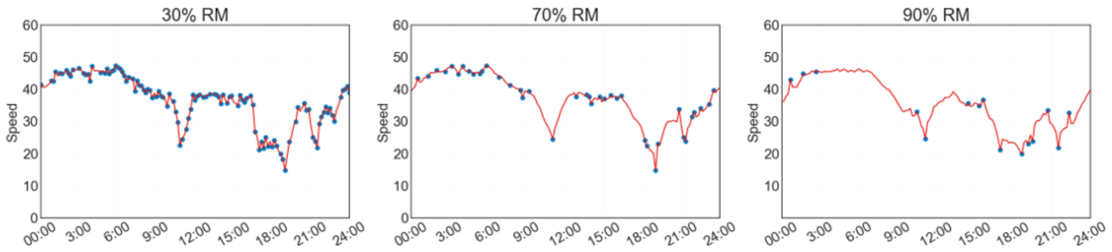

Fig. 7. Results of RM scenario on **G** dataset. This example corresponds to the 81st sensor and the 4th day of the dataset. Purple dots indicate the partially observed data, and red curves indicate the imputed values.

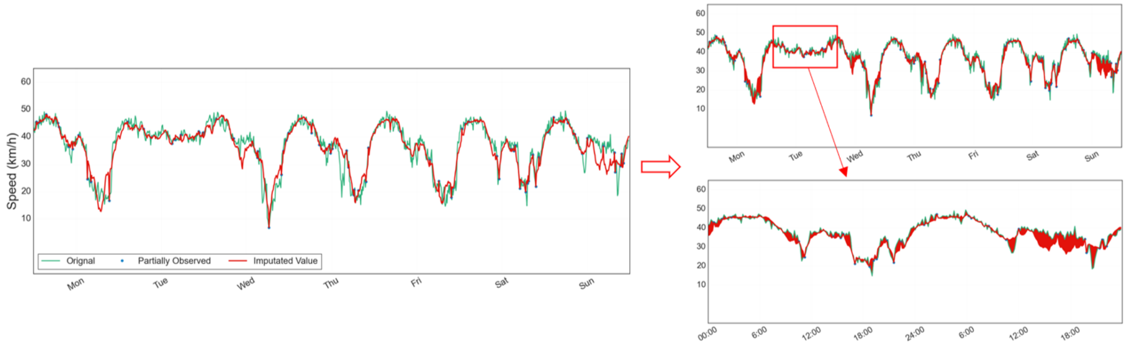

Fig. 8. Imputed values by STRTD on **G** dataset under RM scenario with 90% missing. Note that the red area (residual area) is only used to express the estimation performance, which does not represent the cumulative residual.

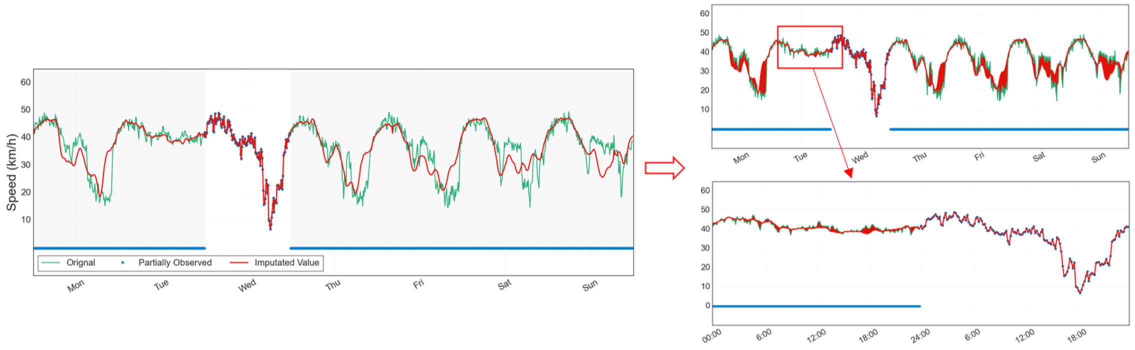

Fig. 9. Imputed values by STRTD on **G** dataset under NM scenario with 70% missing. The gray rectangles indicate the missing area.

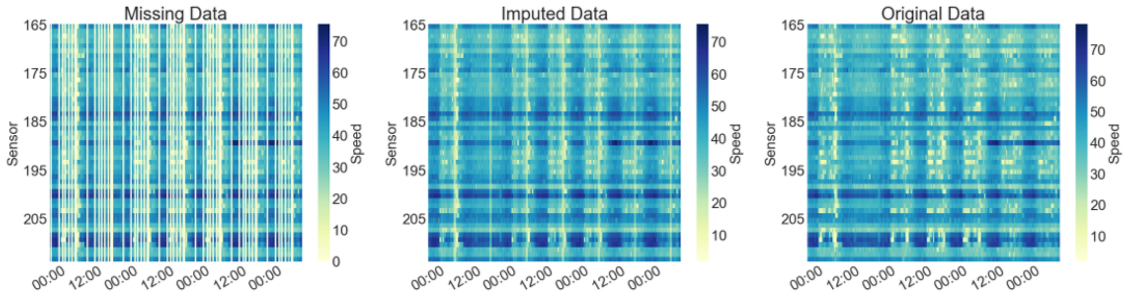

Fig. 10. The visualization of STRTD on **G** dataset under BM scenario with 30% missing. The middle heat map is our STRTD results.

