

Low-Rank Tensor Learning for Spatiotemporal Traffic and Mobility Data: A Survey

Mengying Lei^a, Lijun Sun^{a,*}

^aDepartment of Civil Engineering, McGill University, Montreal, Quebec, H3A 0C3, Canada

Abstract

Spatiotemporal traffic (speed, flow, occupancy, density) and mobility data (timestamped origin–destination trips) are high-dimensional, incomplete, and noisy, yet exhibit strong multiway dependencies across networks and time. Low-rank tensor learning (LRTL) exploits this latent structure to enable completion, denoising, and forecasting at scale. This survey presents a unified treatment of *structure-aware* LRTL for traffic data, spanning (i) *deterministic* approaches—classical tensor decompositions (e.g., CP, Tucker) and rank-surrogate formulations for completion—and (ii) *Bayesian probabilistic* models that provide principled uncertainty quantification (UQ). We emphasize explicit spatial and temporal modeling via graph (Laplacian) and variation local penalties, AR/VAR and state-space dynamics, Hankel embeddings, and kernel/GP priors, clarifying how these enter as regularizers (deterministic) versus priors (Bayesian probabilistic). We organize the literature around core transportation tasks—*imputation* [I], *spatiotemporal extrapolation* [E], *temporal prediction/forecasting* [P], *pattern discovery/anomaly detection* [PD/AD], and *regression* [R]—and synthesize connections, trade-offs, and common evaluation protocols. The review distills practical guidance for model selection and implementation, summarizes representative datasets and metrics, and outlines open challenges, including scalable UQ, topology-aware dynamics, handling regime shifts, and incorporating exogenous signals. Collectively, the survey clarifies when and how LRTL yields tangible gains in accuracy, robustness, and interpretability for real-world intelligent transportation systems.

Keywords: Spatiotemporal Traffic Data, Low-rank Tensor Learning, Tensor Decomposition/Factorization, Rank Surrogates, Bayesian Modeling, Intelligent Transportation Systems.

Contents

1	Introduction	2
2	Preliminaries	4
2.1	Notations	4
2.2	Basic operations	6
3	Deterministic LRTL for Spatiotemporal Traffic Data	8
3.1	Tensor decomposition for spatiotemporal traffic data	8
3.1.1	Decomposition techniques	8
3.1.2	Spatiotemporal modeling	11
3.2	Rank-surrogate relaxations for spatiotemporal traffic data	23
3.2.1	Rank-surrogate models	23
3.2.2	Spatiotemporal modeling	28
4	Bayesian Probabilistic LRTL for Spatiotemporal Traffic Data	32
4.1	Bayesian tensor factorization for spatiotemporal traffic data	33
4.1.1	Basics and fundamental models	33
4.1.2	Spatiotemporal modeling	37
4.2	Connection with deterministic TF	43
4.3	Comparison with Gaussian processes	45

*Corresponding author.

Email addresses: mengying.lei@mcgill.ca (Mengying Lei), lijun.sun@mcgill.ca (Lijun Sun)

5 LRTL for Transportation Tasks	47
5.1 LRTL for spatiotemporal traffic data imputation	48
5.2 LRTL for spatiotemporal traffic data extrapolation	50
5.3 LRTL for spatiotemporal traffic data prediction	53
5.4 LRTL for other tasks on spatiotemporal traffic data	55
5.5 Comparison and summary	57
6 Spatiotemporal Constraints in LRTL for Traffic Data	58
6.1 Spatial constraints	58
6.2 Temporal constraints	63
6.3 Comparison and summary	66
7 Evaluation and Benchmark Datasets	71
7.1 Evaluation analysis	71
7.1.1 Deterministic metrics	71
7.1.2 Probabilistic metrics	72
7.2 Benchmark datasets	73
8 Challenges and Future Directions	75
8.1 Challenges	75
8.2 Future directions	76
9 Conclusion	78

1. Introduction

Spatiotemporal traffic and mobility data—including travel speed, flow/volume, lane occupancy, density, and origin–destination (OD)/time-stamped trip records—now underlie a broad range of intelligent transportation systems (ITS) applications. At city scale, two regimes dominate: (i) *fixed-sensor traffic streams* with regular sampling but frequent missingness (malfunctions, communication failures, drift), and (ii) *mobility/OD data* represented as counts or trajectories that are extremely sparse and biased, often irregularly sampled, and sometimes perturbed for privacy. These characteristics complicate pipelines for travel-time estimation, OD completion and demand forecasting, routing, incident detection, and real-time control, and they motivate scalable methods that *exploit low-dimensional latent structure while explicitly encoding spatial and temporal dependencies* to enable robust imputation, extrapolation, prediction, pattern discovery, and anomaly detection.

Tensors—multidimensional generalizations of matrices—provide a natural representation for such structured data (Kolda and Bader, 2009). A day-level speed dataset, for example, can be organized as a third-order tensor with *location* \times *time-of-day* \times *day* modes, while OD flows are commonly modeled as *origin* \times *destination* \times *time* (ODT) or *origin* \times *destination* \times *time-of-day* \times *day* (ODTD) tensors. **Low-rank tensor learning (LRTL)** leverages intrinsic low-dimensional structure to enable compression, completion, denoising, and forecasting (Cichocki et al., 2015). Methodologically, LRTL comprises two complementary families: (i) *deterministic* optimization-based approaches—classical tensor decompositions such as CP and Tucker, together with rank-surrogate formulations (e.g., tensor nuclear-norm variants)—and (ii) *Bayesian probabilistic* formulations that provide principled uncertainty quantification (UQ) via hierarchical priors and posterior inference (Shi and Shen, 2023).

