Tucker Factorization-based Tensor Completion for Robust Transport Data Imputation

Cheng Lyu[a], Qing-Long Lu[a], Xinhua Wu[b] and Constantinos Antoniou[a,*]

[a]Chair of Transportation Systems Engineering, Technical University of Munich, Munich, Germany
[b]Department of Civil and Environmental Engineering, Northeastern University, Boston, MA, USA

ARTICLE INFO

Keywords:
Tensor completion
Tucker factorization
Data imputation
Changepoint detection
Bregman ADMM

ABSTRACT

Missing values are prevalent in spatio-temporal transport data, undermining the quality of data-driven analysis. While prior works have demonstrated the promise of tensor completion methods for imputation, their performance remains limited for complicated composite missing patterns. This paper proposes a novel imputation framework combining tensor factorization and rank minimization, which is effective in capturing key traffic dynamics and eliminates the need for exhaustive rank tuning. The framework is further supplemented with time series decomposition to account for trends, spatio-temporal correlations, and outliers, with the intention of improving the robustness of imputation results. A Bregman ADMM algorithm is designed to solve the resulting multi-block nonconvex optimization efficiently. Experiments on four real-world transport datasets suggest that the proposed framework outperforms state-of-the-art imputation methods, including the context of complex missing patterns with high missing rates, while maintaining reasonable computation efficiency. Furthermore, the robustness of our model in extreme missing data scenarios, as well as under perturbation in hyperparameters, has been validated. These results also underscore the potential benefits of incorporating temporal modeling for more reliable imputation.

1. Introduction

1.1. Background

Transport data, enriched by contemporary data collection techniques and open data initiatives, propels a multitude of applications, including traffic forecasting, traffic state estimation, and traffic flow analysis (Mahajan et al., 2022; Liu et al., 2021). These data-driven studies have enabled improved decision-making for the efficiency, safety, and sustainability of urban transport systems. However, the integrity of transport data is often undermined by missing values, particularly in time series data like traffic state, public transport ridership, and shared mobility usage. Even with the progression of transportation management systems, the issue of missing data is still not rare. For instance, a considerable number of traffic sensors with over half of their readings missing were reported by Laña et al. (2018), where only 11% of sensors in the study area demonstrated more than 98% data completeness. Another piece of evidence from bicycle volume data indicated that only 54% of the data is viable for use (El Esawey et al., 2015). These issues pose significant challenges for time series modeling, leading to a compromise in terms of analysis and prediction accuracy (Fang et al., 2023). Hence, devising accurate data imputation methods becomes essential to address the challenges of missing values in transport data.

Transport data can be corrupted for various reasons, including communication failure, sensor failure, infrastructure upgrading, and database failure, resulting in different missing patterns. Notably, missing data does not necessarily indicate the absence of entries. As summarized in *AASHTO Guidelines for Traffic Data Programs*, erroneous records, such as repeated, extreme, or error-coded values, also represent missing data as they are unusable for further analysis (AASHTO, 2009). One common missing data scenario in transport pertains to occasional random missing entries, which can generally be addressed by simple methods like temporal interpolation and spatial weighting. However, structured missing entries, characterized by missing data in the form of spatial and temporal clusters, pose more significant challenges for imputation methods. It was early emphasized by Smith et al. (2003) that missing entries in transport data can happen in both temporal and spatial domains, with varying time spans of missing values.

*Corresponding author
cheng.lyu@tum.de (C. Lyu); qinglong.lu@tum.de (Q. Lu); wu.xinh@northeastern.edu (X. Wu); c.antoniou@tum.de (C. Antoniou)
ORCID(s): 0000-0002-6356-6947 (C. Lyu); 0000-0002-6087-8670 (Q. Lu); 0000-0003-0203-9542 (C. Antoniou)

A vast collection of transport data are spatio-temporal data, which can be naturally organized as matrices spanning space and time. The periodicity of transport time series further allows wrapping the temporal dimension into a higher number of dimensions like time-of-the-day and day-of-the-week. Given such inherent tensorial structure, tensor/matrix factorization and completion have emerged as one of the most promising solutions for transport data imputation, offering better imputation accuracy and higher computational efficiency over other methods like simple interpolation and statistical learning. Initially developed within the domain of computer vision and recommendation systems (Liu et al., 2013; Salakhutdinov and Mnih, 2008), these techniques are capable of modeling intricate multilinear relationship and filling in missing data by exploiting the low-rank property of transport data. They have proven effective for understanding urban mobility dynamics (Sun and Axhausen, 2016) and improving imputation accuracy across various missing patterns (Tan et al., 2013; Li et al., 2013; Chen et al., 2019).

Nevertheless, the performance of many existing tensor-based imputation methods are still limited, when applied on spatio-temporal transport data with complex missing patterns. Similar to the cases in image recovery (Yamamoto et al., 2022), some imputation methods have been found to falter faced with composite missing patterns, especially when spatial and temporal missing happens simultaneously. Additionally, data outliers—especially extreme values—can incur instability in imputation results, while variations in traffic dynamics due to supply-side changes, such as fleet expansion or infrastructure damage, may also adversely affect imputation accuracy. To develop an effective imputation method for practical transport data, a significant challenge lies in enhancing the robustness to handle various missing patterns, outliers, and temporal pattern shifts. As such, bridging the gap between existing tensor methods and the multifaceted real-world complexity remains an open research problem.

1.2. Literature Review

The effective imputation of missing values in transport data primarily revolves around exploiting temporal patterns and spatial similarities. These ideas are often realized through interpolation of observed values, fitting a statistical learning model, and optimizing a priori objective. A broad range of methodologies adhering to these ideas have been explored over the years, falling into three general paradigms: simple interpolation, machine learning, and tensor-based methods.

Typical simple interpolation methods include historical averaging, and weighted averaging across neighboring time periods and sensors (Smith et al., 2003). While straightforward to implement, these methods are only effective for isolated missing entries due to their strong dependency on historical or surrounding data. The imputation quality can be improved via parametric models like autoregressive models (Tight et al., 1993), which can relieve the difficulty in determining weights for averaging and reduce the impacts of outliers.

To address complicated missing patterns, recent researches resort to machine learning models, because of their strong predictive power. One line of research leverages clustering to extract temporal patterns, thereby facilitating missing entry imputation (Tang et al., 2015). Clustering can also be integrated with classification models to counteract its limitation with whole-day missing scenarios (Laña et al., 2018). The importance of explicit temporal modeling, e.g., the Prophet model (Li et al., 2020) and Gaussian process with a periodical kernel (Jiang et al., 2022), is later acknowledged for handling complex missing scenarios. However, these methods can become limited for either complicated workflows or high computational complexity. Hence, deep learning, which has seen significant success in traffic forecasting, has been recently introduced for imputation. Noteworthy methods include denoising stacked autoencoder (Duan et al., 2016), graph convolutional neural network (Chen and Chen, 2022), and attention mechanism (Liang et al., 2022). The common idea behind them is to encode the spatio-temporal patterns in latent spaces or a memory module. However, generating appropriate training samples is non-trivial. Most deep learning-based models use a sliding window strategy to prepare samples, which are short time series segments ranging from 15 minutes to one day. Such small window sizes restrict the contextual information input to the model, limiting their applicability in scenarios with longer missing periods.

Tensor-based methods provide a streamlined imputation formulation by exploiting the inherent low-rank property and multi-dimensional relationships present in transport data. One approach is tensor factorization, which breaks transport data into several latent factors; it degenerates into matrix factorization when the transport data is two-dimensional (Li et al., 2013). Each decomposed factor encodes the information of a specific dimension, indicating key time series patterns and correlations among sensors. Missing value imputation can be realized by minimizing the reconstruction error from factors (Tan et al., 2013) or maximizing the posterior likelihood in the Bayesian context (Chen et al., 2019). Regularization can also be applied to enforce spatial and temporal smoothness (Chen et al., 2018). However, a significant limitation of tensor factorization is the need for predefined ranks as hyperparameters, which are hard to determine in practice. Tensor completion, on the other hand, seeks a low-rank approximation of transport data by directly minimizing the rank of the reconstructed tensor. Given that tensor rank is non-differentiable, many studies focused on finding a viable rank approximation, e.g., truncated nuclear norm (Chen et al., 2020) and Schatten-p norm (Nie et al., 2022). However, tensor rank and its approximations are permutation and scale invariant, i.e., permuting or scaling the rows or columns of a matrix does not change its rank and singular values. Therefore, it may result in suboptimal imputation results that deviate from observed temporal patterns when temporal dependencies within the data are not explicitly accounted for via time series modeling. Integrating tensor completion with autoregressive models has proven to be an effective solution to this problem (Chen et al., 2022).

1.3. Objectives and Contribution

This study, following the paradigm of tensor-based method, aims to design a new transport data imputation framework, namely robust Tucker factorization-based tensor completion (RTTC), which is effective for various composite missing patterns. Our contribution can be summarized into the following points:

- The framework leverages tensor factorization techniques combined with rank minimization, effectively eliminating the need for predefined ranks and enhancing the accuracy of imputation.
- To account for outliers and temporal pattern shifts, the framework is embedded with a time series decomposition model, enabling effective modeling of temporal evolution in transport data.
- An improved alternating direction method of multiplier (ADMM) algorithm is designed to solve the multi-block nonconvex separable problem in the proposed framework.

This paper is structured as follows. In Section 2, we present the fundamentals of tensors and establish the problem definition. Section 3 details our novel transport data imputation framework. In Section 4, we present the experimental results, demonstrating the effectiveness of our approach. Finally, in Section 5, we provide a conclusion and outline potential future research directions.

2. Preliminaries

2.1. Notations

In this paper, scalars are represented by italic letters, vectors by boldface lowercase letters, matrices by boldface uppercase letters, and tensors by calligraphic letters, for example, x, $\boldsymbol{x}$, $\boldsymbol{X}$, and $\mathcal{X}$, respectively.

A tensor with k modes, also referred to as a k-way tensor or a k-dimensional tensor, is represented as $\mathcal{X} \in \mathbb{R}^{N_1,N_2,\ldots,N_k}$, where N_n indicates the size of the n-th mode. Subscripts added to a tensor, such as $x_{i_1 i_2 \ldots i_k}$, indicate the $(i_1, i_2, \ldots, i_k)$-th entry of the tensor.

In reference to the transport data discussed in this paper, we begin with a generic matrix representation, $\boldsymbol{X} \in \mathbb{R}^{N_S \times N_T}$, that includes a time series of measurements over a specific period gathered from a range of sensors. Here, N_S represents the number of sensors and N_T refers to the number of time slots. This matrix structure can be reorganized into a tensor structure by reshaping along the temporal mode. For example, a four-way tensor, $\mathcal{X} \in \mathbb{R}^{N_s \times N_w \times N_d \times N_t}$, can be obtained by reshaping the temporal mode into weeks and days-of-the-week, where N_w, N_d, and N_t indicate the number of weeks, days in a week, and time slots in a day, respectively. It is worth noting that, when N_T is not divisible by the product of N_w, N_d, and N_t, the matrix can be padded with null values in the temporal mode to allow proper reshaping. Also, more temporal modes, such as month-of-the-year, can be incorporated to transform the matrix into a higher-order tensor when dealing with data spanning a long time period.

2.2. Tensor Basics

A tensor can be converted from and to a matrix through the folding and unfolding operations along its n-th mode,

$$\mathcal{X} = \mathrm{fold}_n(\boldsymbol{X}_{(n)}; N_1, \ldots, N_k) \in \mathbb{R}^{N_1,N_2,\ldots,N_k}, \tag{1}$$

$$\boldsymbol{X}_{(n)} = \mathrm{unfold}_n(\mathcal{X}) \in \mathbb{R}^{N_n \times (N_1 \times \cdots \times N_{n-1} \times N_{n+1} \times \cdots \times N_k)}. \tag{2}$$

Similar to a matrix, a tensor can be characterized by a range of different norms. The ℓ_1 norm of a tensor $\|\cdot\|_1$ can be defined as the absolute sum of all its entries, and the Frobenius norm of a tensor $\|\cdot\|_F$ can be defined as the square root of the squared sum of all its entries,

$$\|\mathcal{X}\|_1 = \sum_{i_1,i_2,\ldots,i_k} |x_{i_1,i_2,\ldots,i_k}|, \quad \|\mathcal{X}\|_F = \sqrt{\sum_{i_1,i_2,\ldots,i_k} x^2_{i_1,i_2,\ldots,i_k}}. \tag{3}$$

Tensor factorization refers to decomposing a tensor into several factors. In this paper, we focus on Tucker factorization, which can factorize a tensor into the $k+1$ components, including a core tensor and k factor matrices (Kolda and Bader, 2009),

$$\mathcal{X} = \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \cdots \times_k \boldsymbol{U}_k. \tag{4}$$

where $\mathcal{G} \in \mathbb{R}^{r_1 \times \cdots \times r_k}$ is the core tensor and $\boldsymbol{U}_n \in \mathbb{R}^{N_n \times r_n} (n \in [k])$ are the factor matrices, respectively. $r_1, \ldots, r_k$ are the rank of factor matrices. The operator $\times_n$ is the n-mode product.

Given an arbitrary tensor $\mathcal{A} \in \mathbb{R}^{r_1 \times \cdots \times r_k}$ and a matrix $\boldsymbol{U} \in \mathbb{R}^{N_n \times r_n}$, their n-mode product, denoted by $\mathcal{B} \in \mathbb{R}^{r_1 \times \cdots \times r_{n-1} \times N_n \times r_{n+1} \times \cdots \times r_k}$, can be defined by:

$$\mathcal{B} = \mathcal{A} \times_n \boldsymbol{U} \Leftrightarrow \mathcal{B} = \text{fold}_n(\boldsymbol{U} \boldsymbol{A}_{(n)}). \tag{5}$$

The definition can also be written using an element-wise expression, $b_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_k} = \sum_{i_n} a_{i_1 \ldots i_k} u_{j i_n}$. Likewise, the Tucker factorization can be written element-wisely:

$$x_{i_1 \ldots i_k} = \sum_{j_1=1}^{r_1} \cdots \sum_{j_k=1}^{r_k} g_{j_1 \ldots j_k} u_{1,i_1 j_1} \cdots u_{k,i_k j_k}. \tag{6}$$

One natural definition of tensor rank coming with Tucker factorization is the tensor n-rank, defined as the rank of n-mode unfolding matrix of a tensor. The Tucker rank is then defined as the set of n-rank of all unfolding matrices $(\text{rank}_{(1)}(\mathcal{X}), \text{rank}_{(2)}(\mathcal{X}), \ldots, \text{rank}_{(k)}(\mathcal{X}))$, where

$$\text{rank}_{(n)}(\mathcal{X}) = \text{rank}(\boldsymbol{X}_{(n)}). \tag{7}$$

2.3. Problem Definition

We formulate the problem of transport data imputation in a tensorial framework. Given an incomplete tensor $\mathcal{Y}_\Omega = \mathcal{Y} \odot \Omega \in \mathbb{R}^{N_1 \times \cdots \times N_k}$, where $\odot$ is element-wise product, $\mathcal{Y}$ represents the ground truth tensor, and $\Omega \in \{0,1\}^{N_1 \times \cdots \times N_k}$ denotes the binary mask tensor indicating observed entries, our objective is to deduce a restored tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times \cdots \times N_k}$ approximating the true tensor.

3. Imputation Framework

In this section, we first present the idea of using a time series decomposition model as the underlying temporal model of transport data. Then, based on the time series decomposition, we propose a new optimization model combining tensor factorization and rank minimization to enable robust imputation of missing values.