rank tensor measure, such as multiplying the factor matrix rank to encode the Tucker rank [36]. Second, the current framework suffers a high computational cost for large-scale matrix multiplication calculations. One can consider the fast Fourier transform to address this issue [37]. Third, in addition to TDI, we can use the STRTD for spatiotemporal traffic data forecasting even with the missing observations [38]. Also, the proposed spatiotemporal traffic data modeling frameworks can be considered for urban traffic pattern discovery [27].

ACKNOWLEDGMENTS

The authors would like to thank the authors who shared their code and data on websites. This research is partly supported by the Shenzhen Science and Technology Program (Grant No. ZDSYS20210623092007023, JCYJ20200109141218676, K22627501) and Guangdong Province Universities and Colleges Key Areas of Special Projects (Grant No. 2021222012).

APPENDIX A
APG-BASED ALGORITHM FOR THE STRTD

We first denote the STRTD optimization problem as a class of regularized block multi-convex optimization problems:

$$\underset{\{\mathbf{x}_n\}}{\text{minimize}}\ \ell\left(\{\mathbf{x}_n\}\right) + \sum_{n=1}^{N} \beta_n f_n\left(\mathbf{x}_n\right),$$

where $f_n\left(\mathbf{x}_n\right)$ is the given constraint. We consider the APG-based prox-linear operator to update every $\mathbf{x}_n$ by solving a relaxed subproblem with a separable quadratic objective:

$$\begin{aligned}\mathbf{x}_n^k = \underset{\mathbf{x}_n}{\operatorname{argmin}}\{&\left\langle \tilde{\mathbf{g}}^k, \mathbf{x}_n - \tilde{\mathbf{x}}_n^{k-1}\right\rangle + \frac{L_{\mathbf{x}_n}^{k-1}}{2}\left\|\mathbf{x}_n - \tilde{\mathbf{x}}_n^{k-1}\right\|_F^2 \\ &+ \beta_n r_n\left(\mathbf{x}_n\right)\},\end{aligned} \tag{22}$$

where $\tilde{\mathbf{x}}_n^{k-1}$ denotes an extrapolated point and update through

$$\begin{aligned}&\tilde{\mathbf{x}}_n^{k-1} = \mathbf{x}_n^{k-1} + \omega_{k-1}\left(\mathbf{x}_n^{k-1} - \mathbf{x}_n^{k-2}\right),\ \text{for } k \geq 1 \\ &\omega_{k-1} = \frac{t^{k-2}-1}{t^{k-1}}, \quad t^{k-1} = \frac{p + \sqrt{r(t^{k-2})^2 + q}}{2},\end{aligned} \tag{23}$$

where $p, q \in [0,1]$, $r \in [0,4]$, and $\tilde{\mathbf{g}}^k = \nabla_{\mathbf{x}_n}\ell\left(\tilde{\mathbf{x}}_n^{k-1}\right)$ is the partial gradient of objective function ℓ. Guided by [34], we set $p = q = 0.8, r = 4$, and the updating rule (22) under these sequences has $\mathcal{O}\left(1/k^2\right)$ convergence rate.

Then, we provide detailed proof of **Proposition 1** and **Proposition 2**, followed by an explanation of the closed-form updating rule.

Proof of Proposition 1. Obviously, the Frobenius norm and matrix trace are differentiable functions. It remains to prove the convex property of ℓ and the Lipschitz continuous property of $\nabla_{\mathbf{U}_n}\ell$. Let $\Phi = \frac{1}{2}\left\|\mathbf{X}_{(n)} - \mathbf{U}_n\mathbf{G}_{(n)}\mathbf{V}_n^{\mathrm{T}}\right\|_{\mathrm{F}}^2$ and $g = \operatorname{tr}\left(\mathbf{U}_n^{\mathrm{T}}\mathbf{L}_n\mathbf{U}_n\right)$ or $\left\|\mathbf{T}_n\mathbf{U}_n\right\|_{\mathrm{F}}^2$, we have the gradient of $\ell(\mathbf{U}_n) = \Phi(\mathbf{U}_n) + \frac{\beta_n}{2}\Phi(\mathbf{U}_n)$

$$\nabla_{\mathbf{U}_n}\ell(\mathbf{U}_n) = \mathbf{U}_n\mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}} - \mathbf{X}_{(n)}{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}} + \nabla_{\mathbf{U}_n}g(\mathbf{U}_n). \tag{24}$$

where $\mathbf{G}_{\mathbf{V}}^n = \mathbf{G}_{(n)}\mathbf{V}_n^{\mathrm{T}}$ and $\mathbf{V}_n = \left(\otimes_{p\neq n}^1 \mathbf{U}_p\right)$.