While “vanilla” low-rank models explain data through linear algebra alone, *structure-aware* LRTL injects domain knowledge as spatial and temporal constraints so that models respect network topology and recurrent temporal patterns rather than relying solely on algebraic fit. Typical mechanisms include graph/Laplacian regularization for spatial continuity on road networks, temporal-difference or autoregressive penalties for dynamics, Hankel (delay) embeddings for periodicity and regime shifts, and kernel/Gaussian process (GP) priors for spatiotemporal correlations (Shuman et al., 2013; Xiong et al., 2010; Wang et al., 2023b; Luttinen and Ilin, 2009; Lei et al., 2022b, 2024). For mobility applications, alternative tensor layouts (e.g., ODT/ODTD with additional modes such as user segment or travel mode) reveal low-rank communities and directional OD structure but accentuate challenges: long-tail sparsity, event-driven covariate shift, and privacy-induced distortions. Structure-aware LRTL addresses these through (i) spatial graphs defined over origins/destinations (and cross-mode couplings), (ii) temporal dynamics that encode

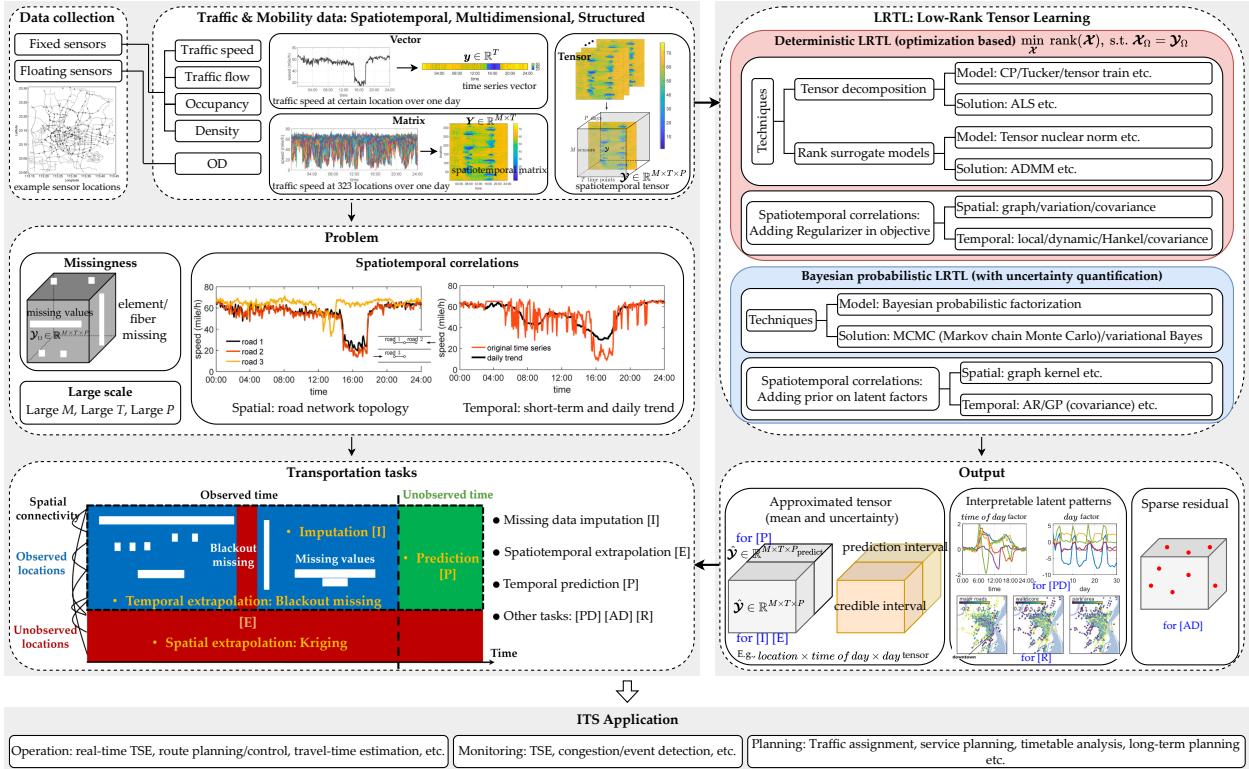


Figure 1: LRTL for spatiotemporal traffic and mobility data.

periodicity and regime changes, and (iii) Bayesian UQ to propagate uncertainty from sparse signals to downstream tasks.

Despite substantial advances, effectively incorporating spatiotemporal correlations and side information within both deterministic and probabilistic frameworks remains challenging at urban scale (Jacob et al., 2020; Chen et al., 2021b; Lei et al., 2024). Several surveys review tensors from general signal processing and machine learning perspectives (Kolda and Bader, 2009; Cichocki et al., 2015; Panagakis et al., 2021; Yokota, 2024; Tokcan et al., 2025), and Li et al. (2024d) focuses on large-scale processing for traffic. However, a unified, traffic-centric treatment that *systematically* covers deterministic optimization and Bayesian probabilistic LRTL *with an explicit focus on spatial and temporal structure* has not yet been provided.

This survey synthesizes LRTL for spatiotemporal traffic data with particular emphasis on integrating spatial and temporal correlations across modeling paradigms (Figure 1). We categorize methods into deterministic LRTL (tensor decompositions and rank-surrogate approaches) and Bayesian probabilistic LRTL, and for each we detail how spatial/temporal dependencies are encoded via regularizers (deterministic) or priors (Bayesian). We organize applications around core ITS tasks: missing-data [\mathbb{I}] imputation, spatiotemporal [\mathbb{E}] extrapolation, temporal [\mathbb{P}] prediction/forecasting (we use prediction and forecasting interchangeably in this paper), and [\mathbb{PD}] / [\mathbb{AD}] for pattern discovery and anomaly detection. Table 1 maps representative real-world scenarios (operation, monitoring, and planning) to data sources, challenges, tasks, and LRTL studies. We further compare spatiotemporal constraints across frameworks and clarify mathematical connections among them, together with scalability, rank selection, and UQ considerations; Figure 2 provides a taxonomy aligned with the paper organization.

The main contributions of this survey are threefold:

- 1) We present a systematic review and detailed categorization of state-of-the-art LRTL methods for spatiotemporal traffic data.
- 2) We provide a structured comparison between deterministic optimization-based and Bayesian probabilistic LRTL approaches, highlighting their advantages, limitations, and practical considerations.
- 3) We compare evaluation metrics and representative benchmark datasets used in LRTL studies for traffic data analysis, identify open challenges and suggest future research directions to advance tensor-based spatiotemporal traffic data analysis.

Table 1: Examples of LRTL for real-world traffic data.