3.1. Time Series Decomposition

In order to effectively model the temporal evolution of time series in transport data, an additive time series decomposition is formulated and later embedded in the proposed framework, as illustrated in Fig. 1. We base the rationale behind the decomposition on the characteristics of actual transport data. An representative traffic volume snippet from a loop detector in London is shown in Fig. 2, where part of observations are missing, as indicated by orange strips. Some major findings include (i) An apparent drop in traffic volume can be noticed around Aug 15, 2019, which is not observed elsewhere. This may signify a long-term supply-side change, such as road construction. Another sharp volume drop is evident around Oct 7, 2019, which quickly recovers to the usual level, possibly due to a temporary lane closure. (ii) Daily periodicity can be easily observed from the plot, with one or two peaks during the daytime and dips at night. (iii) A few spurious extreme values can be seen around Aug 11, 2019 and Aug 14, 2019, which are outliers in need of careful treatment by the imputation model. (iv) Both short-term random gaps and full-day gaps of missing values can be identified throughout the snippet. (v) A notable extended gap of missing values can be spotted

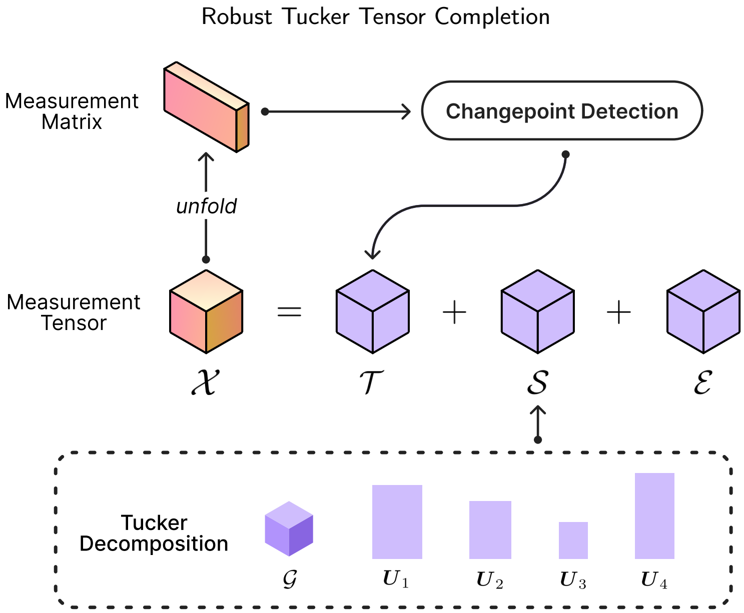


Figure 1: An overview of the time series decomposition. Note that tensors are illustrated as 3-dimensional cubes merely for demonstration purposes; they can be of higher modes to incorporate higher-order information.

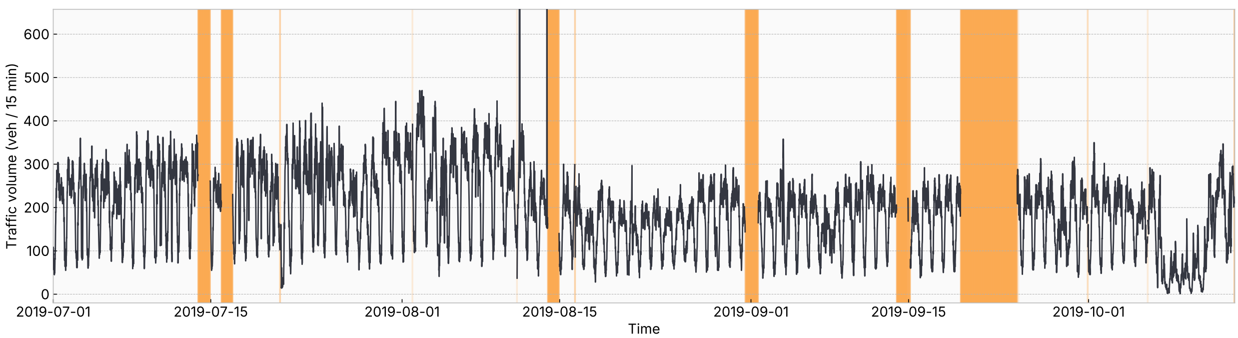


Figure 2: An example time series of traffic volume data collected by a loop detector in London. The orange background in the figure indicate missing values.

in September. This gap is shared across all sensors, indicating a system-wide blackout, possibly due to system upgrade or data loss.

Based on the observations above, we decompose the time series into three components, namely trend, seasonality, and error. The trend component assumes a locally constant trend to provide a stable prior for the imputation. The seasonal component captures the periodicity of the time series and the correlations among sensors, aligning with the low-rank property of the transport data tensor. The error component accounts for the occasional outliers in the time series. To sum up, a transport data tensor can be expressed as the sum of the following three components,

$$\mathcal{X} = \mathcal{T} + \mathcal{S} + \mathcal{E}, \tag{8}$$

where $\mathcal{T}$, $\mathcal{S}$, and $\mathcal{E}$ denote the trend tensor, seasonality tensor, and error tensor, respectively.

The trend tensor is established through a preliminary step of changepoint detection, such that major changes in temporal evolution, e.g., the volume drop in Fig. 2, can be identified. In changepoint detection, we hold the assumption that changepoints occur independently across sensors. Therefore, the tensor is first unfolded back to a matrix, and changepoint detection is performed on time series of each sensor individually. The locations of changepoints in time series are examined using the pruned exact linear time (PELT) algorithm (Killick et al., 2012), where the homogeneity of each segment is measured using the Gaussian kernel function (Arlot et al., 2019). Details about the PELT algorithm is presented in Appendix B.

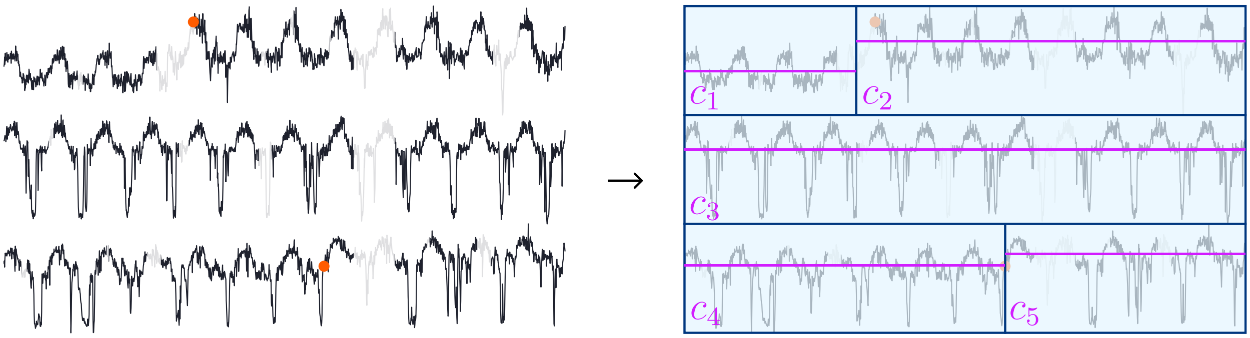


Figure 3: Illustration of time series segmentation from changepoints. Three example time series are shown in the left figure, where the identified changepoints are indicated by red dots and missing values are dimmed in color. Segmentation are performed based on these changepoints, resulting in the piecewise trend model, as shown in the right figure.

The identified changepoints are then used to partition the time series into a sequence of segments. The basic idea behind the trend model is illustrated in Fig. 3. Within each segment, we assume a constant trend component — a piecewise constant trend model — in an effort to enhance the robustness of imputation results. Let us first consider the time series $\boldsymbol{x}$ from an arbitrary sensor. Changepoint detection is applied to the non-missing subseries $\tilde{\boldsymbol{x}}$, resulting in a sequence of changepoints $\boldsymbol{\tau} = (\tau_1, \dots, \tau_{N_\tau})$. To enhance robustness, segmentation is not executed precisely at the identified changepoints. Instead, the average of the indices of the i-th non-missing predecessor and successor around each changepoint is used. For instance, as illustrated in the first sensor of Fig. 3, segments are divided at the midpoint of the missing gap rather than directly at the changepoint. The same procedure is repeated across all sensors to obtain the complete segmentation of the measurement matrix. After segmentation, a unique trend offset c_i, which will be later updated in the optimization model, is placed on the i-th segment, embodying the overall trend and forming the piecewise constant trend model. This approach can effectively capture the mean shift in traffic dynamics. It is worth noting that, different from classical time series decomposition methods like moving average, our trend model is much simplified. It does not involve a moving window, which makes it more stable when handling incomplete data.

The seasonal component of the time series decomposition is managed using a Tucker factorization model, which factorizes the low-rank seasonal tensor into a core tensor and factor matrices, allowing us to effectively capture the spatio-temporal correlations in the data, including the temporal periodicity and cross-sensor correlations. Details of addressing Tucker factorization will be explained in the subsequent subsection. Finally, the additional error component is responsible for managing the occasional outliers in the time series, thereby mitigating their impacts on the imputation results.

3.2. Optimization Problem

To obtain a proper imputation of the missing values, the objective of tensor-based methods is often either minimizing the reconstruction error in the case of tensor factorization, or minimizing the tensor rank when working on the full-sized tensor. Based on the time series decomposition model (Eq. (8)), one may want to directly minimize the error component because of the tensor factorization. However, as reviewed in *Introduction*, the difficulty in determining appropriate ranks for Tucker factorization is one of its major limitations. It is reasonable to apply an additional penalty on large Tucker ranks in order to relieve the burden of rank determination. Additionally, it was suggested by Goulart et al. (2017) that imposing parsimony on the core tensor can help alleviate the adverse effects of misspecified ranks. Thus, the following objective is designed to tackle the aforementioned limitation,

$$\min_{\{\boldsymbol{U}_i\},\mathcal{G},\mathcal{E}} \|\mathcal{E}\|_1 + \mu\|\mathcal{G}\|_1 + \lambda \sum_{i=1}^{k} \operatorname{rank}(\boldsymbol{U}_i) + \frac{\xi}{2} \sum_{i=2}^{k} \|\boldsymbol{D}_i \boldsymbol{U}_i\|_F^2. \tag{9}$$

where $\{\boldsymbol{U}_i\}$ is a shorthand for $\{\boldsymbol{U}_1, \boldsymbol{U}_2, \dots, \boldsymbol{U}_k\}$, and is used for conciseness hereafter.

The objective function consists of four components, including the ℓ_1 norm of the error term, the ℓ_1 norm of the core tensor, the ranks of factor matrices, and the total variation (TV) regularization. The ℓ_1 regularization of both the error term and the core tensor aims to obtain a sparse solution. The last TV regularization term aims to stabilize the tensor completion result by applying a smoothness prior on factor matrices (Wang et al., 2008) with the help of the difference matrix:

$$\boldsymbol{D}_i = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & \cdots & 0 & 0 \\ 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 1 \end{pmatrix} \in \mathbb{R}^{N_i \times N_i}, \tag{10}$$

Note that the TV regularization is only imposed on the temporal factors, as the first mode, i.e., the mode of sensors, is not necessarily smooth in the local neighborhood.

Most tensor completion methods based on rank minimization aim to minimize the sum of the ranks of all unfolding matrices, i.e., $\sum_i \operatorname{rank}(\boldsymbol{S}_{(i)})$ (Liu et al., 2013; Chen et al., 2020; Nie et al., 2022). However, with tensor factorization in the optimization problem, optimizing over unfolding matrices introduces additional difficulty in designing solution algorithms. To circumvent this issue, we directly optimize over the sum of the ranks of all factor matrices, denoted as $\sum_i \operatorname{rank}(\boldsymbol{U}_i)$, which is equivalent to minimizing the sum of ranks of unfolding matrices (Yu et al., 2022).

To ensure the objective function is differentiable, a smooth surrogate for matrix rank is necessary (Nie et al., 2022; Kang et al., 2015). Given that the matrix rank is equal to the number of non-zero values of its singular values, approximations are usually designed in the direction of finding a better surrogate for the ℓ_0 norm. Compared with nuclear norm, which is the convex envelope of matrix rank, nonconvex surrogates, such as truncated nuclear norm and Schatten-p norm, have been shown to provide superior approximations. In our formulation, we employ the γ-norm for a better approximation,

$$\|\boldsymbol{U}\|_\gamma = \sum_i \frac{(1+\gamma)\sigma_i(\boldsymbol{U})}{\gamma + \sigma_i(\boldsymbol{U})}, \tag{11}$$

where $\sigma_i(\boldsymbol{U})$ denote the i-th singular value of factor matrix $\boldsymbol{U}$, and γ is a shape parameter. Substitute the matrix rank in Eq. (9) by the γ-norm above, and the complete optimization problem can be formulated as:

$$\min_{\{\boldsymbol{U}_i\},\mathcal{G},\mathcal{E},\boldsymbol{c},\mathcal{X}} \|\mathcal{E}\|_1 + \mu\|\mathcal{G}\|_1 + \lambda \sum_{i=1}^{k} \|\boldsymbol{U}_i\|_\gamma + \frac{\xi}{2}\sum_{i=2}^{k} \|\boldsymbol{D}_i \boldsymbol{U}_i\|_F^2 \tag{12}$$

$$\text{s.t.} \quad \mathcal{X} = f_{\mathcal{T}}(\boldsymbol{c}) + f_{\mathcal{S}}(\mathcal{G}; \boldsymbol{U}_1, \cdots, \boldsymbol{U}_k) + \mathcal{E} \tag{13}$$

$$\mathcal{X}_\Omega = \mathcal{Y}_\Omega, \tag{14}$$

where the first constraint is the time series decomposition of the transport data tensor. As outlined in Eq. (8), we have $\mathcal{T} = f_{\mathcal{T}}(\boldsymbol{c}) = \sum_i c_i \operatorname{fold}_1(\boldsymbol{\Psi}_i)$ and $\mathcal{S} = f_{\mathcal{S}}(\mathcal{G}; \boldsymbol{U}_1, \cdots, \boldsymbol{U}_k) = \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \cdots \times_k \boldsymbol{U}_k$. The trend function $\mathcal{T}$ connects the changepoints identified by the PELT algorithm with the decision variables $\boldsymbol{c}$. Each time series segment partitioned by changepoints is assigned a variable c_i, which represents the corresponding long-term trend. Additionally, a binary masking matrix, $\boldsymbol{\Psi}_i \in \{0,1\}^{N_s \times N_t}$, is aligned with each segment. The seasonal term $\mathcal{S}$ here is substituted by the Tucker factorization. The last constraint ensures that the values of all observed entries are identical to the imputation results.

4. Solution Algorithm

While the optimization problem above with two equality constraints can be solved using gradient descent by absorbing the constraints into the objective function, it can be computationally inefficient. Fortunately, it is feasible to separate the objective as the sum of several decoupled functions, allowing acceleration with ADMM (Boyd, 2010). The basic idea behind ADMM is to decompose the problem into smaller subproblems, each of which can be solved independently and iteratively.

4.1. Problem Separation

To enable successful separation, auxiliary variables $\{\boldsymbol{V}_i\}$ are introduced to decouple the third and last components in the objective, resulting in the modified problem,

$$\min_{\{\boldsymbol{U}_i\},\mathcal{G},\mathcal{E},\boldsymbol{c},\mathcal{X}} \|\mathcal{E}\|_1 + \mu\|\mathcal{G}\|_1 + \lambda \textstyle\sum_{i=1}^{k} \|\boldsymbol{V}_i\|_\gamma + \frac{\xi}{2}\textstyle\sum_{i=2}^{k} \|\boldsymbol{D}_i \boldsymbol{U}_i\|_F^2 \tag{15}$$

$$\text{s.t.} \quad \mathcal{X} = f_{\mathcal{T}}(\boldsymbol{c}) + f_{\mathcal{S}}(\mathcal{G}; \boldsymbol{U}_1, \cdots, \boldsymbol{U}_k) + \mathcal{E} \tag{16}$$
$$\mathcal{X}_\Omega = \mathcal{Y}_\Omega \tag{17}$$
$$\boldsymbol{U}_i = \boldsymbol{V}_i. \tag{18}$$

The problem is now separable with four decoupled blocks in the objective. Denote the Lagrangian multipliers by $\mathcal{M}$ and $\boldsymbol{\Gamma}$, the augmented Lagrangian is given as follows:

$$\begin{aligned} L(\{\boldsymbol{U}_i, \boldsymbol{V}_i, \boldsymbol{\Gamma}_i\}, \mathcal{G}, \mathcal{E}, \boldsymbol{c}, \mathcal{X}, \mathcal{M}) = \|\mathcal{E}\|_1 + \mu\|\mathcal{G}\|_1 + \lambda \sum_{i=1}^k \|\boldsymbol{V}_i\|_\gamma + \frac{\xi}{2} \sum_{i=2}^k \|\boldsymbol{D}_i \boldsymbol{U}_i\|_F^2 \\ + \frac{\rho}{2}\|\mathcal{X} - f_{\mathcal{T}}(\boldsymbol{c}) - f_{\mathcal{S}}(\mathcal{G}; \boldsymbol{U}_1, \cdots, \boldsymbol{U}_k) - \mathcal{E} + \mathcal{M}\|_F^2 \\ + \sum_{i=1}^k \frac{\rho}{2}\|\boldsymbol{U}_i - \boldsymbol{V}_i + \boldsymbol{\Gamma}_i\|_F^2. \end{aligned} \tag{19}$$

4.2. Bregman ADMM Algorithm

While the problem has become separable after modification, the convergence of standard ADMM, initially designed for two-block problems, is not guaranteed when extended to a higher number of blocks (Chen et al., 2016). To overcome this challenge, we incorporate the Bregman divergence in the subproblems, which has been demonstrated to effectively help optimization convergence (Bauschke et al., 2017; Wang et al., 2018). In this section, we first show how each subproblem can be solved under the Bregman ADMM framework, and then conclude with the complete algorithm.