On the one hand, the Hessian matrix of $\ell(\mathbf{U}_n)$ is given by

$$\nabla_{\mathbf{U}_n}^2\ell(\mathbf{U}_n) = \begin{cases} \mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}} + \beta_n\mathbf{L}_n, & \text{Manifold} \\ \mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}} + \beta_n\mathbf{T}_n^{\mathrm{T}}\mathbf{T}_n, & \text{Temporal} \\ \mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}}, & \text{otherwise} \end{cases} \tag{25}$$

As we know, the functions $\mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}}$, $\mathbf{L}_n$, and $\mathbf{T}_n\mathbf{T}_n^{\mathrm{T}}$ are both positive semi-definite, which shows that $\ell(\mathbf{U}_n)$ is convex.

On the other hand, we need the Lipschitz constant of $\nabla_{\mathbf{U}_n}\ell$. Since $\ell(\mathbf{U}_n)$ is a linear combination of $\Phi(\mathbf{U}_n)$ and $g(\mathbf{U}_n)$, the Lipschitz constant of $\nabla_{\mathbf{U}_n}\ell$ can be calculated as a linear combination of the Lipschitz constants of the $\nabla_{\mathbf{U}_n}\Phi$ and $\nabla_{\mathbf{U}_n}g$. Such as, taken $g(\mathbf{U}_n) = \frac{\beta_n}{2}\operatorname{tr}\left(\mathbf{U}_n^{\mathrm{T}}\mathbf{L}_n\mathbf{U}_n\right)$

$$\nabla_{\mathbf{U}_n}\ell(\mathbf{U}_n) = \mathbf{U}_n\mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}} - \mathbf{X}_{(n)}{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}} + \beta_n\mathbf{L}_n\mathbf{U}_n. \tag{26}$$

For any two matrices $\mathbf{U}_n^1, \mathbf{U}_n^2$, we have

$$\begin{aligned}&\left\|\nabla_{\mathbf{U}_n}\ell(\mathbf{U}_n^1) - \nabla_{\mathbf{U}_n}\ell(\mathbf{U}_n^2)\right\|_{\mathrm{F}}^2 \\ &= \left\|\left(\mathbf{U}_n^1 - \mathbf{U}_n^2\right)\mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}} - \beta_n\mathbf{L}_n\left(\mathbf{U}_n^1 - \mathbf{U}_n^2\right)\right\|_{\mathrm{F}}^2 \\ &\leq \left\|\left(\mathbf{U}_n^1 - \mathbf{U}_n^2\right)\mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}}\right\|_{\mathrm{F}}^2 + \left\|\beta_n\mathbf{L}_n\left(\mathbf{U}_n^1 - \mathbf{U}_n^2\right)\right\|_{\mathrm{F}}^2.\end{aligned} \tag{27}$$

So, we only need to calculate the Lipschitz constant of the composite gradient $\nabla_{\mathbf{U}_n}\ell$ separately. More specifically,

$$\begin{aligned}&\left\|\left(\mathbf{U}_n^1 - \mathbf{U}_n^2\right)\mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}}\right\|_{\mathrm{F}}^2 \\ &= \operatorname{tr}\left(\mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}}\left(\mathbf{U}_n^1 - \mathbf{U}_n^2\right)^{\mathrm{T}}\left(\mathbf{U}_n^1 - \mathbf{U}_n^2\right)\mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}}\right) \\ &\leq \left\|\mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}}\right\|_2^2\left\|\mathbf{U}_n^1 - \mathbf{U}_n^2\right\|_{\mathrm{F}}^2\end{aligned} \tag{28}$$

and

$$\begin{aligned}&\left\|\beta_n\mathbf{L}_n\left(\mathbf{U}_n^1 - \mathbf{U}_n^2\right)\right\|_{\mathrm{F}}^2 \\ &= \operatorname{tr}\left(\beta_n{\mathbf{L}_n}^{\mathrm{T}}\left(\mathbf{U}_n^1 - \mathbf{U}_n^2\right)^{\mathrm{T}}\left(\mathbf{U}_n^1 - \mathbf{U}_n^2\right)\beta_n\mathbf{L}_n\right) \\ &\leq \beta_n\left\|\mathbf{L}_n\right\|_2^2\left\|\mathbf{U}_n^1 - \mathbf{U}_n^2\right\|_{\mathrm{F}}^2\end{aligned} \tag{29}$$

where $\left\|\mathbf{G}_{\mathbf{V}}\right\|_2$ and $\left\|\mathbf{L}_n\right\|_2$ are the spectral norm with respect to $\mathbf{G}_{\mathbf{V}}$ and $\mathbf{L}_n$. Therefore, $\nabla_{\mathbf{U}_n}\ell(\mathbf{U}_n)$ is Lipschitz continuous and the Lipstchitz constant $L_{\mathbf{U}_n}$ is bounded. Furthermore, the gradient of temporal regularization satisfies

$$\left\|\beta_n\mathbf{T}_n^{\mathrm{T}}\mathbf{T}_n\left(\mathbf{U}_n^1 - \mathbf{U}_n^2\right)\right\|_{\mathrm{F}}^2 \leq \beta_n\left\|\mathbf{T}_n^{\mathrm{T}}\mathbf{T}_n\right\|_2^2\left\|\mathbf{U}_n^1 - \mathbf{U}_n^2\right\|_{\mathrm{F}}^2 \tag{30}$$