Application		Data - Collection	Challenge	Task	LRTL research	
User	Requirement					
Operation	TMC, ATIS	Real-time TSE for routing/control	Traffic speed - Fixed sensor	Missingness, corruption, large-scale	Imputation, short-time prediction	Deng et al. (2016), Goulart et al. (2017), Chen et al. (2021a), Zhang et al. (2022)
	ATIS	ETA for route planning	Traffic speed - FCD	Extreme missingness, irregular sampling	Imputation	Ran et al. (2016a)
	QA	Fill gaps for maps	Traffic speed - Crowdsourced	Missingness, heterogeneity	Imputation	Chen et al. (2019b)
	TMC, QA	TSE, detector diagnose	Traffic flow - Loop detector	Missingness, corruption	Imputation	Deng et al. (2021a), Xu et al. (2023b), Chen et al. (2024c)
	Performance monitoring group	Travel-time estimation	Traffic flow - LPR, GPS cell-phone	Missingness	Imputation	Xing et al. (2023b)
Monitoring	QA	TSE	Traffic speed - GPS trajectory	Extreme missingness, noisy	Imputation	Yu et al. (2020)
	Operation analyst	Congestion/ event identification	Travel demand - Trip record	Missingness, distribution shift	Anomaly detection	Sofuo glu and Aviyente (2022)
Planning	Transit planning	Traffic assignment	OD - Taxi GPS	Missingness, sparsity	Prediction	Bhanu et al. (2020)
	Transit planning, transport agency	Service planning, timetable analysis	OD - Smart card data	Sparsity, large-scale	Prediction	Cheng et al. (2022)
	MPO, consultant	Pattern analysis, long-term planning	Travel demand - Trip record	Large-scale, heterogeneity	Regression	Lei et al. (2024)

Abbreviations: TSE: Traffic State Estimation; ETA: Estimated Time of Arrival; TMC: Traffic Management Center; ATIS: Advance Traveler Information System; QA: Quality Assurance; MPO: Metropolitan Planning Organization; FCD: Floating Car Data; LPR: License Plate Recognition.

The remainder of this survey is organized as follows. Section 2 introduces fundamental tensor notations and operations. Section 3 reviews tensor decomposition and tensor completion techniques for spatiotemporal traffic data from an optimization perspective. Section 4 presents Bayesian probabilistic low-rank tensor modeling approaches. Section 5 discusses the application of low-rank tensor learning methods across key transportation tasks, including imputation, extrapolation, prediction, and others. In Section 6, we summarize spatial and temporal constraints commonly incorporated into LRTL frameworks for traffic data. Section 7 outlines widely used evaluation metrics and benchmark datasets. We discuss current challenges and future directions in Section 8 and conclude the survey in Section 9. A companion repository of references and resources is available at <https://github.com/MengyingLei/LRTL4T-Survey>.

2. Preliminaries

We adopt tensor notations, operations, and terminology consistent with well-established studies such as Kolda and Bader (2009); Cichocki et al. (2015). This section introduces the essential tensor concepts used throughout the survey. A summary of key symbols of notations and operations is provided in Table 2.

2.1. Notations

Order. The *order* (also referred to as mode or way) of a tensor indicates the number of its dimensions. For a real-world traffic data, different modes may represent dimensions such as space (locations), time, and covariate variables (Chen et al., 2021b; Lei et al., 2024). We use boldface lowercase letters to denote vectors (first-order tensors), e.g., $\mathbf{x} \in \mathbb{R}^M$, boldface uppercase letters to denote matrices (second-order tensors),

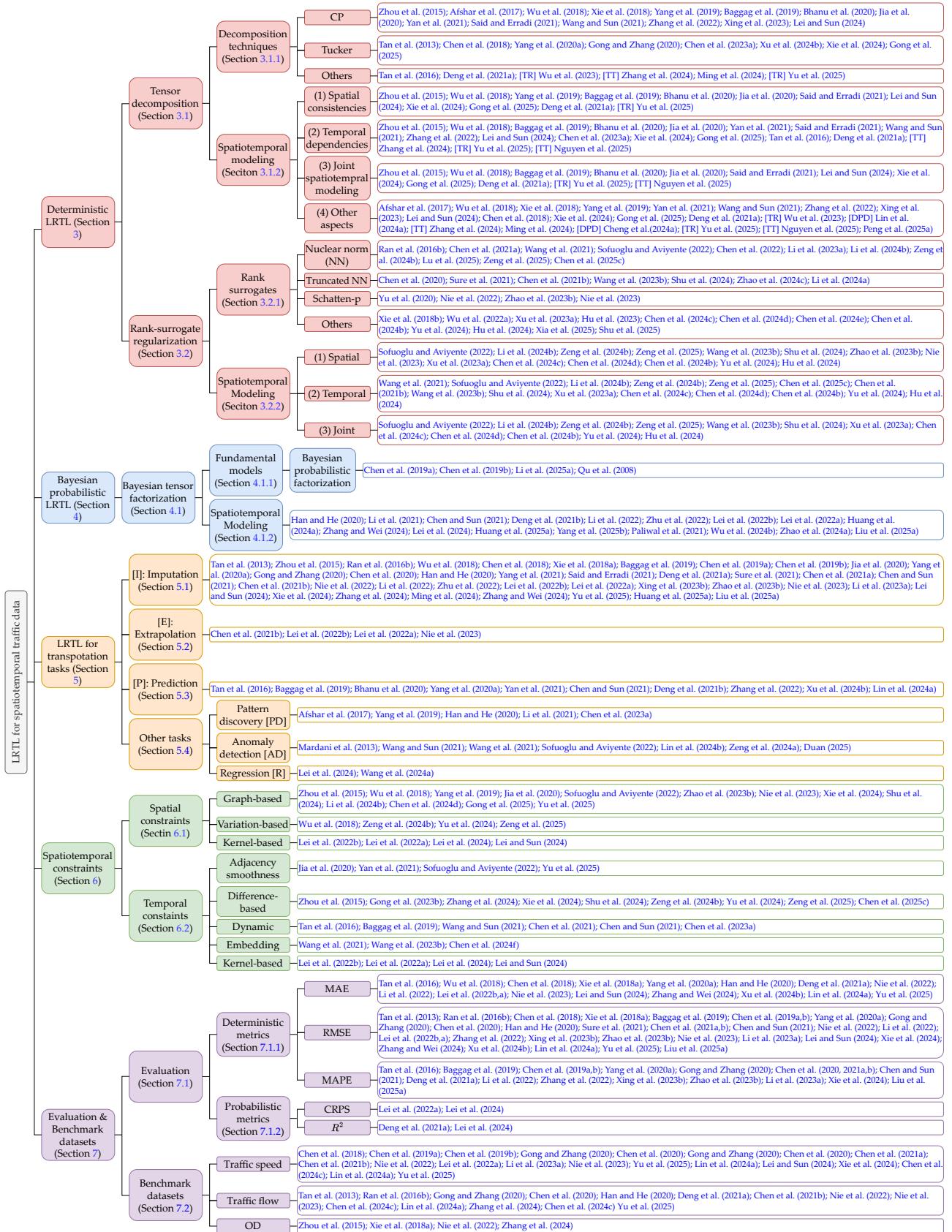


Figure 2: Taxonomy of LRTL research for spatiotemporal traffic data.