4.2.1. Solving U-Subproblems

We begin with the first factor matrix $\boldsymbol{U}_1$, and the subproblem can be described as:

$$\boldsymbol{U}_1^{(t+1)} = \arg\min_{\boldsymbol{U}_1} L(\{\boldsymbol{U}_{i\neq 1}^{(t)}, \boldsymbol{V}_i^{(t)}, \boldsymbol{\Gamma}_i^{(t)}\}, \boldsymbol{U}_1, \mathcal{G}^{(t)}, \mathcal{E}^{(t)}, \boldsymbol{c}^{(t)}, \mathcal{X}^{(t)}, \mathcal{M}^{(t)}) + \frac{\eta}{2}\|\boldsymbol{U}_1 - \boldsymbol{U}_1^{(t)}\|_F^2 \tag{20}$$
$$= \arg\min_{\boldsymbol{U}_1} \frac{\rho}{2}\|\mathcal{Z} - f_{\mathcal{S}}(\mathcal{G}^{(t)}; \boldsymbol{U}_1, \cdots, \boldsymbol{U}_k^{(t)})\|_F^2 + \frac{\rho}{2}\|\boldsymbol{U}_1 - \boldsymbol{V}_1^{(t)} + \boldsymbol{\Gamma}_1^{(t)}\|_F^2 + \frac{\eta}{2}\|\boldsymbol{U}_1 - \boldsymbol{U}_1^{(t)}\|_F^2, \tag{21}$$

where $\eta > 0$ is a coefficient of the Bregman divergence, defined as the Frobenius norm of the updating difference, and $\mathcal{Z} = \mathcal{X}^{(t)} - f_{\mathcal{T}}(\boldsymbol{c}^{(t)}) - \mathcal{E}^{(t)} + \mathcal{M}^{(t)}$. All variables excluding $\boldsymbol{U}_1$ are fixed, resulting in a convex problem. The solution can be easily derived by letting the gradient of the subproblem objective be zero:

$$\boldsymbol{U}_1^{(t+1)} = (\rho \boldsymbol{Z}_{(1)} \boldsymbol{P}_1 (\boldsymbol{G}_{(1)}^{(t)})^\top - \rho \boldsymbol{Q}_1 - \eta \boldsymbol{U}_1^{(t)})(\rho \boldsymbol{G}_{(1)}^{(t)} \boldsymbol{P}_1^\top \boldsymbol{P}_1 (\boldsymbol{G}_{(1)}^{(t)})^\top - (\rho + \eta)\boldsymbol{I})^{-1}, \tag{22}$$

where $\boldsymbol{P}_1 = \boldsymbol{U}_2^{(t)} \otimes \cdots \otimes \boldsymbol{U}_k^{(t)}$ and $\boldsymbol{Q}_1 = \boldsymbol{V}_1^{(t)} - \boldsymbol{\Gamma}_1^{(t)}$. The operator $\otimes$ here denotes the Kronecker product.

A simple closed-form solution for the subproblems of other factor matrices is unfortunately unavailable because of the additional TV regularizer. For an arbitrary factor matrix $\boldsymbol{U}_m$ $(2 \leqslant m \leqslant k)$, its subproblem will be:

$$\boldsymbol{U}_m^{(t+1)} = \arg\min_{\boldsymbol{U}_m} L(\{\boldsymbol{U}_{i<m}^{(t+1)}, \boldsymbol{U}_{i>m}^{(t)}, \boldsymbol{V}_i^{(t)}, \boldsymbol{\Gamma}_i^{(t)}\}, \boldsymbol{U}_m, \mathcal{G}^{(t)}, \mathcal{E}^{(t)}, \boldsymbol{c}^{(t)}, \mathcal{X}^{(t)}, \mathcal{M}^{(t)}) + \frac{\eta}{2}\|\boldsymbol{U}_m - \boldsymbol{U}_m^{(t)}\|_F^2 \tag{23}$$
$$\begin{aligned} = \arg\min_{\boldsymbol{U}_m} \frac{\rho}{2}\|\mathcal{Z} - f_{\mathcal{S}}(\mathcal{G}^{(t)}; \boldsymbol{U}_1^{(t+1)}, \cdots, \boldsymbol{U}_k^{(t)})\|_F^2 + \frac{\xi}{2}\|\boldsymbol{D}_m \boldsymbol{U}_m\|_F^2 \\ + \frac{\rho}{2}\|\boldsymbol{U}_m - \boldsymbol{V}_m^{(t)} + \boldsymbol{\Gamma}_m^{(t)}\|_F^2 + \frac{\eta}{2}\|\boldsymbol{U}_m - \boldsymbol{U}_m^{(t)}\|_F^2. \end{aligned} \tag{24}$$

Again, letting the gradient of the subproblem objective be zero, it can be simplified to a Sylvester equation:

$$\boldsymbol{A}\boldsymbol{U}_m + \boldsymbol{U}_m \boldsymbol{B} = \boldsymbol{R}, \tag{25}$$

where $\boldsymbol{A} = \rho \boldsymbol{I} + \xi \boldsymbol{D}_m^\top \boldsymbol{D}_m$, $\boldsymbol{B} = \eta \boldsymbol{I} + \rho \boldsymbol{G}_{(i)}^{(t)} \boldsymbol{P}_m^\top \boldsymbol{P}_m (\boldsymbol{G}_{(i)}^{(t)})^\top$, and $\boldsymbol{R} = \rho \boldsymbol{Q}_m + \eta \boldsymbol{U}_m^{(t)} + \rho \boldsymbol{Z}_{(i)} \boldsymbol{P}_m (\boldsymbol{G}_{(i)}^{(t)})^\top$. Herein, $\boldsymbol{P}_m = \bigotimes_{i\neq m} \boldsymbol{U}_i^{(t)}$, and $\boldsymbol{Q}_m = \boldsymbol{V}_m^{(t)} - \boldsymbol{\Gamma}_m^{(t)}$. The Sylvester equation can be solved using the Bartels-Stewart algorithm or iterative methods like Krylov subspace methods (Higham, 2002). Details on its solution algorithms can be found in Appendix C.

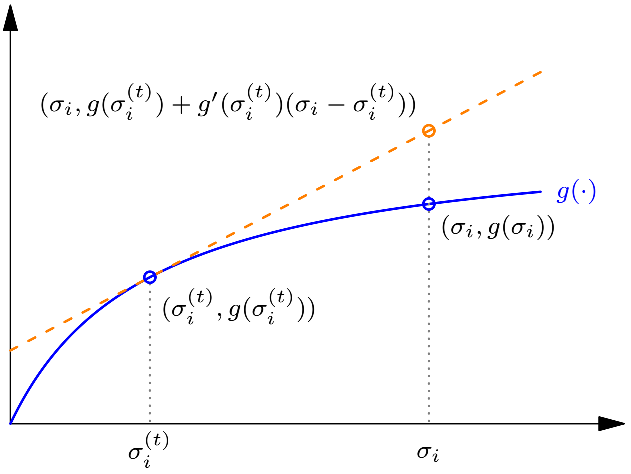


Figure 4: Linearization of γ-norm

4.2.2. Solving V-Subproblems

The subproblem of an arbitrary $\boldsymbol{V}_m$ $(1 \leqslant m \leqslant k)$ can be described as:

$$\boldsymbol{V}_m^{(t+1)} = \arg\min_{\boldsymbol{V}_m} L(\{\boldsymbol{V}_{i<m}^{(t+1)}, \boldsymbol{V}_{i>m}^{(t)}, \boldsymbol{U}_i^{(t+1)}, \boldsymbol{\Gamma}_i^{(t)}\}, \boldsymbol{V}_m, \mathcal{G}^{(t)}, \mathcal{E}^{(t)}, \boldsymbol{c}^{(t)}, \mathcal{X}^{(t)}, \mathcal{M}^{(t)}) + \frac{\eta}{2}\|\boldsymbol{V}_m - \boldsymbol{V}_m^{(t)}\|_F^2 \tag{26}$$

$$= \arg\min_{\boldsymbol{V}_m} \lambda\|\boldsymbol{V}_m\|_\gamma + \frac{\rho}{2}\|\boldsymbol{V}_m - \boldsymbol{U}_m^{(t+1)} - \boldsymbol{\Gamma}_m^{(t)}\|_F^2 + \frac{\eta}{2}\|\boldsymbol{V}_m - \boldsymbol{V}_m^{(t)}\|_F^2 \tag{27}$$

$$= \arg\min_{\boldsymbol{V}_m} \frac{\lambda}{\rho+\eta}\|\boldsymbol{V}_m\|_\gamma + \frac{1}{2}\|\boldsymbol{V}_m - \frac{1}{\rho+\eta}(\rho(\boldsymbol{U}_m^{(t+1)} + \boldsymbol{\Gamma}_m^{(t)}) + \eta\boldsymbol{V}_m^{(t)}\|_F^2 \tag{28}$$

$$= \arg\min_{\boldsymbol{V}_m} \frac{\lambda}{\rho+\eta}\|\boldsymbol{V}_m\|_\gamma + \frac{1}{2}\|\boldsymbol{V}_m - \boldsymbol{W}_m\|_F^2, \tag{29}$$

where $\boldsymbol{W}_m = \frac{1}{\rho+\eta}(\rho(\boldsymbol{U}_m^{(t+1)} + \boldsymbol{\Gamma}_m^{(t)}) + \eta\boldsymbol{V}_m^{(t)})$.

The objective of the subproblem is non-convex and has been simplified as the sum of a γ-norm and a Frobenius norm. It is well-known that the nuclear norm minimization problem can be solved using the singular value thresholding (SVT) method (Cai et al., 2010), where the nuclear norm is written as the ℓ_1-norm of the singular value vector. An extension to weighted ℓ_1-norm of singular value vector, i.e., weighted SVT (WSVT) (Chen et al., 2013; Lu et al., 2014), further allows the adoption of a broader range of norms like truncated nuclear norm (Chen et al., 2020). While the γ-norm is not directly a weighted ℓ_1-norm of the singular value vector, it can be relaxed via linearization.

Lemma 1. *Let σ_i denote the i-th singular value of a matrix $\boldsymbol{V}$. Given the following function:*

$$h(\boldsymbol{V}) = \frac{\lambda}{\rho+\eta}\sum_{i=1}^k g(\sigma_i) + \frac{1}{2}\|\boldsymbol{V} - \boldsymbol{W}\|_F^2, \tag{30}$$

where $g(\sigma_i) = (1+\gamma)\sigma_i/(\gamma+\sigma_i)$ is concave, and its derivative $g'(\sigma_i) = (1+\gamma)\gamma/(\gamma+\sigma_i)^2$ is monotonically decreasing. It can be relaxed to:

$$\tilde{h}(\boldsymbol{V}) = \frac{\lambda}{\rho+\eta}\sum_{i=1}^k \left(g(\sigma_i^{(t)}) + g'(\sigma_i^{(t)})(\sigma_i - \sigma_i^{(t)})\right) + \frac{1}{2}\|\boldsymbol{V} - \boldsymbol{W}\|_F^2 \tag{31}$$

$$= \frac{\lambda}{\rho+\eta}\sum_{i=1}^k g'(\sigma_i^{(t)})\sigma_i + \frac{1}{2}\|\boldsymbol{V} - \boldsymbol{W}\|_F^2 + C, \tag{32}$$

where $C = \lambda/(\rho+\eta)\sum_i(g(\sigma_i^{(t)}) - g'(\sigma_i^{(t)})\sigma_i^{(t)})$ is constant with respect to $\boldsymbol{V}$.

With Lemma 1, the γ-norm of $\boldsymbol{V}_m$ can be relaxed to its weighted ℓ_1-norm, as illustrated in Fig. 4. Then, the $\boldsymbol{V}_m$ subproblem can be solved by WST. Specifically, let the singular value decomposition (SVD) of $\boldsymbol{W}_m$ be $\boldsymbol{A\Sigma B}^\top$, the update rule is given by,

$$\boldsymbol{V}_m^{(t+1)} = \boldsymbol{A}\mathscr{T}_{\frac{\lambda}{\rho+\eta}\boldsymbol{\omega}}(\boldsymbol{\Sigma})\boldsymbol{B}^\top, \tag{33}$$

where $\boldsymbol{\omega} = \{g'(\sigma_i^{(t)})\}$. The WST operator is defined by $\mathscr{T}_{\frac{\lambda}{\rho+\eta}\boldsymbol{\omega}}(\boldsymbol{\Sigma}) = (\boldsymbol{\Sigma} - \frac{\lambda}{\rho+\eta}\boldsymbol{\omega}\boldsymbol{I})_+$, where $(\cdot)_+ = \max(\cdot, 0)$.

4.2.3. Solving $\mathcal{G}$-Subproblem

The subproblem of $\mathcal{G}$ can be written and simplified as:

$$\mathcal{G}^{(t+1)} = \arg\min_{\mathcal{G}} L(\{\boldsymbol{U}_i^{(t+1)}, \boldsymbol{V}_i^{(t+1)}, \boldsymbol{\Gamma}_i^{(t)}\}, \mathcal{G}, \mathcal{E}^{(t)}, \boldsymbol{c}^{(t)}, \mathcal{X}^{(t)}, \mathcal{M}^{(t)}) + \frac{\eta}{2}\|\mathcal{G} - \mathcal{G}^{(t)}\|_F^2 \tag{34}$$

$$= \arg\min_{\mathcal{G}} \|\mathcal{G}\|_1 + \frac{\rho}{2\mu}\|f_{\mathcal{S}}(\mathcal{G}; \boldsymbol{U}_1^{(t+1)}, \cdots, \boldsymbol{U}_k^{(t+1)}) - \mathcal{Z}\|_F^2 + \frac{\eta}{2\mu}\|\mathcal{G} - \mathcal{G}^{(t)}\|_F^2. \tag{35}$$

Direct analysis with n-mode product is inconvenient. Hence, the objective of this subproblem is rewritten in the matrix form using Eq. (5). It can be noticed that the subproblem is a generalized ℓ_1-minimization problem:

$$\mathcal{G}^{(t+1)} = \arg\min_{\mathcal{G}} \ \|\mathcal{G}\|_1 + \phi(\mathcal{G}) \quad \Leftrightarrow \quad \boldsymbol{G}_{(1)}^{(t+1)} = \arg\min_{\boldsymbol{G}_{(1)}} \ \|\boldsymbol{G}_{(1)}\|_1 + \varphi(\boldsymbol{G}_{(1)}). \tag{36}$$

The last two terms in Eq. (35) is represented by $\phi(\mathcal{G})$. Without loss of generality, the tensors are unfolded along the first mode, resulting in its matrix form $\varphi(\boldsymbol{G}_{(1)})$, along with the gradient $\varphi'(\boldsymbol{G}_{(1)})$:

$$\varphi(\boldsymbol{G}_{(1)}) = \frac{1}{2\mu}\left(\rho\|\boldsymbol{U}_1^{(t+1)}\boldsymbol{G}_{(1)}\boldsymbol{P}_1^\top - \boldsymbol{Z}_{(1)}\|_F^2 + \eta\|\boldsymbol{G}_{(1)} - \boldsymbol{G}_{(1)}^{(t)}\|_F^2\right) \tag{37}$$

$$\varphi'(\boldsymbol{G}_{(1)}) = \frac{1}{\mu}\left(\rho(\boldsymbol{U}_1^{(t+1)})^\top(\boldsymbol{U}_1^{(t+1)}\boldsymbol{G}_{(1)}\boldsymbol{P}_1^\top - \boldsymbol{Z}_{(1)})\boldsymbol{P}_1 + \eta(\boldsymbol{G}_{(1)} - \boldsymbol{G}_{(1)}^{(t)})\right), \tag{38}$$

where $\boldsymbol{P}_1 = \bigotimes_{i=2}^k \boldsymbol{U}_i^{(t+1)}$.