Combine with the above Equations, we define the Lipschitz constant $L_{\mathbf{U}_n}$ as

$$L_{\mathbf{U}_n} = \begin{cases} \left\|\mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}}\right\|_2 + \beta_n\left\|\mathbf{L}_n\right\|_2, & \text{Manifold} \\ \left\|\mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}}\right\|_2 + \beta_n\left\|\mathbf{T}_n^{\mathrm{T}}\mathbf{T}_n\right\|_2, & \text{Temporal} \\ \left\|\mathbf{G}_{\mathbf{V}}^n{\mathbf{G}_{\mathbf{V}}^n}^{\mathrm{T}}\right\|_2, & \text{otherwise} \end{cases} \tag{31}$$

This completes the proof. □

To solve (10), we take the derivative and set it to zeros, then we have

$$\mathbf{U}_n \longleftarrow \mathcal{P}_+\left(\tilde{\mathbf{U}}_n - \frac{1}{L_{\mathbf{U}_n}}\nabla_{\mathbf{U}_n}\ell\left(\tilde{\mathbf{U}}_n\right)\right), \tag{32}$$

where $\mathcal{P}_+(\mathbf{U})$ is the function that projects the negative entries of $\mathbf{U}$ into zeros and $\tilde{\mathbf{U}}_n$ is updated by

$$\tilde{\mathbf{U}}_n^k = \mathbf{U}_n^k + \omega_k\left(\mathbf{U}_n^k - \mathbf{U}_n^{k-1}\right),\ \text{for } k \geq 1.$$

with the updated step size ω_k using (17).

Proof of Proposition 2. As in Proposition 1, verifying the convex and Lipschitz continuous properties is straightforward. For the vectorization form, we have

$$\begin{aligned}\operatorname{vec}\left(\nabla_{\mathcal{G}}f(\mathcal{G})\right) &= \left(\otimes_{n=N}^1\mathbf{U}_n^{\mathrm{T}}\mathbf{U}_n\right)\operatorname{vec}(\mathcal{G}) \\ &- \left(\otimes_{n=N}^1\mathbf{U}_n^{\mathrm{T}}\right)\operatorname{vec}(\mathcal{X}),\end{aligned} \tag{33}$$

Then, the Hessian matrix $\mathrm{vec}\left(\nabla^2_{\mathcal{G}} f(\mathcal{G})\right) = \otimes_{n=N}^{1} \mathbf{U}_n^{\mathrm{T}}\mathbf{U}_n$, which is positive semi-definite and assures $f(\mathcal{G})$ is convex. Furthermore, we use the properties of Kronecker product to calculate $\nabla_{\mathcal{G}} f(\mathcal{G})$ as follows

$$\begin{aligned}\nabla_{\mathcal{G}} f(\mathcal{G}) = \mathcal{G} \times_1 \mathbf{U}_1^{\mathrm{T}}\mathbf{U}_1 \times \cdots \times_N \mathbf{U}_N^{\mathrm{T}}\mathbf{U}_N \\ - \mathcal{X} \times_1 \mathbf{U}_1^{\mathrm{T}} \times \cdots \times_N \mathbf{U}_N^{\mathrm{T}}.\end{aligned} \tag{34}$$

For any given $\mathcal{G}_1$ and $\mathcal{G}_2$, we have

$$\begin{aligned}&\left\|\mathrm{vec}\left(\nabla_{\mathcal{G}} f(\mathcal{G}_1)\right) - \mathrm{vec}\left(\nabla_{\mathcal{G}} f(\mathcal{G}_2)\right)\right\|_{\mathrm{F}} \\ &= \left\|\otimes_{n=N}^{1} \mathbf{U}_n^{\mathrm{T}}\mathbf{U}_n \left(\mathrm{vec}(\mathcal{G}_1) - \mathrm{vec}(\mathcal{G}_2)\right)\right\|_{\mathrm{F}} \\ &\leq \left\|\otimes_{n=N}^{1} \mathbf{U}_n^{\mathrm{T}}\mathbf{U}_n\right\|_2 \left\|\mathrm{vec}(\mathcal{G}_1) - \mathrm{vec}(\mathcal{G}_2)\right\|_{\mathrm{F}} \\ &= \textstyle\prod_{n=1}^{N} \left\|\mathbf{U}_n^{\mathrm{T}}\mathbf{U}_n\right\|_2 \left\|\mathrm{vec}(\mathcal{G}_1) - \mathrm{vec}(\mathcal{G}_2)\right\|_{\mathrm{F}}.\end{aligned} \tag{35}$$