Table 2: Symbols for basic tensor notations and operations.

Symbol	Notation
$\mathbb{R}^{I_1 \times \dots \times I_D}$	D -dimensional real tensor space
\boldsymbol{x}	Vector (column by default)
\boldsymbol{X}	Matrix
$\boldsymbol{\chi}$	Tensor
$x_m, x_{mt}, x_{i_1 \dots i_D}$	Entry of a vector, matrix, tensor
$\boldsymbol{x}_{:t}, \boldsymbol{x}_{m:}$	Column, row of a matrix
$\boldsymbol{x}_{:tp}, \boldsymbol{x}_{m:p}, \boldsymbol{x}_{mt:}$	Fiber of a tensor
$\boldsymbol{X}_{m::}, \boldsymbol{X}_{:t::}, \boldsymbol{X}_{::p::}$	Slice of a tensor
Symbol	Operation
$\boldsymbol{X}_{(d)}$	Mode- d unfolding of tensor $\boldsymbol{\chi}$
$\text{vec}(\boldsymbol{X})$	Vectorization of matrix \boldsymbol{X}
$\text{vec}(\boldsymbol{\chi})$	Vectorization of tensor $\boldsymbol{\chi}$
$\boldsymbol{A} \times_d \boldsymbol{B}$	Mode- d product of tensor \boldsymbol{A} and matrix \boldsymbol{B}
$\boldsymbol{A} \otimes \boldsymbol{B}$	Kronecker product of matrices \boldsymbol{A} and \boldsymbol{B}
$\boldsymbol{A} \odot \boldsymbol{B}$	Khatri-Rao product of matrices \boldsymbol{A} and \boldsymbol{B}
$\boldsymbol{A} \circledast \boldsymbol{B}$	Hadamard product of matrices \boldsymbol{A} and \boldsymbol{B}
$\boldsymbol{a} \circ \boldsymbol{b}$	Outer product of vectors \boldsymbol{a} and \boldsymbol{b}
$\ \boldsymbol{x}\ _2$	ℓ_2 -norm of vector \boldsymbol{x}
$\ \boldsymbol{X}\ _F$	Frobenius norm of matrix \boldsymbol{X}
$\text{diag}(\boldsymbol{x})$	Diagonal matrix with diagonal elements being \boldsymbol{x}
$\text{diag}(\boldsymbol{X})$	Vector of diagonal elements of \boldsymbol{X}
$\text{tr}(\boldsymbol{X})$	Trace of matrix \boldsymbol{X}

e.g., $\boldsymbol{X} \in \mathbb{R}^{M \times T}$, and boldface script letters to denote higher-order tensors (order ≥ 3), e.g., $\boldsymbol{\chi} \in \mathbb{R}^{I_1 \times \dots \times I_D}$, where $D \geq 3$.

Entry. An *entry* (or element) is a scalar value in a tensor that represents the smallest unit of data. For a D th-order tensor, each entry is indexed by D dimensions. We denote the m th entry in a vector $\boldsymbol{x} \in \mathbb{R}^M$ by x_m , where $m \in \{1, \dots, M\}$, the (m, t) th entry of a matrix $\boldsymbol{X} \in \mathbb{R}^{M \times T}$ by x_{mt} , where $m \in \{1, \dots, M\}$ and $t \in \{1, \dots, T\}$, and the (i_1, \dots, i_D) th entry of a D th-order tensor $\boldsymbol{\chi} \in \mathbb{R}^{I_1 \times \dots \times I_D}$ (with $D \geq 3$) by $x_{i_1 \dots i_D}$, where $i_d \in [1, I_d]$ for $d = 1, \dots, D$. For example, consider a third-order traffic speed data tensor structured by *location* \times *time of day* \times *day*, each entry represents the traffic speed recorded at a specific location and time point on a given day.

Fiber. A *fiber* is a one-dimensional sub-array of a tensor, obtained by fixing all indices except one. In a matrix $\boldsymbol{X} \in \mathbb{R}^{M \times T}$, the t th column and m th row—denoted by $\boldsymbol{x}_{:t} \in \mathbb{R}^M$ and $\boldsymbol{x}_{m:} \in \mathbb{R}^T$, respectively, where $m = 1, \dots, M$, $t = 1, \dots, T$,—are referred to as the mode-1 and mode-2 fibers. Note that throughout this paper, all vectors are considered column vectors by default. For a third-order tensor $\boldsymbol{\chi} \in \mathbb{R}^{M \times T \times P}$, there are three types of fibers: mode-1 (column) fibers $\boldsymbol{x}_{:tp} \in \mathbb{R}^M$, mode-2 (row) fibers $\boldsymbol{x}_{m:p} \in \mathbb{R}^T$, and mode-3 (tube) fibers $\boldsymbol{x}_{mt:} \in \mathbb{R}^P$. Each of these corresponds to fixing all but one index along the respective mode. The fiber structures along the three dimensions of a third-order traffic speed tensor are illustrated in Figure 3.

Slice. A *slice* is a two-dimensional sub-array of a tensor, defined by fixing all indices except two. A third-order tensor $\boldsymbol{\chi} \in \mathbb{R}^{M \times T \times P}$ has three types of slices: horizontal slices (fixing the first index) $\boldsymbol{X}_{m::} \in \mathbb{R}^{T \times P}$, lateral slices (fixing the second index) $\boldsymbol{X}_{:t::} \in \mathbb{R}^{M \times P}$, and frontal slices (fixing the third index) $\boldsymbol{X}_{::p::} \in \mathbb{R}^{M \times T}$. Each slice provides a matrix view of the tensor along a specific pair of modes. We illustrate the three slice types of a third-order traffic speed tensor in Figure 3.