Considering that function $\varphi(\cdot)$ is differentiable and convex, the update rule can be given by the soft thresholding (ST) method (Yin et al., 2008; Boyd, 2010):

$$\boldsymbol{G}_{(1)}^{(t+1)} = \mathrm{sgn}(\boldsymbol{H}) \odot (|\boldsymbol{H}| - \delta)_+, \tag{39}$$

where $\boldsymbol{H} = \boldsymbol{G}_{(1)}^{(t)} - \delta\varphi'(\boldsymbol{G}_{(1)}^{(t)})$, and $\odot$ is the Hadamard product. We empirically set the reciprocal of the step size $1/\delta$ as $(\rho\|\boldsymbol{U}_1^{(t+1)\top}\boldsymbol{U}_1^{(t+1)}\|_2\|\boldsymbol{P}_1^\top\boldsymbol{P}_1\|_2 + \eta)/\mu$, which is bounded by the Lipschitz constant of $\varphi'(\boldsymbol{G}_{(1)}^{(t)})$, as suggested by Hale et al. (2008), where $\|\cdot\|_2$ denotes the spectral norm.

4.2.4. Solving Other Subproblems

Upon updating the factor matrices and the core tensor, the updated seasonal component can be computed by $\mathcal{S} = f_{\mathcal{S}}(\mathcal{G}^{(t+1)}; \boldsymbol{U}_1^{(t+1)}, \cdots, \boldsymbol{U}_k^{(t+1)})$. Then, the $\mathcal{E}$ subproblem can be simplified to a ℓ_1-minimization problem:

$$\mathcal{E}^{(t+1)} = \arg\min_{\mathcal{E}} L(\{\boldsymbol{U}_i^{(t+1)}, \boldsymbol{V}_i^{(t+1)}, \boldsymbol{\Gamma}_i^{(t)}\}, \mathcal{G}^{(t+1)}, \mathcal{E}, \boldsymbol{c}^{(t)}, \mathcal{X}^{(t)}, \mathcal{M}^{(t)}) + \frac{\eta}{2}\|\mathcal{E} - \mathcal{E}^{(t)}\|_F^2 \tag{40}$$

$$= \arg\min_{\mathcal{E}} \|\mathcal{E}\|_1 + \frac{\rho+\eta}{2}\|\mathcal{E} - \mathcal{B}\|_F^2, \tag{41}$$

where $\mathcal{B} = \frac{1}{\rho+\eta}\left(\rho\left(\mathcal{X}^{(t)} - f_{\mathcal{T}}(\boldsymbol{c}^{(t)}) - \mathcal{S} + \mathcal{M}^{(t)}\right) + \eta\mathcal{E}^{(t)}\right)$. The update rule can be given by the ST method:

$$\mathcal{E}^{(t+1)} = \mathrm{sgn}(\mathcal{B}) \odot \left(|\mathcal{B}| - \frac{1}{\rho+\eta}\right)_+. \tag{42}$$

The $\boldsymbol{c}$ subproblem can be solved for each of its elements independently as they are separated by masks $\{\boldsymbol{\Psi}_i\}$, corresponding to one time series segment each. The update rule of an arbitrary c_m can be derived by:

$$c_m^{(t+1)} = \arg\min_{c_m} \; L(\{\boldsymbol{U}_i^{(t+1)}, \boldsymbol{V}_i^{(t+1)}, \boldsymbol{\Gamma}_i^{(t)}\}, \mathcal{G}^{(t+1)}, \mathcal{E}^{(t+1)}, c_m, \mathcal{X}^{(t)}, \mathcal{M}^{(t)}) + \frac{\eta}{2}(c_m - c_m^{(t)})^2 \tag{43}$$

$$= \arg\min_{c_m} \; \frac{\rho}{2}\|c_m \operatorname{fold}_1(\boldsymbol{\Psi}_m) - (\mathcal{X}^{(t)} - \mathcal{S} - \mathcal{E}^{(t+1)} + \mathcal{M}^{(t)})\|_F^2 + \frac{\eta}{2}(c_m - c_m^{(t)})^2 \tag{44}$$

$$= \frac{1}{\rho + \eta}(\rho \bar{\psi}_m + \eta c_m^{(t)}), \tag{45}$$

where $\bar{\psi}_m = \sum_{uv}(\boldsymbol{\Psi}_m \odot (\boldsymbol{X}_{(1)}^{(t)} - \boldsymbol{S}_{(1)} - \boldsymbol{E}_{(1)}^{(t+1)} + \boldsymbol{M}_{(1)}^{(t)}))_{uv} / \|\boldsymbol{\Psi}_m\|_1$. And then we have $\mathcal{T} = f_{\mathcal{T}}(\boldsymbol{c}^{(t+1)})$.

A closed-form solution also exists for the $\mathcal{X}$ subproblem. Note that only the unobserved entries (indicated by Ω^-) are updated due to the constraint Eq. (17):

$$\mathcal{X}_{\Omega^-}^{(t+1)} = \arg\min_{\mathcal{X}} \; L(\{\boldsymbol{U}_i^{(t+1)}, \boldsymbol{V}_i^{(t+1)}, \boldsymbol{\Gamma}_i^{(t)}\}, \mathcal{G}^{(t+1)}, \mathcal{E}^{(t+1)}, \boldsymbol{c}^{(t+1)}, \mathcal{X}, \mathcal{M}^{(t)}) + \frac{\eta}{2}(\mathcal{X} - \mathcal{X}^{(t)})^2 \tag{46}$$

$$= \arg\min_{\mathcal{X}} \; \frac{\rho}{2}\|\mathcal{X} - (\mathcal{T} + \mathcal{S} + \mathcal{E}^{(t+1)} - \mathcal{M}^{(t)}))\|_F^2 + \frac{\eta}{2}(\mathcal{X} - \mathcal{X}^{(t)})^2 \tag{47}$$

$$= \frac{1}{\rho + \eta}\left(\rho(\mathcal{T} + \mathcal{S} + \mathcal{E}^{(t+1)} - \mathcal{M}^{(t)}) + \eta \mathcal{X}^{(t)}\right). \tag{48}$$

Finally, the overall algorithm is concluded by Algorithm 1. Before updating variables, we first initialize $\mathcal{X}$ using historical average. Then, $\mathcal{G}$ and all $\boldsymbol{U}_i$ are computed using higher-order SVD (Kolda and Bader, 2009). The auxiliary variables $\boldsymbol{V}_i$ are set identical to $\boldsymbol{U}_i$. Other variables, including $\mathcal{E}$, $\mathcal{M}$ and all $\boldsymbol{\Gamma}_i$, are initialized as zero.

Algorithm 1 Bregman ADMM for Robust Transport Data Imputation

Input: Incomplete tensor $\mathcal{Y}_\Omega$; missing mask Ω; hyperparameters $\mu, \lambda, \xi, \{r_i\}, \rho, \eta$; convergence threshold ϵ.
Output: Imputed tensor $\mathcal{X}$.

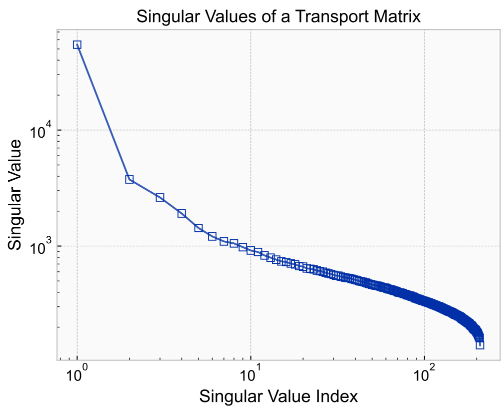
```
Initialize primal variables and Lagrangian multipliers; set tolerance Δ ← ∞; set t ← 1.
repeat
    for i ← 1, …, k do
        Update U_i using Eq. (22) or Eq. (25).
    end for
    for i ← 1, …, k do
        Update V_i using Eq. (33).
    end for
    Update 𝒢 using Eq. (39).
    Compute 𝒮 ← f_𝒮(𝒢^(t+1); U_1^(t+1), ⋯, U_k^(t+1)).
    Update ℰ using Eq. (42).
    Update c using Eq. (45).
    Compute 𝒯 ← f_𝒯(c^(t+1)).
    X̄ ← 𝒳. Update 𝒳 using Eq. (48).
    Update Lagrangian multiplier ℳ ← ℳ + 𝒳 − 𝒯 − 𝒮 − ℰ.
    for i ← 1, …, k do
        Update Lagrangian multiplier Γ_i ← Γ_i + U_i − V_i.
    end for
    t ← t + 1.
    Δ ← ‖𝒳 − X̄‖_F^2 / ‖X̄‖_F^2.
until Δ < ε
```

5. Experiment Settings

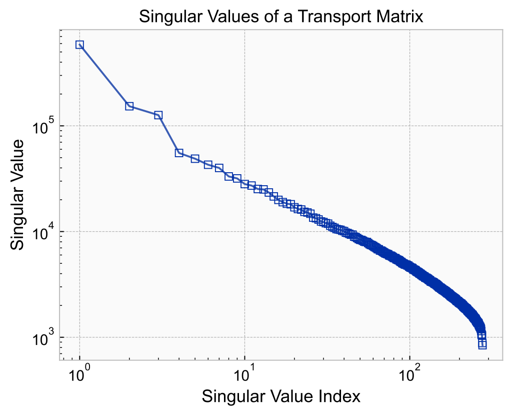
To evaluate the imputation performance of the proposed RTTC, four datasets are employed in this study. This section presents a detailed overview of the metadata and basic statistics of these datasets, as well as the preprocessing

Table 1
Metadata of Datasets

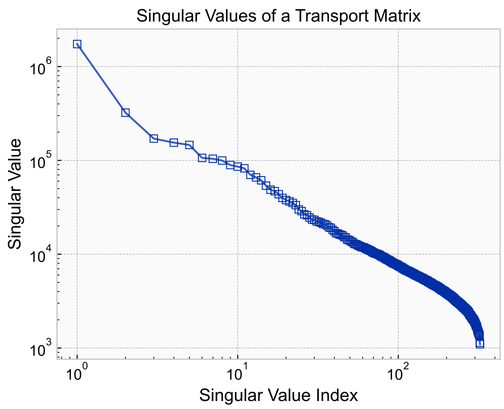
City	# Sensors	Duration	Interval	Measurements	Tensor Size
Guangzhou	209	Aug 1, 2016 — Sep 30, 2016	10 min	Speed (km/h)	$209 \times 9 \times 7 \times 144$
London	276	Jul 1, 2019 — Oct 13, 2019	15 min	Volume (veh/15min)	$276 \times 15 \times 7 \times 96$
Madrid	324	Jun 1, 2021 — Sep 13, 2021	15 min	Volume (veh/15min)	$324 \times 15 \times 7 \times 96$
Melbourne	313	Jun 1, 2020 — Sep 13, 2020	15 min	Volume (veh/15min)	$313 \times 15 \times 7 \times 96$



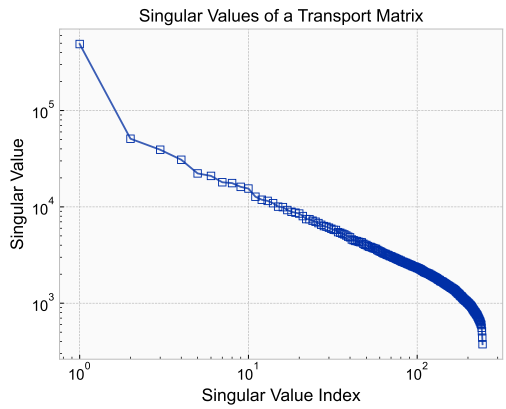
(a) Guangzhou



(b) London



(c) Madrid



(d) Melbourne

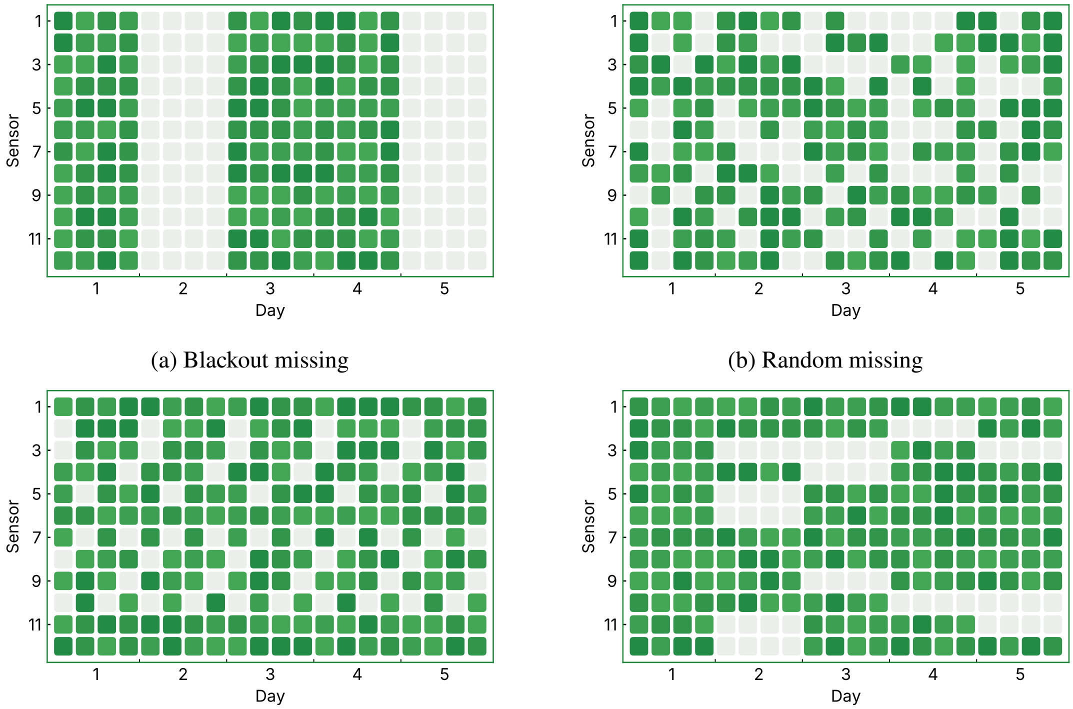
Figure 5: Singular values of transport data from four cities

methods. Furthermore, we provide the details of the experimental settings, including the design of missing patterns, base models for comparison, and configuration of hyperparameters.

5.1. Data Description

The datasets used in this study include one traffic speed data from Guangzhou (China) and three traffic volume data from London (UK), Madrid (Spain), and Melbourne (Australia), respectively[1]. To ensure the reliability of our analysis, we remove any traffic sensor data with less than 95% completeness, excluding time periods where all traffic detectors have no reading. For the traffic volume data, we select sensors located in the city centers, which typically experience higher traffic densities. Detailed metadata for all four datasets can be found in Table 1.

[1] The Guangzhou data is available at `https://zenodo.org/record/1205229`. The other three datasets were gathered by the NeurIPS Traffic4cast 2022 Challenge at `https://www.iarai.ac.at/traffic4cast/challenge/`.



(a) Blackout missing

(b) Random missing

(c) Day missing

(d) Time missing

Figure 6: Typical examples of missing patterns. A simplified matrix of 12×20 is used for demonstration, where each sensor has four records per day. The green cells indicate observed values, while the gray ones are missing values.