So, the Lipschitz constant of $\nabla_{\mathcal{G}} f(\mathcal{G})$ is $L_{\mathcal{G}} = \prod_{n=1}^{N} \left\|\mathbf{U}_n^{\mathrm{T}}\mathbf{U}_n\right\|_2$. This completes the proof. □

Based on the results given by Proposition 2, we can use the soft thresholding operator [33] to solve the composite model (12), and the result is

$$\hat{\mathcal{G}} = T_{L_{\mathcal{G}}}^{f,g}(\mathcal{G}) = \mathcal{S}_{\frac{\alpha}{L_{\mathcal{G}}}}\left(\tilde{\mathcal{G}} - \frac{1}{L_{\mathcal{G}}}\nabla_{\mathcal{G}} f\left(\tilde{\mathcal{G}}\right)\right) \tag{36}$$

where $\mathcal{S}_{\zeta}(\cdot)$ is 'shrinkage' operator defining component-wisely as

$$\mathcal{S}_{\mu}(x) = \mathrm{sign}(x) \cdot \max(0, |x| - \mu).$$

and $\tilde{\mathcal{G}}$ is updated by

$$\tilde{\mathcal{G}}^k = \mathcal{G}^k + \omega_k\left(\mathcal{G}^k - \mathcal{G}^{k-1}\right), \text{ for } k \geq 1.$$

with the updated step size (17).

Appendix B
Convergence Analysis

We provide convergence proof for the proposed algorithm, which is given in the following three steps:

Square summable: We express (6) as $\mathbb{F}(\Theta) = \mathbb{F}_1(\Theta) + \mathbb{F}_2(\Theta)$, $\Theta = \{\{\mathbf{U}_n\}, \mathcal{G}\}$, where $\mathbb{F}_1$ is either function ℓ or f and $\mathbb{F}_2$ is either the l_1 norm or a non-negative projector. The prox-linear updating rule indicates

$$\hat{\Theta} = \underset{\Theta}{\mathrm{argmin}} \left\langle \nabla_{\Theta}\mathbb{F}_1(\tilde{\Theta}), \Theta - \tilde{\Theta}\right\rangle + \frac{L_{\Theta}}{2}\|\Theta - \tilde{\Theta}\|_F^2 + \mathbb{F}_2(\Theta), \tag{37}$$

where $\tilde{\Theta}$ is the extrapolation point. For any $\Theta^k = \{\{\mathbf{U}_n^k\}, \mathcal{G}^k\}$ generated by Algorithm 1, it is worth noting that Algorithm 1 takes L_{Θ}^{k-1} as the Lipschitz constant of $\nabla_{\Theta}\mathbb{F}_1(\Theta^k)$, the (38) is satisfied.

$$\begin{aligned}\mathbb{F}_1(\Theta^k) \leq \mathbb{F}_1(\Theta^{k-1}) + \left\langle \nabla_{\Theta}\mathbb{F}_1(\Theta^{k-1}), \Theta^k - \Theta^{k-1}\right\rangle \\ + \frac{L_{\Theta^{k-1}}}{2}\|\Theta^k - \Theta^{k-1}\|_F^2, \text{ for any} k = 1, \cdots, K.\end{aligned} \tag{38}$$

Considering the convexity of $\mathbb{F}_1, \mathbb{F}_2$, then the proximal gradient inequality assures that

$$\mathbb{F}(\Theta) - \mathbb{F}(\hat{\Theta}) \geq \frac{L_{\Theta}}{2}\|\hat{\Theta} - \tilde{\Theta}\|_F^2 + L_{\Theta}\left\langle \tilde{\Theta} - \Theta, \hat{\Theta} - \tilde{\Theta}\right\rangle. \tag{39}$$

Based on the results given by ***Proposition 1*** and ***Proposition 2***, we have $\nabla_{\Theta}\mathbb{F}_1(\Theta)$ is Lipschitz continuous, which has bounded Lipschitz constant. Then for three successive $\Theta^{k-2}, \Theta^{k-1}, \Theta^k$ given by the updated step (17), we have

$$\begin{aligned}&\mathbb{F}(\Theta^{k-1}) - \mathbb{F}(\Theta^k) \\ &\geq \frac{L_{\Theta}^{k-1}}{2}\|\Theta^k - \tilde{\Theta}^{k-1}\|_F^2 + L_{\Theta}^{k-1}\left\langle \tilde{\Theta}^{k-1} - \Theta^{k-1}, \Theta^k - \tilde{\Theta}^{k-1}\right\rangle \\ &\geq \frac{L_{\Theta}^{k-1}}{2}\|\Theta^{k-1} - \Theta^k\|_F^2 - \frac{L_{\Theta}^{k-2}\delta_{\omega}}{2}\|\Theta^{k-2} - \Theta^{k-1}\|_F^2,\ \delta_{\omega}<1.\end{aligned} \tag{40}$$