2.2. Basic operations

Unfolding. *Unfolding* (also called matricization) is the process of rearranging a tensor into a matrix by flattening it along a specific dimension. For a D th-order tensor $\boldsymbol{\chi} \in \mathbb{R}^{I_1 \times \dots \times I_D}$, we denote its mode- d unfolding (for $d = 1, \dots, D$) as $\boldsymbol{X}_{(d)} \in \mathbb{R}^{I_d \times \frac{N}{I_d}}$, where $N = \prod_{d=1}^D I_d$ is the total number of elements in the tensor. The

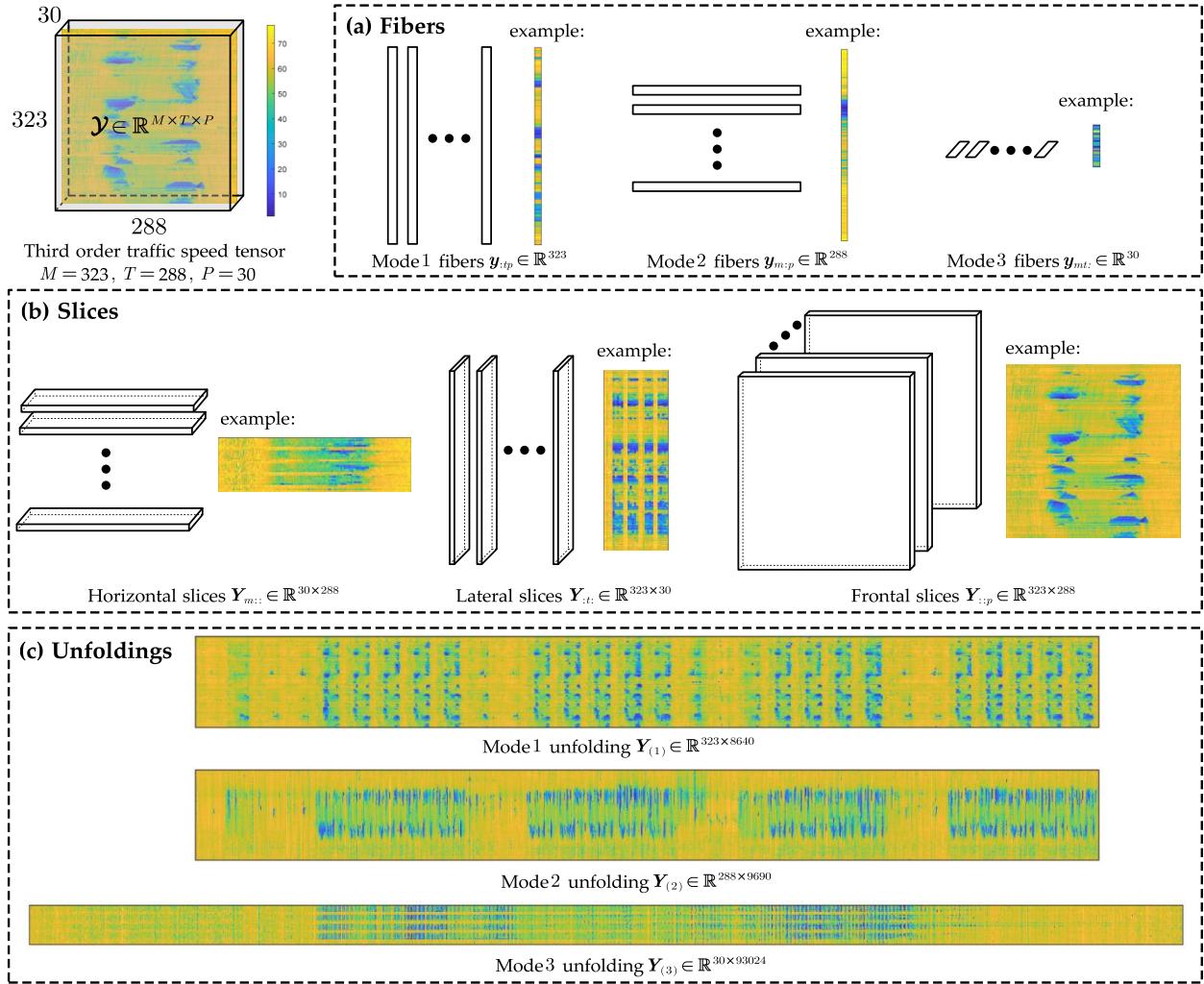


Figure 3: Illustration of tensor notations using the Seattle traffic speed dataset. We represent the data as a third-order *location* \times *time of day* \times *day* tensor \mathcal{Y} with the size of $323 \times 288 \times 30$. The figure shows: (a) Fibers; (b) Slices; and (c) Mode- d unfolding matrices for $d = 1, 2, 3$.

vectorization of a matrix X , denoted by $\text{vec}(X)$, stacks all column vectors in X as a single column vector, and the vectorization of a tensor \mathcal{X} is defined by $\text{vec}(\mathcal{X}) = \text{vec}(X_{(1)})$. We explain mode-wise unfolding matrices for a third-order traffic speed tensor constructed using Seattle traffic speed dataset (available at <https://github.com/zhiyongc/Seattle-Loop-Data>) in Figure 3.

Folding. *Folding* (also known as *tensorization*) is the inverse operation of unfolding. It reconstructs a higher-order tensor from its unfolded matrix by reshaping the matrix back into the original tensor dimensions along a specified mode.

Tensor matrix multiplication. The tensor d -mode product defines the multiplication of a D th-order tensor with a matrix (or a vector, as a special case of matrix) along its d th mode, where $d = 1, \dots, D$. Given a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_D}$ and a matrix $A \in \mathbb{R}^{J \times I_d}$, the mode- d product is denoted by $\mathcal{X} \times_d A$ and results in a tensor of size $I_1 \times \dots \times I_{d-1} \times J \times I_{d+1} \times \dots \times I_D$. Element-wise, there has $(\mathcal{X} \times_d A)_{i_1 \dots i_{d-1} j i_{d+1} \dots i_D} = \sum_{i_d=1}^{I_d} x_{i_1 i_2 \dots i_d} \times a_{j i_d}$.