The singular values of the data matrices of all four cities are demonstrated in Fig. 5. The dominance of large singular values across all datasets is evident, substantiating the rationality of exploiting the low-rank property for missing value imputation. Furthermore, for the proposed framework, the data matrices are organized into four-way tensors $\mathcal{Y} \in \mathbb{R}^{N_s \times N_w \times N_d \times N_t}$ with the structure of sensor $\times$ week $\times$ day-of-the-week $\times$ time-of-the-day.

5.2. Missing Patterns

Contrary to the majority of studies, which typically create individual missing scenarios for each pattern, our experiment comprises a composite missing mask by integrating four fundamental missing patterns. The mixing-up of missing patterns can be expressed formally as:

$$\Omega = \Omega_{\mathrm{BM}} \odot \Omega_{\mathrm{RM}} \odot \Omega_{\mathrm{DM}} \odot \Omega_{\mathrm{TM}}, \tag{49}$$

where BM refers to the *blackout missing* scenario; RM refers to *random missing* scenario; DM refers to *day missing*; and TM refers to *time missing*. These missing patterns can all be observed in the test data, as illustrated in Fig. 6. BM represents scenarios where an entire day's data is missing across all sensors, akin to a complete system blackout. This is a crucial test case for transport data, as it simulates extreme situations like system-wide failures or data collection interruptions (Chen et al., 2022). RM is the most common case in transport data where traffic state readings are randomly unavailable due to sensor failures, communication issues, or other unpredictable factors. DM involves missing data in the same time across days, whereas TM is characterized by missing values of all time slots in a day (Nie et al., 2022). The DM pattern can often be seen in sensors that are turned off during nighttime.

5.3. Base Models and Performance Metrics

In our experiment, we compare the proposed RTTC with several simple baselines and state-of-the-art tensor-based transport data imputation models. The detailed hyperparameter settings of all the models can be found in Appendix A.

- **Historical average (HA)**. HA is a naïve baseline that averages the observed values over each time-of-the-day.

Table 2
Missing Rates of All Evaluation Scenarios

Scenarios	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BM	10%	10%	10%	10%	10%	10%	10%	10%	30%	30%	30%	30%	30%	30%	30%	30%
RM	10%	10%	10%	10%	30%	30%	30%	30%	10%	10%	10%	10%	30%	30%	30%	30%
DM	10%	10%	30%	30%	10%	10%	30%	30%	10%	10%	30%	30%	10%	10%	30%	30%
TM	10%	30%	10%	30%	10%	30%	10%	30%	10%	30%	10%	30%	10%	30%	10%	30%
Overall	34%	48%	49%	60%	48%	60%	60%	69%	52%	62%	62%	71%	62%	71%	71%	77%

- **Low-Rank Tensor Completion with Truncated Schattern-p Norm (LRTC-TSpN)**. LRTC-TSpN is the state-of-the-art transport data imputation model based on rank minimization paradigm (Nie et al., 2022).
- **Bayesian Gaussian CANDECOMP/PARAFAC Decomposition (BGCP)**. BGCP is a Bayesian model based on tensor factorization paradigm, which extends the Bayesian probabilistic matrix factorization model (Salakhutdinov and Mnih, 2008) from matrix to tensor (Chen et al., 2019).
- **Temporal Regularized Matrix Factorization (TRMF)**. TRMF is a matrix factorization-based model with an integrated autoregressive model for temporal modeling (Yu et al., 2016).
- **Low-Rank Autoregressive Tensor Completion (LATC)**. LATC is a rank minimization-based model with an integrated autoregressive model for temporal modeling (Chen et al., 2022).

Two performance metrics, including an absolute metric (mean absolute error, MAE) and a relative metric (symmetric mean absolute percentage error, SMAPE), are used to quantify the imputation quality of the models, as defined below:

$$\text{MAE} = \frac{1}{|I(\tilde{\Omega})|}\sum_{i\in I(\tilde{\Omega})}|x_i - y_i|, \quad \text{SMAPE} = \frac{1}{|I(\tilde{\Omega})|}\sum_{i\in I(\tilde{\Omega})}\frac{|x_i - y_i|}{|x_i| + |y_i|}\times 100\%, \tag{50}$$

where $I(\tilde{\Omega})$ is the index set of testing entries given the mask $\tilde{\Omega}$.

6. Experiment Results

In this section, the proposed imputation framework is first compared with base models on various evaluation scenarios, including some of extreme missing rates. Additionally, the sensitivity of hyperparameters on the imputation performance is examined through sensitivity analysis. Moreover, the computation time of different models are compared with each other.

6.1. Imputation Performance

The performance of imputation models is first evaluated on 16 composite missing scenarios. Each of these missing scenarios is subjected to two basic missing rates, namely, 10% and 30%. Consequently, this yields a total of 16 unique missing scenarios. The resultant composite missing rates, a product of the combination of these patterns, vary between 34% and 77%. A complete list of all missing scenarios for evaluation and the corresponding overall missing rate is shown in Table 2. And the imputation performance of all the models on 16 missing scenarios in the four cities are listed in Table 3.

On a general note, the proposed RTTC achieves the lowest error in most of the scenarios, particularly, with a higher advantage over base models in London and Madrid as well as scenarios with higher missing rates. It is worth noting that some state-of-the-art imputation models, despite exemplary performance on individual missing scenarios substantiated by prior literature and empirical observations, can suffer from compromised reliability in the presence of intricate composite missing scenarios, resulting in an imputation error higher than that of historical average.

The performance of the proposed RTTC is consistently better than base models on London and Madrid datasets. Larger missing rates usually result in lower imputation accuracy due to less observed information, conforming with the error values in the tables. Nevertheless, a smaller degradation can be observed for the proposed model, even when faced

Table 3
Imputation Performance of All Models in 16 Composite Missing Scenarios

Scenario	BM	RM	DM	TM	HA	LRTC-TSpN	TRMF	BGCP	LATC	RTTC (Ours)
Guangzhou										
1	10%	10%	10%	10%	4.18/6.26	3.66/5.47	3.17/4.92	2.82/4.44	**2.61/4.10**	2.64/4.18
2	10%	10%	10%	30%	3.94/5.95	3.29/4.99	3.16/4.91	2.86/4.49	**2.68/4.21**	2.73/4.31
3	10%	10%	30%	10%	5.22/7.52	3.19/4.84	3.13/4.86	2.77/4.35	**2.58/4.06**	2.62/4.14
4	10%	10%	30%	30%	4.87/7.09	3.07/4.69	3.17/4.94	2.80/4.40	**2.66/4.19**	2.67/4.23
5	10%	30%	10%	10%	3.94/5.95	3.16/4.80	3.12/4.86	2.73/4.29	**2.55/4.03**	2.59/4.10
6	10%	30%	10%	30%	3.84/5.82	3.06/4.68	3.17/4.93	2.79/4.40	**2.64/4.16**	2.69/4.24
7	10%	30%	30%	10%	4.87/7.09	3.00/4.60	3.14/4.90	2.74/4.30	**2.60/4.10**	2.61/4.13
8	10%	30%	30%	30%	4.69/6.86	2.97/4.57	3.22/5.02	2.77/4.35	2.68/4.22	**2.66/4.20**
9	30%	10%	10%	10%	3.83/5.77	4.37/6.31	4.68/6.78	3.30/5.04	3.15/4.82	**2.91/4.56**
10	30%	10%	10%	30%	3.78/5.72	3.98/5.83	4.45/6.52	3.29/5.05	3.08/4.74	**2.91/4.56**
11	30%	10%	30%	10%	4.76/6.93	3.92/5.75	4.38/6.43	3.23/4.94	3.03/4.66	**2.84/4.45**
12	30%	10%	30%	30%	4.62/6.77	3.69/5.47	4.20/6.22	3.25/4.98	3.01/4.66	**2.89/4.53**
13	30%	30%	10%	10%	3.78/5.73	3.91/5.73	4.39/6.46	3.19/4.90	3.01/4.63	**2.87/4.48**
14	30%	30%	10%	30%	3.77/5.72	3.67/5.44	4.21/6.26	3.20/4.93	3.00/4.64	**2.88/4.52**
15	30%	30%	30%	10%	4.62/6.77	3.65/5.41	4.17/6.19	3.12/4.81	2.97/4.60	**2.82/4.42**
16	30%	30%	30%	30%	4.54/6.68	3.49/5.23	4.02/6.03	3.16/4.87	2.98/4.63	**2.87/4.50**
London										
1	10%	10%	10%	10%	65.41/12.89	56.40/10.89	52.55/10.57	56.18/9.42	36.77/**7.79**	**36.32**/7.81
2	10%	10%	10%	30%	64.33/12.59	52.48/10.02	56.02/10.89	78.23/9.71	38.52/**8.01**	**38.35**/8.12
3	10%	10%	30%	10%	74.98/14.61	50.69/9.74	50.06/10.40	49.96/9.04	42.57/8.02	**36.61/7.71**
4	10%	10%	30%	30%	72.58/14.11	50.27/9.50	54.58/10.62	112.93/9.56	43.34/8.26	**38.26/8.05**
5	10%	30%	10%	10%	64.25/12.51	49.25/9.72	49.39/10.14	49.82/8.92	36.27/7.66	**35.83/7.65**
6	10%	30%	10%	30%	63.89/12.40	49.07/9.50	51.94/10.49	62.91/9.28	38.25/**7.95**	**38.11**/8.04
7	10%	30%	30%	10%	72.52/14.06	48.01/9.35	47.65/9.99	46.92/8.77	42.32/8.08	**36.79/7.73**
8	10%	30%	30%	30%	71.26/13.80	48.94/9.34	50.61/10.35	78.02/9.42	43.55/8.36	**38.49/8.04**
9	30%	10%	10%	10%	64.74/12.71	76.50/14.48	63.20/13.01	68.60/11.02	43.60/9.10	**40.01/8.37**
10	30%	10%	10%	30%	64.22/12.50	70.26/13.25	68.67/13.11	79.93/10.94	43.73/9.02	**41.32/8.61**
11	30%	10%	30%	10%	72.73/14.15	68.72/13.00	57.55/11.98	54.68/10.32	47.35/9.10	**40.36/8.33**
12	30%	10%	30%	30%	71.44/13.84	65.95/12.40	62.56/12.41	80.49/10.47	47.70/9.18	**41.55/8.51**
13	30%	30%	10%	10%	64.11/12.47	68.56/13.12	58.97/12.01	53.20/10.21	41.95/8.74	**39.75/8.27**
14	30%	30%	10%	30%	63.96/12.38	65.82/12.51	63.27/12.22	93.50/10.71	42.92/8.84	**41.52/8.54**
15	30%	30%	30%	10%	71.36/13.82	64.83/12.36	55.09/11.40	52.45/10.02	46.78/9.02	**41.00/8.41**
16	30%	30%	30%	30%	70.63/13.65	63.71/12.05	58.99/11.76	69.94/10.18	47.70/9.19	**42.64/8.72**
Madrid										
1	10%	10%	10%	10%	194.42/21.14	160.27/17.44	88.13/13.38	85.06/11.80	67.35/9.59	**53.36/9.08**
2	10%	10%	10%	30%	182.28/19.96	148.02/15.31	80.88/12.57	77.00/11.20	65.53/9.55	**54.19/9.28**
3	10%	10%	30%	10%	242.42/25.66	152.47/15.31	79.45/12.44	75.73/10.98	85.42/9.82	**56.29/9.44**
4	10%	10%	30%	30%	225.80/24.09	149.26/14.56	76.78/12.09	71.89/10.79	81.87/9.84	**59.19/9.46**
5	10%	30%	10%	10%	181.95/19.94	130.58/14.88	80.10/12.53	73.95/10.91	64.31/9.42	**53.16/9.15**
6	10%	30%	10%	30%	176.67/19.43	134.51/14.28	76.81/12.18	71.86/10.72	63.94/9.47	**54.99/9.36**
7	10%	30%	30%	10%	225.67/24.08	138.64/14.33	77.45/12.32	69.85/10.66	80.97/9.78	**56.22/9.42**
8	10%	30%	30%	30%	216.53/23.21	143.75/14.23	75.19/12.03	70.04/10.66	79.84/9.87	**61.55/9.74**
9	30%	10%	10%	10%	182.76/19.97	286.84/29.58	160.92/20.72	118.78/14.38	70.72/10.07	**60.12/9.92**
10	30%	10%	10%	30%	177.05/19.45	256.49/26.14	143.67/19.11	108.99/13.62	69.28/9.96	**59.46/9.70**
11	30%	10%	30%	10%	225.74/24.06	257.01/26.00	139.31/18.18	106.29/13.45	86.01/10.29	**57.81/9.58**
12	30%	10%	30%	30%	216.60/23.22	242.29/24.24	129.68/17.34	100.68/13.03	85.05/10.35	**62.50/9.77**
13	30%	30%	10%	10%	177.04/19.44	245.48/25.83	146.32/20.24	106.76/13.49	67.67/9.84	**57.39/9.48**
14	30%	30%	10%	30%	174.01/19.17	233.82/24.11	134.10/18.46	101.29/13.10	68.60/9.90	**58.33/9.63**
15	30%	30%	30%	10%	216.57/23.21	234.81/24.04	132.31/18.51	99.94/13.06	83.39/10.24	**59.25/9.57**
16	30%	30%	30%	30%	210.97/22.69	293.14/29.11	124.91/17.37	97.91/12.80	85.34/10.45	**64.66/9.98**
Melbourne										
1	10%	10%	10%	10%	73.74/25.20	40.22/15.57	28.00/16.00	30.28/14.70	**18.54**/9.87	18.74/**9.52**
2	10%	10%	10%	30%	68.21/23.54	33.51/13.93	30.29/16.01	39.60/14.32	**18.91**/10.33	19.65/**9.99**
3	10%	10%	30%	10%	95.81/31.62	33.31/13.81	25.98/15.32	27.38/14.11	19.50/10.20	**19.05/9.59**
4	10%	10%	30%	30%	88.25/29.40	30.82/13.39	26.37/15.24	31.09/14.08	20.03/10.70	**19.48/9.95**
5	10%	30%	10%	10%	68.32/23.53	32.85/13.66	27.94/15.85	26.41/13.83	**18.06**/9.94	19.03/**9.58**
6	10%	30%	10%	30%	65.85/22.80	30.44/13.26	27.82/15.64	29.96/13.72	**18.81**/10.47	19.07/**9.79**
7	10%	30%	30%	10%	88.30/29.39	30.36/13.20	25.45/15.15	25.33/13.55	19.48/10.45	**19.02/9.50**
8	10%	30%	30%	30%	84.13/28.17	29.38/13.19	25.81/15.23	29.04/13.69	20.16/10.93	**19.45/9.88**
9	30%	10%	10%	10%	67.95/23.32	68.56/23.46	59.37/24.37	42.41/17.67	23.67/11.74	**21.99/10.28**
10	30%	10%	10%	30%	65.58/22.64	59.35/21.15	56.26/22.86	45.19/17.36	22.87/11.75	**22.16/10.33**
11	30%	10%	30%	10%	87.75/29.20	59.31/20.97	47.88/21.62	38.43/16.77	**23.45**/11.71	24.39/**10.99**
12	30%	10%	30%	30%	83.80/28.05	54.43/19.89	48.43/21.29	38.79/16.47	23.32/11.92	**22.20/10.50**
13	30%	30%	10%	10%	65.65/22.64	58.99/20.88	47.04/21.30	37.95/16.64	22.28/11.43	**21.59/10.24**
14	30%	30%	10%	30%	64.38/22.28	54.12/19.80	47.11/20.88	38.99/16.42	22.22/11.67	**20.56/9.96**
15	30%	30%	30%	10%	83.78/28.03	54.18/19.70	42.75/19.91	35.41/16.04	22.87/11.72	**21.03/9.99**
16	30%	30%	30%	30%	81.37/27.32	51.34/19.21	41.73/19.59	37.57/16.13	23.11/12.02	**22.47/10.47**

*The performance metrics in the table are displayed as MAE/SMAPE. The best results are highlighted in boldface.