Summing the above inequality over k from 1 to K, we have

$$\mathbb{F}(\Theta^0) - \mathbb{F}(\Theta^K) \geq \sum_{k=1}^{K} \text{const.}\ \|\Theta^{k-1} - \Theta^k\|_F^2. \tag{41}$$

Letting $K \to \infty$ and observing $\mathbb{F}$ is lower bounded, we have

$$\sum_{k=1}^{\infty} \left\|\Theta^{k-1} - \Theta^k\right\|_F^2 < \infty. \tag{42}$$

Subsequence convergence: Recall the prox-linear operator mentioned in (37), which is a block multi-convex minimization problem. Depending on the square summable property, we set $\hat{\Theta}$ as a limit point of Θ. For the given $\mathcal{X}$, we have

$$\begin{aligned}\hat{\mathcal{G}} = \underset{\mathcal{G}}{\mathrm{argmin}} \left\langle \nabla_{\mathcal{G}} f(\tilde{\mathcal{G}}, \{\hat{\mathbf{U}}_n\}), \mathcal{G} - \tilde{\mathcal{G}}\right\rangle \\ + \frac{\tilde{L}_{\mathcal{G}}}{2}\|\mathcal{G} - \tilde{\mathcal{G}}\|_F^2 + \alpha\|\mathrm{vec}(\mathcal{G})\|_1.\end{aligned} \tag{43}$$

Letting $\tilde{\mathcal{G}} \to \hat{\mathcal{G}}$, we have $\tilde{L}_{\mathcal{G}} \to \hat{L}_{\mathcal{G}}$ and get

$$\left\langle \nabla_{\mathcal{G}} f(\mathcal{G}) + \alpha\mathbb{P}_{\mathcal{G}}, \mathcal{G} - \hat{\mathcal{G}}\right\rangle \geq \mathbf{0}, \text{ for some } \mathbb{P}_{\mathcal{G}} \in \partial\|\mathrm{vec}(\mathcal{G})\|_1. \tag{44}$$

Hence, $\hat{\mathcal{G}}$ satisfies the first-order optimality condition of (12). Similarly, we have for all $\mathbf{U}_n$ that

$$\left\langle \nabla_{\mathbf{U}_n}\ell(\mathbf{U}_n) + \mathbb{P}_{\mathbf{U}_n}, \mathbf{U}_n - \hat{\mathbf{U}}_n\right\rangle \geq \mathbf{0}, \text{ for all } \mathbf{U}_n \geq 0. \tag{45}$$

The above equations give the first-order optimality conditions of (6), and then subsequence $\Theta^k = \{\{\mathbf{U}_n^k\}, \mathcal{G}^k\}$ converges to critical point $\hat{\Theta}$. Furthermore, at the kth iteration of Algorithm 1, we performs re-update when $\mathbb{F}\left(\Theta_k\right) < \mathbb{F}\left(\Theta_{k-1}\right)$, which assures the objective $\mathbb{F}$ nonincreasing. Hence, the convergence result still holds with an extra updated (18).

Global convergence: It is straightforward to verify $\ell(\cdot), \|\cdot\|, \mathcal{P}_+(\cdot)$ are semi-algebraic functions and then demonstrate that $\mathbb{F}$ satisfies the Kurdyka–Lojasiewicz (KL) property [39] at $\hat{\Theta}$, namely, there exists $\mu, \rho > 0, \eta \in [0, 1]$, and a neighborhood $\mathcal{B}(\hat{\Theta}, \rho) = \left\{\Theta : \|\Theta - \hat{\Theta}\|_F^2 \leq \rho\right\}$ such that

$$\|\mathbb{F}(\Theta) - \mathbb{F}(\hat{\Theta})\|^{\eta} \leq \mu \cdot \mathrm{dist}(\mathbf{0}, \partial\mathbb{F}(\Theta)), \text{ for all } \Theta \in \mathcal{B}(\hat{\Theta}, \rho). \tag{46}$$

Combining the subsequence convergence and KL property, the sequence Θ^k converges globally to a critical point $\hat{\Theta}$ of equation (6).

Appendix C
Computational Complexity Analysis

The Tucker decomposition algorithms compute the huge matrix multiplication and suffer from very high computational complexity; we combine the low-rank approximation with population Tucker decomposition strategies to reduce the computational complexity [40]. Here, we analyze the computational complexity of the proposed STRTD. Suppose that $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and the core tensor $\mathcal{G} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, we have the basic computational complexity: the computational cost of $\mathbf{U}_n^{\mathrm{T}} \mathbf{U}_n$ is $\mathcal{O}(I_n^3)$ and the mode-n product with the matrix $\mathbf{U}_n$ of tensor $\mathcal{G}$ is $\mathcal{O}(\sum_{n=1}^{N} \prod_{i=1}^{n} I_i \prod_{j=1}^{N} I_j)$. Furthermore, we reformulate the Kronecker product in $\mathbf{G}_{\mathbf{V}}^n = \mathbf{G}_{(n)} \mathbf{V}_n^{\mathrm{T}}$ but let