Kronecker product. Given two matrices $A \in \mathbb{R}^{M \times T}$ and $B \in \mathbb{R}^{P \times Q}$, the *Kronecker product* of A and B is defined as $A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1T}B \\ \vdots & \ddots & \vdots \\ a_{M1}B & \cdots & a_{MT}B \end{bmatrix} \in \mathbb{R}^{MP \times TQ}$.

Khatri-Rao product. Let $A = [a_{:,1}, \dots, a_{:,T}] \in \mathbb{R}^{M \times T}$ and $B = [b_{:,1}, \dots, b_{:,Q}] \in \mathbb{R}^{P \times Q}$ be two matrices with the same number of columns, i.e., $T = Q$. The *Khatri-Rao product* of A and B is defined as the column-wise

Kronecker product: $\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_{\cdot 1} \otimes \mathbf{b}_{\cdot 1}, \dots, \mathbf{a}_{\cdot T} \otimes \mathbf{b}_{\cdot T}] \in \mathbb{R}^{MP \times T}$. Each column of the resulting matrix is the Kronecker product of the corresponding columns from \mathbf{A} and \mathbf{B} .

Hadamard product. The *Hadamard product* refers to the element-wise multiplication of two matrices of the same size. Given two $M \times T$ commensurate matrices \mathbf{A} and \mathbf{B} , their Hadamard product is defined as

$$\mathbf{A} \circledast \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & \cdots & a_{1T}b_{1T} \\ \vdots & \ddots & \vdots \\ a_{M1}b_{M1} & \cdots & a_{MT}b_{MT} \end{bmatrix} \in \mathbb{R}^{M \times T}.$$

Each entry of the resulting matrix is the product of the corresponding entries in \mathbf{A} and \mathbf{B} .

Outer product. The *outer product* of two vectors $\mathbf{a} \in \mathbb{R}^M$ and $\mathbf{b} \in \mathbb{R}^T$ is denoted by $\mathbf{a} \circ \mathbf{b}$, and produces a $M \times T$ matrix, where each entry is the product of an element from \mathbf{a} and an element from \mathbf{b} : $(\mathbf{a} \circ \mathbf{b})_{mt} = a_m \times b_t$. In matrix form: $\mathbf{a} \circ \mathbf{b} = \begin{bmatrix} a_1b_1 & \cdots & a_1b_T \\ \vdots & \ddots & \vdots \\ a_Mb_1 & \cdots & a_Mb_T \end{bmatrix} \in \mathbb{R}^{M \times T}$.

This operation is commonly used in tensor decomposition such as CP decomposition, where rank-one tensors are formed by outer products of vectors. *Norm.* The ℓ_2 -norm of a vector $\mathbf{x} \in \mathbb{R}^M$ is defined as the square root of the sum of its squared elements: $\|\mathbf{x}\|_2 = \sqrt{\sum_{m=1}^M x_m^2}$. The *Frobenius norm* of a matrix $\mathbf{X} \in \mathbb{R}^{M \times T}$ is similarly defined as the square root of the sum of the squares of all its entries: $\|\mathbf{X}\|_F = \sqrt{\sum_{m=1}^M \sum_{t=1}^T x_{mt}^2}$. These norms are commonly used in optimization objective functions and regularization terms or tensor-based models.

With these standardized definitions and operations, the subsequent sections are well-positioned to systematically introduce and compare a broad range of LRTL methods developed for spatiotemporal traffic data analysis.

3. Deterministic LRTL for Spatiotemporal Traffic Data

Deterministic LRTL methods can be categorized into two types: (1) **tensor decomposition/factorization**, which expresses a data tensor using a small number of latent factors; and (2) **rank-surrogate-based** models, which solve low-rank estimation by minimizing a proxy for tensor rank (Li et al., 2017). In this section, we systematically review both categories—**tensor decomposition** (Section 3.1) and **rank-surrogate models** (Section 3.2)—with an emphasis on applications to spatiotemporal traffic data. We highlight how these methods integrate spatial and temporal consistencies via regularization techniques. An illustration of the two methodological tracks on a third-order tensor is shown in Figure 4.

3.1. Tensor decomposition for spatiotemporal traffic data

Tensor decomposition (a.k.a. **tensor factorization**, TF) represents a multiway tensor with a small number of latent components along each mode, yielding compact and often interpretable structure (Kolda and Bader, 2009; Cichocki et al., 2015). TF has been widely used for imputation and pattern discovery in high-dimensional traffic datasets (Li et al., 2024d; Shi et al., 2020a). We review representative techniques—CANDECOMP/PARAFAC (CP) and Tucker—and how they are adapted to spatiotemporal traffic modeling.

3.1.1. Decomposition techniques

Formulation. Let $\mathcal{Y} \in \mathbb{R}^{I_1 \times \dots \times I_D}$ denote a D th-order data tensor, for example, $D = 3$ for a *location* \times *time of day* \times *day* traffic speed dataset. Low-rank tensor decomposition compresses \mathcal{Y} by utilizing multiway correlations (Kolda and Bader, 2009; Cichocki et al., 2015; Yokota, 2024). Two standard paradigms are widely used: CP and Tucker.

CP (rank- R). The rank- R CP model decomposes \mathcal{Y} as a sum of R rank-1 outer products of latent vectors (Carroll and Chang, 1970; Harshman et al., 1970; Kiers, 2000):

$$\mathcal{Y} \approx \sum_{r=1}^R \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \dots \circ \mathbf{a}_r^{(D)} = [\![\boldsymbol{\lambda}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(D)}]\!], \quad (1)$$