Table 4
Imputation Performance of All Models in Extreme Missing Scenarios

Scenario	BM	RM	DM	TM	HA	LRTC-TSpN	TRMF	BGCP	LATC	RTTC (Ours)
Guangzhou										
H1	30%	50%	30%	30%	4.49/6.62	4.19/6.10	3.87/5.84	3.10/4.79	2.98/4.64	**2.87/4.51**
H2	30%	70%	30%	30%	4.49/6.63	4.36/6.32	3.75/5.80	3.09/4.79	3.01/4.70	**2.98/4.67**
H3	50%	50%	50%	50%	5.12/7.40	5.32/7.54	4.46/6.68	3.54/5.42	**3.42/5.22**	3.48/5.34
H4	50%	70%	50%	50%	5.18/7.51	5.46/7.70	4.18/6.36	3.57/5.46	**3.52/5.36**	3.69/5.59
London										
H1	30%	50%	30%	30%	70.16/13.51	62.41/11.87	54.75/11.21	67.61/10.11	48.56/9.36	**44.07/8.89**
H2	30%	70%	30%	30%	70.29/13.48	62.66/11.93	55.90/11.52	56.05/9.87	50.72/9.74	**45.19/9.22**
H3	50%	50%	50%	50%	76.13/14.61	78.87/14.71	66.61/13.28	60.19/10.70	63.11/11.43	**52.61/10.86**
H4	50%	70%	50%	50%	76.98/14.71	80.03/14.93	66.38/13.45	**56.81/10.70**	73.37/13.21	58.71/11.94
Madrid										
H1	30%	50%	30%	30%	206.42/22.27	226.51/22.68	125.77/16.59	94.64/12.54	90.36/10.86	**63.40/9.79**
H2	30%	70%	30%	30%	203.20/21.99	239.62/23.64	115.29/16.15	93.13/12.47	112.09/12.37	**67.24/10.33**
H3	50%	50%	50%	50%	233.94/24.84	306.54/30.34	168.59/20.98	108.97/13.88	177.85/16.40	**95.08/13.19**
H4	50%	70%	50%	50%	233.56/24.82	326.70/33.43	137.91/18.59	109.65/**13.92**	230.30/20.71	**104.35**/14.65
Melbourne										
H1	30%	50%	30%	30%	79.40/26.74	48.93/18.78	41.97/19.78	35.50/15.88	23.26/12.33	**23.32/11.57**
H2	30%	70%	30%	30%	77.98/26.31	47.07/18.57	34.54/18.20	33.83/15.90	24.13/13.16	**23.21/11.38**
H3	50%	50%	50%	50%	91.86/30.37	59.89/23.94	57.77/24.34	40.50/17.20	33.60/16.51	**30.65/13.72**
H4	50%	70%	50%	50%	91.72/30.30	59.67/24.18	58.99/25.08	38.62/17.56	39.67/17.73	**37.63/17.23**

*The performance metrics in the table are displayed as MAE/SMAPE. The best results are highlighted in boldface.

with high missing rates, indicating its robustness over various missing patterns. For instance, the error gap between the best and the worst scenario in London for the proposed model is approximately 6.8 veh/15min. However, for LATC, the best base model in our experiment, the gap is around 11.4 veh/15min. The gap can be even larger for other models except historical average, which is robust but with higher imputation error.

Compared with London and Madrid, the imputation error in Guangzhou and Melbourne is generally smaller, and the proposed model does not show discernible merit in scenarios with low missing rates, which could be attributed to less challenging traffic dynamics there. Clues can be found by recalling the distribution of singular values of all cities demonstrated in Fig. 5, where the primary singular values of Guangzhou and Melbourne show stronger dominance compared with the other cities, implying higher volatility and more pattern changes in temporal dynamics (e.g., see Fig. 2. Still, in these two cities, lower imputation error of the proposed model over base models can be noticed in scenarios with a high BM missing rate.

It can also be concluded from the results that explicit or implicit temporal modeling is crucial to robust imputation under complicated missing scenarios. For models without temporal modeling, such as LRTC-TSpN, the performance is not satisfactory in most evaluation scenarios, potentially due to the permutation and scale invariance of matrix rank, which is an inherent limitation of the rank minimization paradigm. In other words, by minimizing the rank of a tensor, the solution remains unchanged regardless of the order and scale of fibers (analogous to columns/rows of matrix in higher order). Consequently, when many fibers or complete slices are missing, imputation models will find it difficult to properly recover missing values unless equipped with a time series prior.

Among all missing patterns, RM, DM and TM are missing patterns that are easier to handle. An increase in the missing rate regarding these missing patterns, as per the tables, incurs limited perturbations in terms of the errors in most cases. In comparison, most models are more sensitive to changes in the missing rates of BM. Although models like TRMF, BGCP, and LATC are all embedded with time series models, they still suffer from discernible performance degradation or instability during imputation.

In addition, the imputation performance of these models in extreme missing scenarios is also examined, as listed in Table 4. Four composite scenarios are constructed, where the composite missing rates of H1, H2, H3, and H4 are around 83%, 90%, 94%, and 96%, respectively. Across all models, there is an apparent increase in imputation errors in extreme missing scenarios compared with those in basic scenarios, particularly when composite missing rate goes beyond 90%. Nevertheless, RTTC can still consistently outperform base models in most of these challenging scenarios. It is also observed that imputation models with temporal modeling demonstrate greater robustness in these extreme scenarios, as evidenced by the lower imputation errors.

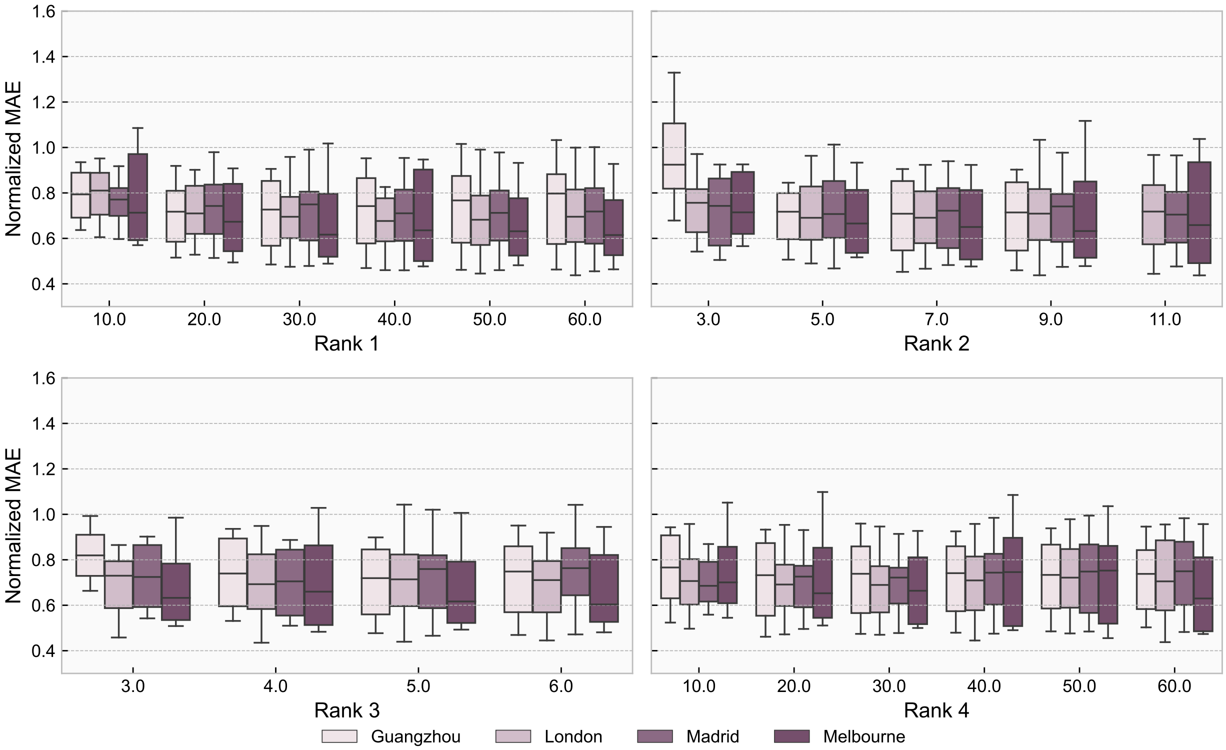


Figure 7: Sensitivity analysis on tensor ranks. For demonstration purpose, the imputation error is normalized to the same level across cities. The axis label "Rank i" indicates the rank of factor matrix $\boldsymbol{U}_i$. No box is displayed for Guangzhou data when the mode-2 rank equals 11, as it exceeds the number of weeks in the data.

6.2. Sensitivity Analysis

While the proposed RTTC outperforms base models in terms of accuracy and robustness, one may question its practical feasibility due to the requirement of tuning more hyperparameters. A sensitivity analysis was first conducted specifically focusing on tensor ranks for Tucker factorization, as demonstrated in Fig. 7. In general, most rank configurations do not exert a significant impact on the imputation error, showing an observable degree of consistency. Minor deviations do exist; however, the overall performance remains comparably stable against base models. There are also very few exceptions, where increases in imputation error can be noted, e.g., small rank values in Guangzhou data. This is possibly because of under-fitting with small rank values, which may limit the model from capturing all necessary details for accurately recovering the traffic dynamics. Therefore, in practice, a trade-off between quality and efficiency is needed for practical application. Nonetheless, thanks to the additional rank minimization, the risk of over-fitting for selecting a large rank value is minimized, which greatly relieves the burden of rank tuning compared with other tensor factorization-based models.

Further sensitivity analysis was performed on other model parameters, encompassing coefficients μ, λ, ξ and the norm parameter γ (see Fig. 8). In general, these hyperparameters do not exert a substantial effect on the imputation error, with the exception of the coefficient μ. However, the impact of μ on imputation error is as minor as other hyperparameters when it is less than 1. This analysis underscores the robustness of the proposed RTTC in relation to hyperparameter sensitivity, suggesting that avoiding overly small rank values and overly large μ will suffice to achieve optimal performance, while the influences of other parameters remain mostly invariant.

6.3. Decomposition & Factorization

The embedded time series decomposition model allows us to distinctly segregate the temporal trend, spatio-temporal correlations, and outliers in the data using three components, namely trend, seasonality, and error. This not only enhances the accuracy of the imputation result but also provides us with a granular view of the temporal dynamics. Fig. 9 depicts the decomposed imputation result on an example time series of traffic volume. For clarity

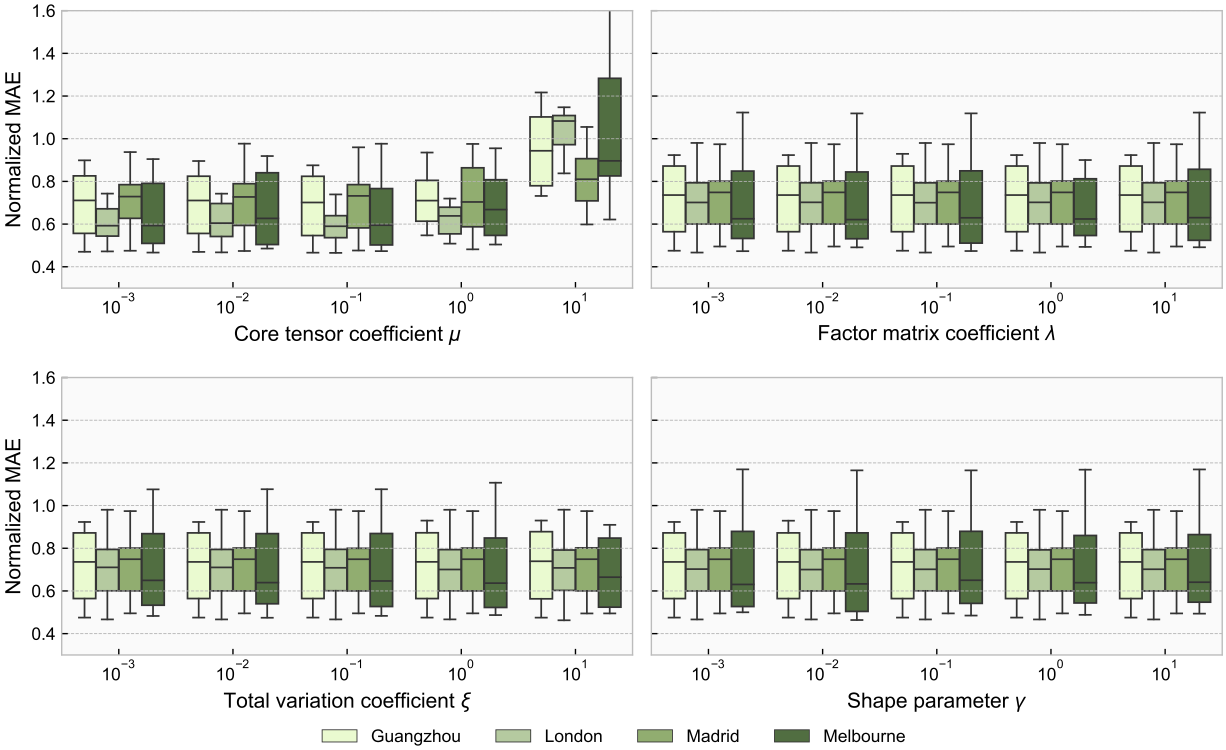


Figure 8: Sensitivity analysis on other model parameters. For demonstration purpose, the imputation error is normalized to the same level across cities.

of demonstration, the sum of the trend and seasonality components is shown as a blue curve in subfigure (a), which successfully reconstructs temporal dynamics with less than 30% of observations. The sole failure of imputation appears around Aug 2, 2019, when there is black-out missing nearby. Without additional information like events, it is reasonable to assume the volume retains the same pattern as those of the previous and subsequent days.

Each of the three decomposed components plays a pivotal role in the imputation process. The trend component assigns a unique constant to every segment of the time series on the basis of changepoint detection, aiming to accommodate supply-side changes near a specific sensor that cannot be modeled by tensor factorization alone. In the given example, two changepoints, along with three segments, are identified. The decrement in traffic volume observed on August 14 aligns coherently with the trend reduction of the second segment. Conversely, the obvious drop in the third segment is not reflected in the changes in the corresponding trend value, which could be attributed to the seasonality component. By examining the volume profiles of other loop detectors, analogous volume drops were recorded. Such pattern is hence discerned by the seasonality component, which is responsible for handling cross-sensor correlations. Working in collaboration with the trend, the seasonality component is capable of modeling the periodicity in the temporal domain as well as the similarity amongst sensors. As demonstrated in subfigure (c), it can effectively distinguish between the weekday and weekend pattern, thanks to the four-way representation of tensor. Additionally, the error component adeptly captures outliers and minor fluctuations in data, thereby mitigating the adverse effects of anomalous observations on tensor factorization and rank minimization.