$$\mathcal{Y} = \mathcal{G} \times_1 \mathbf{U}_1 \cdots \times_{n-1} \mathbf{U}_{n-1} \times_{n+1} \mathbf{U}_{n+1} \cdots \times_N \mathbf{U}_N,$$

such that we have $\mathbf{G}_{\mathbf{V}}^n = \mathcal{Y}_{(n)}$ and its computational cost is

$$\begin{aligned} \mathcal{O}\left(\mathbf{G}_{\mathbf{V}}^n\right) \; = \; & \mathcal{O}\left(\sum_{j=1}^{n-1} \left(\prod_{i=1}^{j} I_i \right) \left(\prod_{i=j}^{N} I_i \right) \right) + \\ & \mathcal{O}\left(\left(\prod_{i=1}^{n} I_i \right) \sum_{j=n+1}^{N} \left(\prod_{i=n+1}^{j} I_i \right) \left(\prod_{i=j}^{N} I_i \right) \right) \\ & \leq \mathcal{O}\left(\sum_{n=1}^{N} \left(\prod_{i=1}^{n} I_i \right) \left(\prod_{j=n}^{N} I_j \right) \right) \end{aligned} \tag{47}$$

Also, we conclude that the computational cost of tensor unfolding, soft-thresholding operator, and projection to nonnegative is negligible compared to gradient computing.

Considering the proposed APG-based optimization for core tensor "shrinkage", the computation of $\nabla_{\mathcal{G}} f\left(\mathcal{G}\right)$ requires

$$\mathcal{O}\left(\sum_{n=1}^{N} I_n^3 + \sum_{n=1}^{N} I_n \prod_{i=1}^{N} I_i + \sum_{n=1}^{N} \left(\prod_{i=1}^{n} I_i \right) \left(\prod_{j=n}^{N} I_j \right) \right). \tag{48}$$

where the first part comes from the computation of all $\mathbf{U}_n^{\mathrm{T}} \mathbf{U}_n$, and the second and third parts are respectively, from the computations of the first and second terms in (34).

Similarly, we use (47) to calculate the computational complexity of $\nabla_{\mathbf{U}_n} \ell(\mathbf{U}_n)$ and requires

$$\mathcal{O}\left(I_n \left(\prod_{i=1}^{n} I_i \right) + I_n^3 \right) + \mathcal{O}\left(\prod_{i=1}^{n} I_i \right) + \mathcal{O}\left(I_n^3\right) + \mathcal{O}\left(\mathbf{G}_{\mathbf{V}}^n\right). \tag{49}$$

The first three parts are from the computations of the three terms in (24), and (49) is dominated by the last part. So, the computational cost of $\nabla_{\mathcal{G}} f\left(\mathcal{G}\right)$ and $\nabla_{\mathbf{U}_n} \ell(\mathbf{U}_n)$ are

$$\mathcal{O}\left(\sum_{n=1}^{N} \left(\prod_{i=1}^{n} I_i \right) \left(\prod_{j=n}^{N} I_j \right) \right). \tag{50}$$

References

[1] X. Chen, Z. He, and J. Wang, "Spatial-temporal traffic speed patterns discovery and incomplete data recovery via SVD-combined tensor decomposition," *Transportation Research Part C: Emerging Technologies*, vol. 86, pp. 59–77, 2018.

[2] S. Moritz and T. Bartz-Beielstein, "imputeTS: Time Series Missing Value Imputation in R," *The R Journal*, vol. 9, no. 1, pp. 207–218, 2017.

[3] T. Thomas and E. Rajabi, "A systematic review of machine learning-based missing value imputation technique," *Data Technologies and Applications*, 2021.

[4] 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.

[5] X. Chen, M. Lei, N. Saunier, and L. Sun, "Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2021.

[6] T. G. Kolda and B. W. Bader, "Tensor decompositions and application," *SIAM Review*, vol. 5, no. 3, p. 455-500, 2009.

[7] Q. Song, H. Ge, J. Caverlee, and X. Hu, "Tensor completion algorithms in big data analytics," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 1, p. 48, 2019.

[8] 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," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 66–77, 2019.

[9] M. T. Bahadori, Q. R. Yu, and Y. Liu, "Fast multivariate spatio-temporal analysis via low rank tensor learning," in *Neural Information Processing Systems (NIPS)*, 2014, p. 3491–3499.

[10] A. B. Said and A. Erradi, "Spatiotemporal tensor completion for improved urban traffic imputation," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2021.

[11] 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, 2012.

[12] Y. Wang, Y. Zhang, X. Piao, H. Liu, and K. Zhang, "Traffic data reconstruction via adaptive spatial-temporal correlations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1531–1543, 2019.