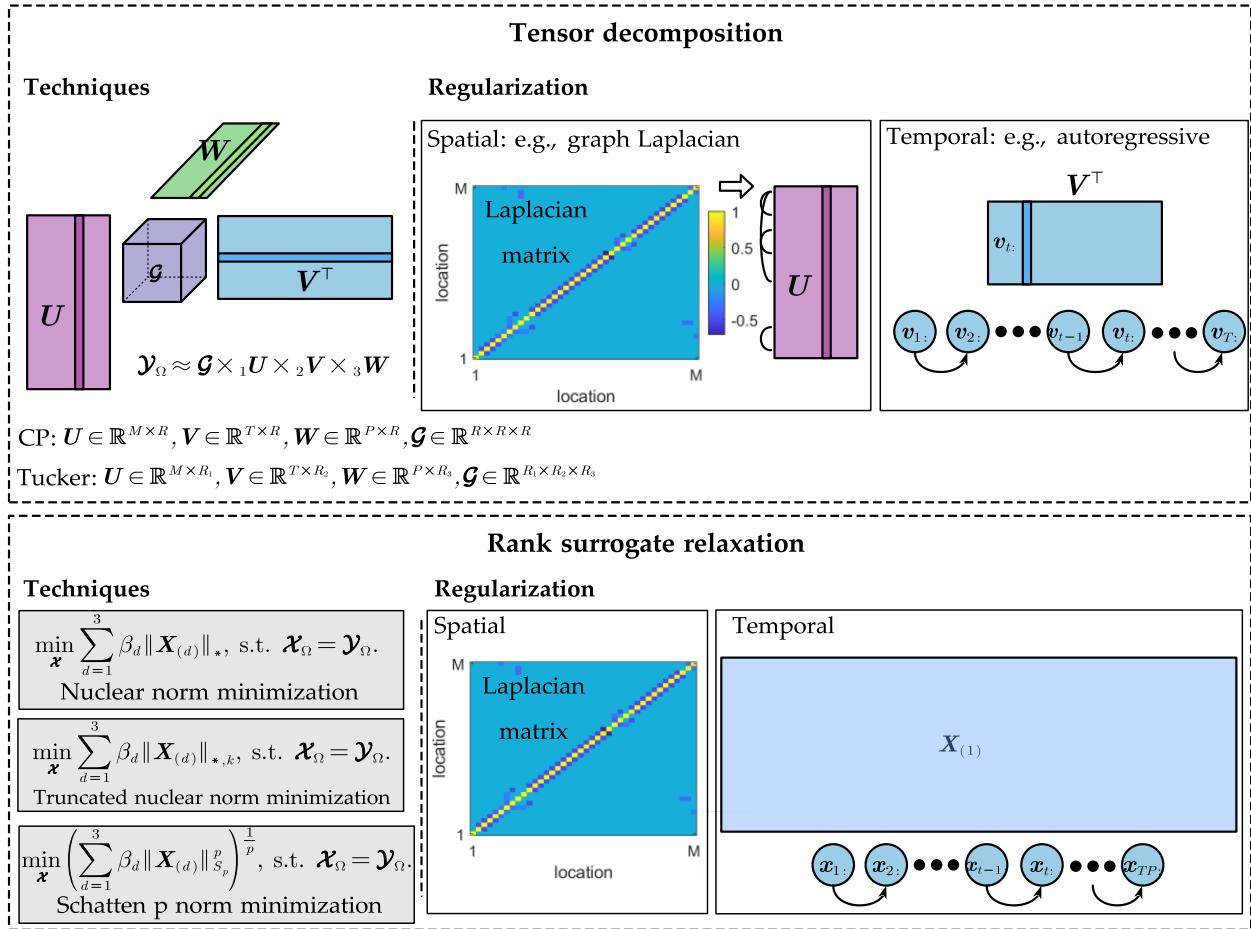


Figure 4: Illustration of tensor decomposition and rank-surrogate minimization on a third-order incomplete tensor \mathcal{Y}_{Ω} with dimensions $\text{location} \times \text{time of day} \times \text{day}$ of size $M \times T \times P$.

where λ_r is a scalar weight, $\mathbf{a}_r^{(d)} \in \mathbb{R}^{I_d}$ is the r th latent factor for mode- d , $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_R]^T \in \mathbb{R}^R$, and $\mathbf{A}^{(d)} = [\mathbf{a}_1^{(d)}, \dots, \mathbf{a}_R^{(d)}] \in \mathbb{R}^{I_d \times R}$ are mode- d factor matrices. The mode- d unfolding $\mathbf{Y}_{(d)}$ and vectorization $\text{vec}(\mathbf{Y})$ can be obtained by Khatri-Rao product with $\mathbf{Y}_{(d)} = \mathbf{A}^{(d)} \boldsymbol{\Lambda} (\mathbf{A}^{(D)} \odot \dots \odot \mathbf{A}^{(d+1)} \odot \mathbf{A}^{(d-1)} \odot \dots \odot \mathbf{A}^{(1)})^T$ and $\text{vec}(\mathbf{Y}) = (\mathbf{A}^{(D)} \odot \mathbf{A}^{(D-1)} \odot \dots \odot \mathbf{A}^{(1)}) \boldsymbol{\lambda}$, respectively, where $\boldsymbol{\Lambda} \in \mathbb{R}^{R \times R}$ is a diagonal matrix with diagonal elements being $\boldsymbol{\lambda}$. These expressions are equivalent ways to represent the CP decomposition model and are useful for deriving solution algorithms.

Tucker (multilinear rank (R_1, \dots, R_D)). The multilinear rank (R_1, \dots, R_D) Tucker model approximates \mathbf{Y} by a multilinear dense core tensor \mathbf{G} of size $R_1 \times R_2 \times \dots \times R_D$ (typically $R_d \ll I_d$ for all $d \in [1, D]$) and factor matrices $\mathbf{A}^{(d)} \in \mathbb{R}^{I_d \times R_d}$ (Tucker, 1963, 1966):

$$\begin{aligned} \mathbf{Y} &\approx \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \dots \sum_{r_D=1}^{R_D} g_{r_1 r_2 \dots r_D} \left(\mathbf{a}_{r_1}^{(1)} \circ \mathbf{a}_{r_2}^{(2)} \circ \dots \circ \mathbf{a}_{r_D}^{(D)} \right) \\ &= \mathbf{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \dots \times_D \mathbf{A}^{(D)} = [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(D)}], \end{aligned} \quad (2)$$

where $g_{r_1 \dots r_D}$ are the entries of the core tensor \mathbf{G} , and $\mathbf{a}_{r_d}^{(d)} \in \mathbb{R}^{I_d}$ is the r_d th column of $\mathbf{A}^{(d)}$. In Tucker decomposition, the core \mathbf{G} interacts with mode- d factor matrix $\mathbf{A}^{(d)}$ by tensor-matrix multiplication, i.e., \times_d . Similar as CP decomposition, the mode- d unfolding and vectorization of \mathbf{Y} can be computed by Kronecker product with forms $\mathbf{Y}_{(d)} = \mathbf{A}^{(d)} \mathbf{G}_{(d)} (\mathbf{A}^{(D)} \otimes \mathbf{A}^{(D-1)} \otimes \dots \otimes \mathbf{A}^{(d+1)} \otimes \mathbf{A}^{(d-1)} \otimes \dots \otimes \mathbf{A}^{(1)})^T$ and $\text{vec}(\mathbf{Y}) = (\mathbf{A}^{(D)} \otimes \mathbf{A}^{(D-1)} \otimes \dots \otimes \mathbf{A}^{(1)}) \text{vec}(\mathbf{G})$, respectively.