Apart from the time series decomposition, the tensor factorization for the seasonality component can innately decompose the data into multiple basic elements, allowing for understanding the underlying patterns and structures within the data (Sun and Axhausen, 2016). In subsequent visualization, factor matrices are normalized, with the core tensor being scaled accordingly, for demonstration purpose. The core tensor $\mathcal{G}$ is showcased in Fig. 10, the shape of which, $30 \times 9 \times 5 \times 20$, corresponds with the prescribed Tucker rank. To visualize the four-way tensor, the tensor slices of the sensor (S) and time-of-the-day (T) modes are tiled along the week (W) and day-of-the-week (D) modes, where the suffix number indicates the indices of the latent patterns. The core tensor encodes the interactions across

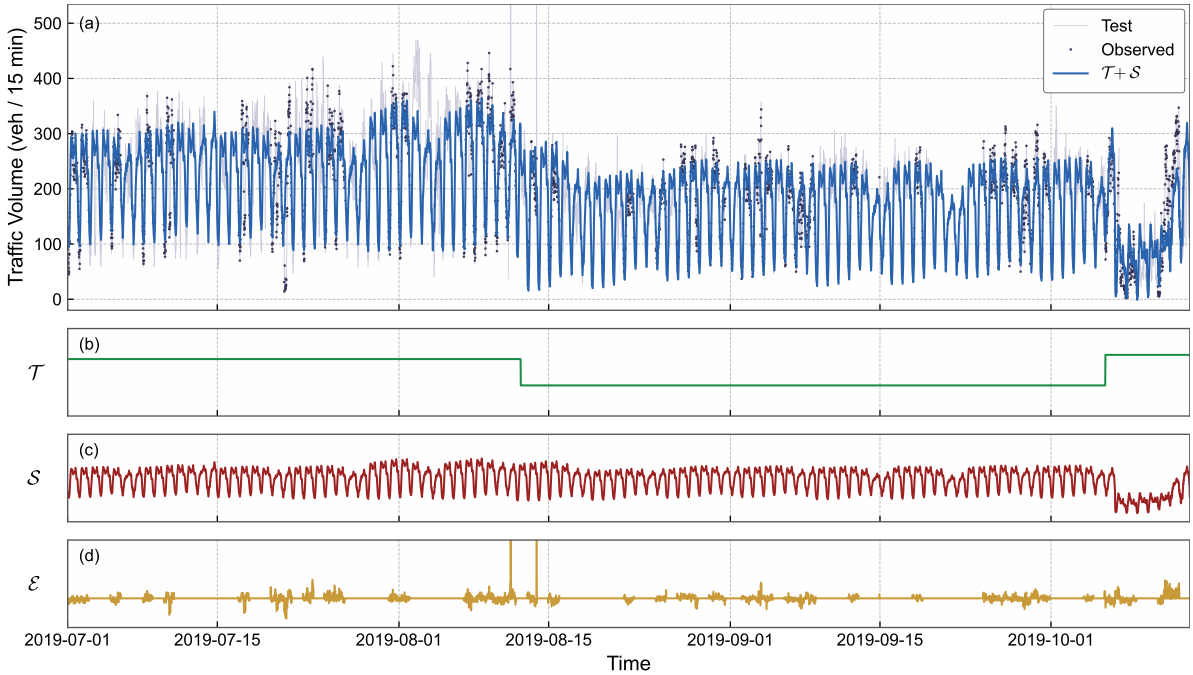


Figure 9: Time series decomposition of the imputation result on the time series of traffic volume data collected by a loop detector in London. (a) The sum of the trend and seasonality components are compared against the ground truth. (b)–(d) The three subfigures show the trend, seasonality, and error components, respectively.

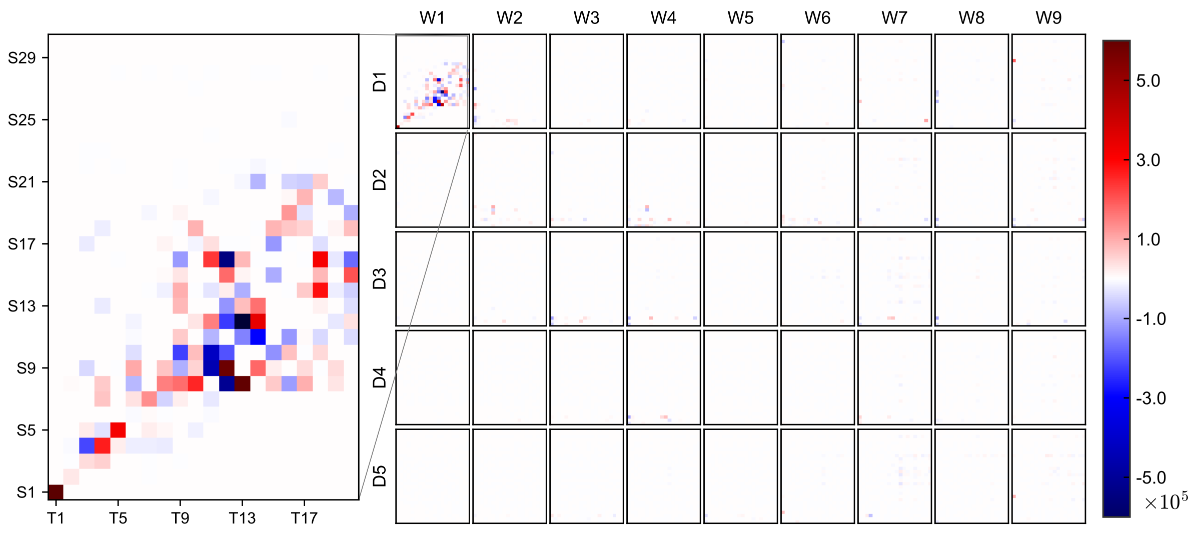


Figure 10: Visualization of the core tensor. The zoomed subfigure is an enlarged view of the tensor slice of W1 and D1. The massive blanks are values close to zero, indicating the sparsity of the resulting core tensor.

modes; a larger value indicates a stronger interaction among the corresponding dimensions of modes. In terms of the week mode and the day-of-the-week mode, the core tensor is dominated by the leading entry and concentrates near the diagonal, which is displayed in the zoomed view. Such parsimony is induced by the ℓ^1-norm in the optimization objective, thereby limiting the interactions among modes and helping achieving a low rank result.

One can further dig into the patterns in the factor matrices. Most interactions in the example above revolve around W1 and D1, and we therefore place the focus on the sensor-mode and time-of-the-day mode factor matrices, i.e., $\boldsymbol{U}_1$ and $\boldsymbol{U}_4$. The row factor in $\boldsymbol{U}_1$ corresponding to the example sensor is first extracted, as shown in Fig. 11 (a). Among

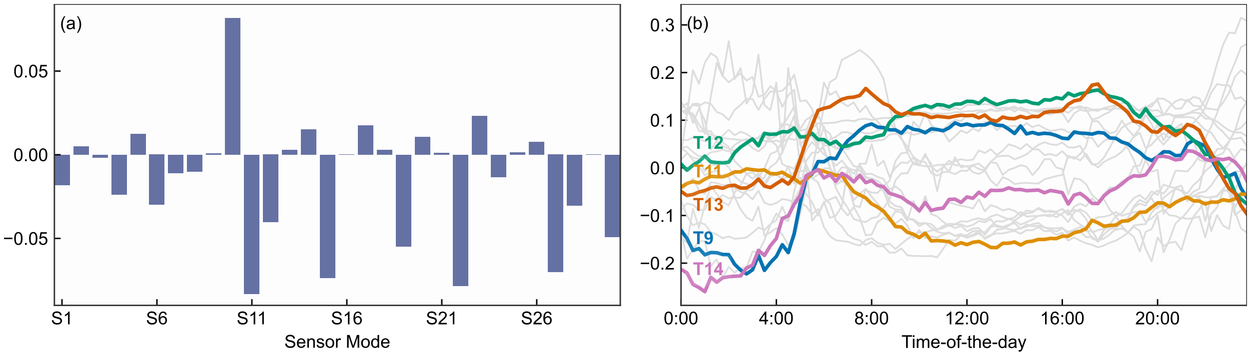


Figure 11: Visualization of factor matrices. (a) Sensor-mode row factor of the example sensor in the factor matrix $\boldsymbol{U}_1$. (b) Time-of-the-day patterns in the factor matrix $\boldsymbol{U}_4$. The colored curves are the patterns with the most contribution.

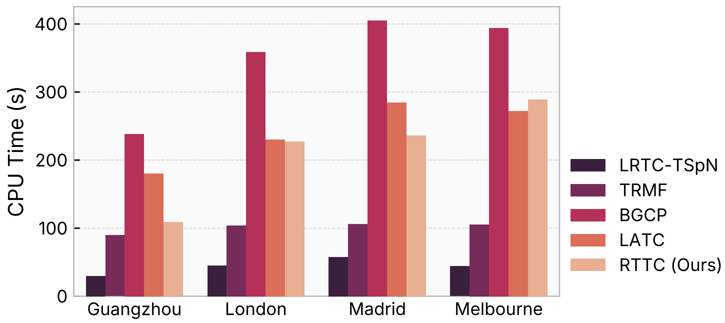


Figure 12: Average computation time across various missing scenarios.

all latent patterns, it can be observed that S10, S11, S15, S22, and S27 show more influence on the resulting tensor than others. Based on Fig. 10, the time-of-the-day patterns with the most interactions with those sensor patterns can be roughly identified as T9, T11, T12, T13, and T14, which are demonstrated in Fig. 11 (b). T9 shows an unimodal distribution with higher traffic during the daytime and the evening and lower traffic in the nighttime. Both T13 and T14 exhibit a bimodal distribution, with a morning peak at around 6:00–8:00 *a.m.* and an evening peak at around 17:00–21:00 *p.m.* In comparison, T11 and T12 are flatter patterns going higher and lower, respectively, in the daytime.

It should be noted that the proposed framework is primarily designed for missing value imputation rather than pattern discovery. The Tucker ranks selected for accuracy consideration may conflict with the purpose of interpretation, since they could be too high to be easily interpreted and no non-negative constraints are involved. Therefore, careful consideration should be given to the trade-off between imputation accuracy and interpretability, depending on on the focus of the practical application.

6.4. Computation Time

The average computation time across all missing scenarios of different models is demonstrated in Fig. 12. The experiments were performed on a workstation equipped with an Intel Core i7–13700K CPU and 128GB of RAM. Notably, LRTC-TSpN and TRMF are faster than others due to lower computational complexity. However, it should be noted that they also presented a higher imputation error and lower stability in complicated missing scenarios. In comparison, BGCP and LATC exhibit lower computational efficiency, which are constraint by the sampling process and autoregression model fitting, respectively. In general, the proposed model finished all imputation tasks within five minutes, which is similar to the computation time of LATC. The primary performance bottleneck of our model lies in solving the Sylvester equation for factor matrices $\boldsymbol{U}_m (m > 1)$. Details are discussed in Appendix C. Most other subproblems in our model have a relatively low time complexity. The $\boldsymbol{U}_1$-subproblem involves matrix operations, resulting in a complexity of $\mathcal{O}(N R)$, where $N = \prod_i N_i$ is the total number of entries of the input tensor, and $R = \prod_i r_i$ is the total number of entries of the core tensor. The complexity of $\boldsymbol{V}$-subproblems is overshadowed by SVD of $\boldsymbol{W}_m$ with a $\mathcal{O}(N_m r_m^2)$ complexity. Similar to $\boldsymbol{U}_1$, the $\mathcal{G}$-subproblem also has a $\mathcal{O}(NR)$ complexity. The remaining subproblems of $\boldsymbol{E}$, $\boldsymbol{c}$, and $\boldsymbol{X}$ involve only element-wise tensor operations, sharing a $\mathcal{O}(N)$ complexity.

7. Conclusion

In this paper, we addressed the pervasive problem of missing values in transport data, which undermines the integrity and efficacy of data-driven transportation analysis. As one of the most promising solutions to this problem, current tensor-based methods are still limited in terms of robustness facing complicated composite missing patterns. To amend this gap, we proposed a novel tensor-based imputation framework (RTTC) that integrates a time series decomposition model to simultaneously account for long-term trends, spatio-temporal correlations, and outliers in the data. The combination of tensor factorization and rank minimization also eliminates the need for exhaustive rank tuning in conventional tensor factorization-based methods. In addition, a Bregman ADMM algorithm is developed to solve the resulting multi-block separable nonconvex optimization problem efficiently.

Experiments on four real-world transport datasets demonstrate that the proposed framework can outperform state-of-the-art imputation methods, especially in the presence of complex missing patterns with high missing rates. The results highlight the importance of integrating temporal modeling in tensor completion framework. Sensitivity analysis also underscores the stability of our framework with respect to hyperparameter settings. It is also demonstrated that the our framework has the potential of providing interpretable results from both the perspectives of time series decomposition and tensor factorization.

Future work in this direction may focus on incorporating supplementary information to further assist imputation, such as road topology and weather conditions. Holidays and events can also be explicitly involved to inject additional knowledge for imputation. From a methodological perspective, the total variation regularizer for smoothing imputation results is not the ideal choice in the time series context. It is worthwhile to investigate better regularization for temporal modeling (Yang et al., 2021; Chen et al., 2022). Furthermore, non-negative constraints may also be helpful in enhancing the interpretability of the imputation framework.

Acknowledgements

This work was supported by the PANAMERA project (Grant Number: 19I21016F) funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK). This research was also partially supported by the European Interest Group CONCERT-Japan DARUMA project (Grant Number: 01DR21010) funded by the German Federal Ministry of Education and Research (BMBF).

Appendix A Hyperparameter Settings

We show the hyperparameters of base models as follows. For LRTC-TSpN, we set norm parameter $p = 0.9$, and decay rate $\beta = 2$. The truncation rate is set as $\theta = 0.05$ for Guangzhou dataset, and $\theta = 0.001$ for others. For BGCP, we set tensor rank $r = 30$. For TRMF, we set tensor rank $r = 10$, and coefficients $\lambda_x = \lambda_w = \lambda_\theta = 1$. For LATC, we set truncation parameter $r = 10$, and coefficient $\lambda = 10^{-5}$. Finally, for our model, we used tensor ranks $\boldsymbol{r} = \{30, 9, 5, 20\}$, coefficients $\mu = 0.1, \lambda = 0.1, \xi = 1.0$, and norm parameter $\gamma = 0.01$. Regarding model training, we trained all the models for at most 250 iterations with the convergence criterion 10^{-4}. The updating step size ρ of all the models was selected optimally from the range 10^{-2} to 10^{-5}. Additionally, in our model, the coefficient of Bregman divergence η was always set to equal ρ. The optimal set of hyperparameters can be determined through Bayesian optimization with cross validation, as detailed in Bergstra et al. (2011), which is more efficient compared with traditional techniques like grid search. It's noteworthy that although our method seems to have many hyperparameters, Section 6.2 demonstrates that their influence on imputation performance is minor.

Appendix B Changepoint Detection

The changepoint detection in our implementation is accomplished using the pruned exact linear time (PELT) algorithm, which features high efficiency, scalability, and adaptability. The basic idea of PELT is to identify points in the time series where the statistical properties change significantly. Formally, we consider a time series represented by $\boldsymbol{x} = (x_1, x_2, \dots, x_N)$, which is normalized for zero mean and unit variance. Then consider a sequence $\boldsymbol{\tau} =$

$(\tau_0, \tau_1, \dots, \tau_{N_\tau+1})$, where τ_0 and $\tau_{N_\tau+1}$ mark the time series' endpoints, while others constitute the index set of changepoints. The changepoint detection optimizes towards the following objective function,

$$\min_{\boldsymbol{\tau}} \sum_{i=0}^{N_\tau} \mathrm{Cost}(\boldsymbol{x}_{\tau_i+1:\tau_{i+1}}) + \alpha \mathrm{Pen}(\boldsymbol{\tau}), \tag{51}$$

where $\alpha > 0$ is a balancing coefficient. The cost function $\mathrm{Cost}(\boldsymbol{x}_{\tau_i+1:\tau_{i+1}})$ measures the homogeneity of each segment, and the penalty function $\mathrm{Pen}(\boldsymbol{\tau})$ is a complexity regularizer. In this study, a kernelized cost function with the Gaussian kernel is employed (Arlot et al., 2019). The original times series is projected to the reproducing kernel Hilbert space by the Gaussian kernel function $\kappa(\cdot,\cdot)$ and a corresponding implicit transformation $\phi(x_i) = \kappa(x_i, \cdot) \in \mathbb{H}$. The cost is defined as the sum of squared distance between the segment and the its mean in the projected space,

$$\begin{aligned} \mathrm{Cost}(\boldsymbol{x}_{\tau_i+1:\tau_{i+1}}) &= \sum_{j=\tau_i+1}^{\tau_{i+1}} \|\phi(x_j) - \bar{\phi}(\boldsymbol{x}_{\tau_i+1:\tau_{i+1}})\|_{\mathbb{H}}^2 \\ &= \sum_{j=\tau_i+1}^{\tau_{i+1}} \kappa(x_j, x_j) - \frac{1}{\tau_{i+1}-\tau_i} \sum_{j=\tau_i+1}^{\tau_{i+1}} \sum_{k=\tau_i+1}^{\tau_{i+1}} \kappa(x_j, x_k). \end{aligned} \tag{52}$$

Substituting the Gaussian kernel $\kappa(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2\right)$ into the Eq. (52), the cost function can then be rewritten as,

$$\mathrm{Cost}(\boldsymbol{x}_{\tau_i+1:\tau_{i+1}}) = \tau_{i+1} - \tau_i - \frac{1}{\tau_{i+1}-\tau_i} \sum_{j=\tau_i+1}^{\tau_{i+1}} \sum_{k=\tau_i+1}^{\tau_{i+1}} \exp\left(-\|x_j - x_k\|^2\right). \tag{53}$$

For the penalty function, we directly relate it to the count of changepoints N_τ, which supports the linear time complexity of PELT. Since the number of changepoints are unknown beforehand, PELT works in a sequential way to evaluate the cost relating to each candidate changepoint. Detailed steps can be found in Algorithm 2.