[13] X. Wang, Y. Wu, D. Zhuang, and L. Sun, "Low-rank Hankel tensor completion for traffic speed estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 5, pp. 4862–4871, 2023.

[14] X. Li, M. K. Ng, G. Cong, Y. Ye, and Q. Wu, "MR-NTD: Manifold regularization nonnegative Tucker decomposition for tensor data dimension reduction and representation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 8, pp. 1787–1800, 2017.

[15] P. Wu, L. Xu, and Z. Huang, "Imputation methods used in missing traffic data: A literature review," in *Artificial Intelligence Algorithms and Applications*, 2020, pp. 662–677.

[16] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, p. 717–772, 2009.

[17] 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, 2013.

[18] B. Ran, H. Tan, Y. Wu, and P. J. Jin, "Tensor based missing traffic data completion with spatial–temporal correlation," *Physica A : Statistical Mechanics and its Applications*, vol. 446, pp. 54–63, 2016.

[19] H. Tan, J. Feng, Z. Chen, F. Yang, and W. Wang, "Low multilinear rank approximation of tensors and application in missing traffic data," *Advances in Mechanical Engineering*, vol. 6, pp. 1575–1597, 2014.

[20] T. Yokota, B. Erem, S. Guler, S. K. Warfield, and H. Hontani, "Missing slice recovery for tensors using a low-rank model in embedded space," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8251–8259.

[21] C. Pan, C. Ling, H. He, L. Qi, and Y. Xu, "A low-rank and sparse enhanced Tucker decomposition approach for tensor completion," *Applied Mathematics and Computation*, vol. 465, p. 128432, 2024.

[22] X. Chen, Z. He, and L. Sun, "A bayesian tensor decomposition approach for spatiotemporal traffic data imputation," *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 73–84, 2019.

[23] X. Chen, Y. Chen, N. Saunier, and L. Sun, "Scalable low-rank tensor learning for spatiotemporal traffic data imputation," *Transportation Research Part C: Emerging Technologies*, vol. 129, p. 103226, 2021.

[24] Q. Shi, J. Yin, J. Cai, A. Cichocki, T. Yokota, L. Chen, M. Yuan, and J. Zeng, "Block Hankel tensor ARIMA for multiple short time series forecasting," in *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 30, no. 04, 2020, pp. 5758–5766.

[25] Y. Wu, H. Tan, Y. Li, J. Zhang, and X. Chen, "A fused CP factorization method for incomplete tensors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 751–764, 2019.

[26] Z. Zhang, Y. Chen, H. He, and L. Qi, "A tensor train approach for internet traffic data completion," *Annals of Operations Research*, vol. 06, pp. 12–19, 2021.

[27] X. Chen, C. Zhang, X. Chen, N. Saunier, and L. Sun, "Discovering dynamic patterns from spatiotemporal data with time-varying low-rank autoregression," *arXiv*, vol. abs/2211.15482, 2022.

[28] Y.-L. Chen, C.-T. Hsu, and H.-Y. M. Liao, “Simultaneous tensor decomposition and completion using factor priors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 577–591, 2014.

[29] Q. Yu, X. Zhang, Y. Chen, and L. Qi, “Low tucker rank tensor completion using a symmetric block coordinate descent method,” *Numerical Linear Algebra with Applications*, vol. 30, no. 3, p. e2464, 2023.

[30] W. Gong, Z. Huang, and L. Yang, “Accurate regularized tucker decomposition for image restoration,” *Applied Mathematical Modeling*, vol. 123, no. 11, pp. 75–86, 2023.

[31] T. K. Sinha, J. Naram, and P. Kumar, “Nonnegative low-rank tensor completion via dual formulation with applications to image and video completion,” in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2022, pp. 3732–3740.

[32] Y. Xu and W. Yin, “A Block Coordinate Descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[33] Y. Xu, “Alternating proximal gradient method for sparse nonnegative Tucker decomposition,” *Mathematical Programming Computation*, vol. 5, no. 3, p. 455–500, 2015.

[34] J. Liang, T. Luo, and C.-B. Schönlieb, “Improving “fast iterative shrinkage-thresholding algorithm”: Faster, smarter, and greedier,” *SIAM Journal on Scientific Computing*, vol. 44, no. 3, pp. A1069–A1091, 2022.

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

[36] Q. Xie, Q. Zhao, D. Meng, and Z. Xu, “Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 8, pp. 1888–1902, 2018.

[37] R. Yamamoto, H. Hontani, A. Imakura, and T. Yokota, “Fast algorithm for low-rank tensor completion in delay-embedded space,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 2048–2056.

[38] X. Chen and L. Sun, “Bayesian temporal factorization for multidimensional time series prediction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4659–4673, 2022.

[39] J. Bolte, A. Daniilidis, and A. Lewis, “The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems,” *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 1205–1223, 2007.

[40] G. Zhou, A. Cichocki, Q. Zhao, and S. Xie, “Efficient nonnegative Tucker decompositions: Algorithms and uniqueness,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4990–5003, 2015.