Algorithm 2 PELT Changepoint Detection

Input: Time series $\boldsymbol{x}$, balancing coefficient α.
Output: Optimal changepoint set $\mathcal{R}_N$.

```
Initialize an array of changepoint sets R ← ∅; initialize an array of objective values P ← {−α}; initialize an array
of candidates Θ ← {0}.
for t ← 1, N do
    Find the best changepoint till t: τ* ← arg min_{τ∈Θ} (P_τ + Cost(x_{τ+1:t}) + α).
    Update objective value: P_t ← P_{τ*} + Cost(x_{τ*+1:t}) + α.
    Update changepoints: R_t ← R_{τ*} ∪ {τ*}.
    Prune the candidate set: Θ ← Θ ∩ {τ | P_τ + Cost(x_{τ+1:t}) ≤ P_{τ*}} ∪ {τ*}.
end for
```

Appendix C Solution to Sylvester Equation

The Sylvester equation is the major computational bottleneck of the solution algorithm. In our experiments, the Bartels-Stewart algorithm was adopted (see Algorithm 3), with a time complexity of $\mathcal{O}(N_m^3)$. Despite the cubic complexity of this subproblem, the overall computation time is still acceptable — less than five minutes — for the test data with over 3 million entries. However, the soution efficiency can be further improved via Krylov subspace methods. The original formulation of the equation, $\boldsymbol{A}\boldsymbol{U}_m + \boldsymbol{U}_m\boldsymbol{B} = \boldsymbol{R}$, can be first transformed into a linear system $\mathcal{S}(\boldsymbol{U}_m) = \mathrm{vec}(\boldsymbol{R})$ by defining a Sylvester operator $\mathcal{S}(\boldsymbol{U}_m) = \mathrm{vec}(\boldsymbol{A}\boldsymbol{U}_m + \boldsymbol{U}_m\boldsymbol{B})$, where $\mathrm{vec}(\cdot)$ is the vectorization operator. The resulting linear system can then be solved using conjugate gradient algorithm (Hestenes and Stiefel, 1952), as given in Algorithm 4. The time complexity can be reduced to $\mathcal{O}(N_m^2 r_m)$, which is more efficient than standard Bartels-Stewart algorithm.

Algorithm 3 Bartels-Stewart Algorithm for $\boldsymbol{U}$-subproblems

Input: Matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{R}$.
Output: Updated factor matrix $\boldsymbol{U}$.
1: Compute real Schur decomposition of $\boldsymbol{A} = \boldsymbol{Q}_A \boldsymbol{T}_A \boldsymbol{Q}_A^\top$.
2: Compute real Schur decomposition of $\boldsymbol{B} = \boldsymbol{Q}_B \boldsymbol{T}_B \boldsymbol{Q}_B^\top$.
3: Transform $\boldsymbol{R}$ to new coordinates: $\tilde{\boldsymbol{R}} \leftarrow \boldsymbol{Q}_A^\top \boldsymbol{R} \boldsymbol{Q}_B$.
4: Initialize $\boldsymbol{U}$ with zeros having the same shape as $\tilde{\boldsymbol{R}}$.
5: **for** $i \leftarrow N_m, \ldots, 1$ **do**
6: **for** $j \leftarrow r_m, \ldots, 1$ **do**
7: $\boldsymbol{U}_{(i,j)} \leftarrow (\tilde{\boldsymbol{R}}_{(i,j)} - \boldsymbol{T}_{A(i,:i)} \boldsymbol{U}_{(:i,j)} - \boldsymbol{U}_{(i,j+1:)} \boldsymbol{T}_{B(j+1:,j)}) / (\boldsymbol{T}_{A(i,i)} + \boldsymbol{T}_{B(j,j)})$.
8: **end for**
9: **end for**
10: Transform $\boldsymbol{U}$ back to original coordinates: $\boldsymbol{U} \leftarrow \boldsymbol{Q}_A \boldsymbol{U} \boldsymbol{Q}_B^\top$.

Algorithm 4 Conjugate Gradient Algorithm for $\boldsymbol{U}$-subproblems

Input: Factor matrix $\boldsymbol{U}$; Sylvester operator $\mathcal{S}$; matrix $\boldsymbol{R}$.
Output: Updated factor matrix $\boldsymbol{U}$.
1: Initialize $\boldsymbol{u}$ as $\operatorname{vec}(\boldsymbol{U})$.
2: Initialize residual $\boldsymbol{r}_1 \leftarrow \operatorname{vec}(\boldsymbol{R}) - \mathcal{S}(\boldsymbol{U})$; initialize search direction $\boldsymbol{s}_1 \leftarrow \boldsymbol{r}_1$.
3: **for** $i \leftarrow 1, \ldots, K$ **do**
4: Reshape $\boldsymbol{s}_i$ as $\boldsymbol{S}_i$.
5: Compute step size: $\alpha_i \leftarrow \|\boldsymbol{r}_i\|^2 / (\boldsymbol{s}_i^\top \mathcal{S}(\boldsymbol{S}_i))$.
6: Update vectorized factor: $\boldsymbol{u}_{i+1} \leftarrow \boldsymbol{u}_i + \alpha_i \boldsymbol{s}_i$.
7: Update residual: $\boldsymbol{r}_{i+1} \leftarrow \boldsymbol{r}_i - \alpha_i \mathcal{S}(\boldsymbol{S}_i)$.
8: Update search direction: $\boldsymbol{s}_{i+1} \leftarrow \boldsymbol{r}_{i+1} + \|\boldsymbol{r}_{i+1}\|^2 / \|\boldsymbol{r}_i\|^2 \boldsymbol{s}_i$.
9: **end for**
10: Reshape $\boldsymbol{u}_{i+1}$ as $\boldsymbol{U}$.

References

AASHTO, 2009. AASHTO Guidelines for Traffic Data Programs. American Association of State Highway and Transportation Officials, Washington, D.C.

Arlot, S., Celisse, A., Harchaoui, Z., 2019. A kernel multiple change-point algorithm via model selection. Journal of Machine Learning Research 20, 1–56.

Bauschke, H.H., Bolte, J., Teboulle, M., 2017. A descent lemma beyond Lipschitz gradient continuity: First-order methods revisited and applications. Mathematics of Operations Research 42, 330–348. doi:10.1287/moor.2016.0817.

Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for hyper-parameter optimization, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., Granada, Spain. pp. 2546–2554.

Boyd, S., 2010. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning 3, 1–122. doi:10.1561/2200000016.

Cai, J.F., Candès, E.J., Shen, Z., 2010. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization 20, 1956–1982. doi:10.1137/080738970.

Chen, C., He, B., Ye, Y., Yuan, X., 2016. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. Mathematical Programming 155, 57–79. doi:10.1007/s10107-014-0826-5.

Chen, K., Dong, H., Chan, K.S., 2013. Reduced rank regression via adaptive nuclear norm penalization. Biometrika 100, 901–920. doi:10.1093/biomet/ast036.

Chen, X., Cai, Y., Ye, Q., Chen, L., Li, Z., 2018. Graph regularized local self-representation for missing value imputation with applications to on-road traffic sensor data. Neurocomputing 303, 47–59. doi:10.1016/j.neucom.2018.04.029.

Chen, X., He, Z., Sun, L., 2019. A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation. Transportation Research Part C: Emerging Technologies 98, 73–84. doi:10.1016/j.trc.2018.11.003.

Chen, X., Lei, M., Saunier, N., Sun, L., 2022. Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation. IEEE Transactions on Intelligent Transportation Systems 23, 12301–12310. doi:10.1109/TITS.2021.3113608.

Chen, X., Yang, J., Sun, L., 2020. A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation. Transportation Research Part C: Emerging Technologies 117, 102673. doi:10.1016/j.trc.2020.102673.

Chen, Y., Chen, X.M., 2022. A novel reinforced dynamic graph convolutional network model with data imputation for network-wide traffic flow prediction. Transportation Research Part C: Emerging Technologies 143, 103820. doi:10.1016/j.trc.2022.103820.

Duan, Y., Lv, Y., Liu, Y.L., Wang, F.Y., 2016. An efficient realization of deep learning for traffic data imputation. Transportation Research Part C: Emerging Technologies 72, 168–181. doi:10.1016/j.trc.2016.09.015.

El Esawey, M., Mosa, A.I., Nasr, K., 2015. Estimation of daily bicycle traffic volumes using sparse data. Computers, Environment and Urban Systems 54, 195–203. doi:10.1016/j.compenvurbsys.2015.09.002.

Fang, S., Zhang, C., Xiang, S., Pan, C., 2023. AutoMSNet: Multi-source spatio-temporal network via automatic neural architecture search for traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems 24, 2827–2841. doi:10.1109/TITS.2022.3225553.

Goulart, J.d.M., Kibangou, A., Favier, G., 2017. Traffic data imputation via tensor completion based on soft thresholding of tucker core. Transportation Research Part C: Emerging Technologies 85, 348–362. doi:10.1016/j.trc.2017.09.011.

Hale, E.T., Yin, W., Zhang, Y., 2008. Fixed-point continuation for ℓ1-minimization: Methodology and convergence. SIAM Journal on Optimization 19, 1107–1130. doi:10.1137/070698920.

Hestenes, M.R., Stiefel, E., 1952. Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards 49, 409–436.

Higham, N.J., 2002. Accuracy and Stability of Numerical Algorithms. 2 ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. doi:10.1137/1.9780898718027.

Jiang, W., Zheng, N., Kim, I., 2022. Missing data imputation for transfer passenger flow identified from in-station WiFi systems. Transportmetrica B , 1–18doi:10.1080/21680566.2022.2064935.

Kang, Z., Peng, C., Cheng, Q., 2015. Robust PCA via nonconvex rank approximation, in: Proceedings of the 15th IEEE International Conference on Data Mining, IEEE, Atlantic City, USA. pp. 211–220. doi:10.1109/ICDM.2015.15.

Killick, R., Fearnhead, P., Eckley, I.A., 2012. Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association 107, 1590–1598. doi:10.1080/01621459.2012.737745.

Kolda, T.G., Bader, B.W., 2009. Tensor Decompositions and Applications. SIAM Review 51, 455–500. doi:10.1137/07070111X.

Laña, I., Olabarrieta, I.I., Vélez, M., Del Ser, J., 2018. On the imputation of missing data for road traffic forecasting: New insights and novel techniques. Transportation Research Part C: Emerging Technologies 90, 18–33. doi:10.1016/j.trc.2018.02.021.

Li, H., Li, M., Lin, X., He, F., Wang, Y., 2020. A spatiotemporal approach for traffic data imputation with complicated missing patterns. Transportation Research Part C: Emerging Technologies 119, 102730. doi:10.1016/j.trc.2020.102730.

Li, L., Li, Y., Li, Z., 2013. Efficient missing data imputing for traffic flow by considering temporal and spatial dependence. Transportation Research Part C: Emerging Technologies 34, 108–120. doi:10.1016/j.trc.2013.05.008.

Liang, Y., Zhao, Z., Sun, L., 2022. Memory-augmented dynamic graph convolution networks for traffic data imputation with diverse missing patterns. Transportation Research Part C: Emerging Technologies 143, 103826. doi:10.1016/j.trc.2022.103826.

Liu, J., Musialski, P., Wonka, P., Ye, J., 2013. Tensor completion for estimating missing values in visual data. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 208–220. doi:10.1109/TPAMI.2012.39.

Liu, Y., Lyu, C., Zhang, Y., Liu, Z., Yu, W., Qu, X., 2021. DeepTSP: Deep traffic state prediction model based on large-scale empirical data. Communications in Transportation Research 1, 100012. doi:10.1016/j.commtr.2021.100012.

Lu, C., Tang, J., Yan, S., Lin, Z., 2014. Generalized nonconvex nonsmooth low-rank minimization, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Columbus, USA. pp. 4130–4137. doi:10.1109/CVPR.2014.526.

Mahajan, V., Kuehnel, N., Intzevidou, A., Cantelmo, G., Moeckel, R., Antoniou, C., 2022. Data to the people: A review of public and proprietary data for transport models. Transport Reviews 42, 415–440. doi:10.1080/01441647.2021.1977414.

Nie, T., Qin, G., Sun, J., 2022. Truncated tensor schatten p-norm based approach for spatiotemporal traffic data imputation with complicated missing patterns. Transportation Research Part C: Emerging Technologies 141, 103737. doi:10.1016/j.trc.2022.103737.

Salakhutdinov, R., Mnih, A., 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo, in: Proceedings of the 25th International Conference on Machine Learning, ACM Press, Helsinki, Finland. pp. 880–887. doi:10.1145/1390156.1390267.

Smith, B.L., Scherer, W.T., Conklin, J.H., 2003. Exploring imputation techniques for missing data in transportation management systems. Transportation Research Record 1836, 132–142. doi:10.3141/1836-17.

Sun, L., Axhausen, K.W., 2016. Understanding urban mobility patterns with a probabilistic tensor factorization framework. Transportation Research Part B: Methodological 91, 511–524. doi:10.1016/j.trb.2016.06.011.

Tan, H., Feng, G., Feng, J., Wang, W., Zhang, Y.J., Li, F., 2013. A tensor-based method for missing traffic data completion. Transportation Research Part C: Emerging Technologies 28, 15–27. doi:10.1016/j.trc.2012.12.007.

Tang, J., Zhang, G., Wang, Y., Wang, H., Liu, F., 2015. A hybrid approach to integrate fuzzy C-means based imputation method with genetic algorithm for missing traffic volume data estimation. Transportation Research Part C: Emerging Technologies 51, 29–40. doi:10.1016/j.trc.2014.11.003.

Tight, M., Redfern, E., Watson, S., Clark, S., 1993. Outlier Detection and Missing Value Estimation in Time Series Traffic Count Data. Technical Report 401. The University of Leeds. Leeds, UK.

Wang, F., Cao, W., Xu, Z., 2018. Convergence of multi-block Bregman ADMM for nonconvex composite problems. Science China Information Sciences 61, 122101. doi:10.1007/s11432-017-9367-6.

Wang, Y., Yang, J., Yin, W., Zhang, Y., 2008. A New Alternating Minimization Algorithm for Total Variation Image Reconstruction. SIAM Journal on Imaging Sciences 1, 248–272. doi:10.1137/080724265.

Yamamoto, R., Hontani, H., Imakura, A., Yokota, T., 2022. Fast algorithm for low-rank tensor completion in delay-embedded space, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, New Orleans, LA, USA. pp. 2048–2056. doi:10.1109/CVPR52688.2022.00210.

Yang, J.M., Peng, Z.R., Lin, L., 2021. Real-time spatiotemporal prediction and imputation of traffic status based on LSTM and Graph Laplacian regularized matrix factorization. Transportation Research Part C: Emerging Technologies 129, 103228. doi:10.1016/j.trc.2021.

103228.

Yin, W., Osher, S., Goldfarb, D., Darbon, J., 2008. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. SIAM Journal on Imaging Sciences 1, 143–168. doi:10.1137/070703983.

Yu, H.F., Rao, N., Dhillon, I.S., 2016. Temporal Regularized Matrix Factorization for High-dimensional Time Series Prediction, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, ACM Press, Barcelona, Spain. pp. 847–855.

Yu, Q., Zhang, X., Chen, Y., Qi, L., 2022. Low Tucker rank tensor completion using a symmetric block coordinate descent method. Numerical Linear Algebra with Applications 30, e2464. doi:10.1002/nla.2464.