Ranks and identifiability. Both CP and Tucker can be seen as higher-order generalizations of matrix

singular value decomposition (SVD), designed to extract principal components from multi-way array data (Kolda and Bader, 2009). The smallest R for which Eq. (1) holds exactly is the *CP rank*; computing it is known to be NP-hard (Håstad, 1990). In Tucker, the d -rank of \mathcal{Y} is defined as the column rank of the mode- d unfolding $\mathbf{Y}_{(d)}$ (Kruskal, 1989; De Lathauwer et al., 2000). For a core tensor of minimal size $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_D}$, R_d represents the d -rank, and the tuple (R_1, R_2, \dots, R_D) is referred to as the *multilinear rank* of \mathcal{Y} . CP can be viewed as a special case of Tucker where the core tensor \mathcal{G} is constrained to be super-diagonal and all mode ranks are equal, i.e., $R_1 = \dots = R_D$. A CP model is essentially unique (up to column permutation and per-component scaling) under Kruskal’s condition; whereas Tucker decomposition is non-unique up to invertible rotations within each mode subspace (Kolda and Bader, 2009). The tensor ranks quantify the intrinsic low-dimensional structure of the data. It is important to choose an appropriate rank: overly small ranks can overlook meaningful data structure (underfitting), while overly large ranks can lead to overfitting. In practice, CP rank is often set using domain knowledge or cross-validation to balance reconstruction accuracy and model complexity (Zhang, 2019). For Tucker, one can leverage singular-value decay in the mode unfoldings and/or apply sparsity regularization or thresholding on the core to automatically select components and promote parsimony (Grasedyck, 2010; Yokota et al., 2016a; Gong et al., 2025).

Computation. Let Ω be the index set of observed entries and $(\cdot)_{\Omega}$ the projection onto Ω . Estimating CP/Tucker typically solves a masked least-squares fit (objective) on the observed entries:

$$\begin{aligned} \text{CP: } & \min_{\boldsymbol{\lambda}, \{\mathbf{A}^{(d)}\}_{d=1}^D} \frac{1}{2} \left\| \left(\mathcal{Y} - [\mathbf{\Lambda}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(D)}] \right)_{\Omega} \right\|_F^2, \\ \text{Tucker: } & \min_{\mathcal{G}, \{\mathbf{A}^{(d)}\}_{d=1}^D} \frac{1}{2} \left\| \left(\mathcal{Y} - [\mathcal{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(D)}] \right)_{\Omega} \right\|_F^2. \end{aligned} \quad (3)$$

These objectives are usually solved with alternating least squares (ALS), which iteratively updates one variable at a time by solving a least-squares problem while keeping the others fixed (Carroll and Chang, 1970; Harshman et al., 1970). The subproblems are often formed using Khatri-Rao or Kronecker product based unfolding formulations introduced earlier. For example, the subproblem for $\mathbf{A}^{(d)}$ in a CP model solves

$$\min_{\mathbf{A}^{(d)}} \frac{1}{2} \left\| \left(\mathbf{Y}_{(d)} \right)_{\Omega} - \left(\mathbf{A}^{(d)} \boldsymbol{\Lambda} \left(\mathbf{A}^{(D)} \odot \dots \odot \mathbf{A}^{(d+1)} \odot \mathbf{A}^{(d-1)} \odot \dots \odot \mathbf{A}^{(1)} \right)^{\top} \right)_{\Omega} \right\|_F^2,$$

where $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\lambda})$. CP is simple and scalable but can converge slowly when the rank R is large (Phan et al., 2013; Battaglino et al., 2018). Practical accelerations include line search (Rajih et al., 2008), improved/randomized initialization (Battaglino et al., 2018), and structural regularization such as nonnegativity or smoothness (Afshar et al., 2018). Tucker can be computed by higher-order SVD (HOSVD) for an initial orthogonal approximation (De Lathauwer et al., 2000), then refined by higher-order orthogonal iteration (HOOI) (Kolda and Bader, 2009). The computational costs increase with data size and ranks (Kolda and Sun, 2008; Papalexakis et al., 2016). To address scalability, randomized/incremental SVD (Minster et al., 2020; Ahmadi-Asl et al., 2021; Sun et al., 2008) and distributed/parallel implementations have been developed (Minster et al., 2024; Papalexakis et al., 2016).

Application to traffic data. Traffic measurements naturally form multidimensional spatiotemporal tensors with strong spatial and temporal structure (Shekhar et al., 2015), e.g., a *location* \times *time of day* \times *day* tensor for loop-detector speeds (Lei et al., 2022b,a) and an *origin* \times *destination* \times *time* tensor for OD flow. Tensor decompositions provide effective frameworks to uncover latent structures and dependencies in spatiotemporal traffic data (Wu et al., 2018; Chen et al., 2019b, 2023a). For instance, CP often yields interpretable components: spatial latent factors separating regions with high/low traffic levels; temporal factors for diurnal trends such as morning and afternoon rush hours, mid-day off-peak periods; and day factors distinguishing weekdays and weekends (Lei et al., 2022b). Tucker offers additional flexibility: the latent factors are analogous, while the core encodes interactions among components across modes (Chen et al., 2018). Furthermore, decompositions support missing data completion (Lei et al., 2022a; Lei and Sun, 2024), compression for storage and efficient downstream learning (Wu et al., 2018), and unsupervised clustering (Lei et al., 2024), etc.

Beyond CP and Tucker, many decomposition variants have also been explored for spatiotemporal traffic data analysis. Examples include spectral transform domain graph tensor-SVD (GT-SVD) approach (Deng et al., 2021a), dimension preserved decomposition (DPD) (Lin et al., 2024a; Chen et al., 2025a), dynamic low-rank decomposition for time-evolving data (Tan et al., 2016), tensor triple decomposition (TD) (Qi et al., 2021; Ming et al., 2024), and advanced formats such as tensor train (TT) (Zhang et al., 2024), tensor ring

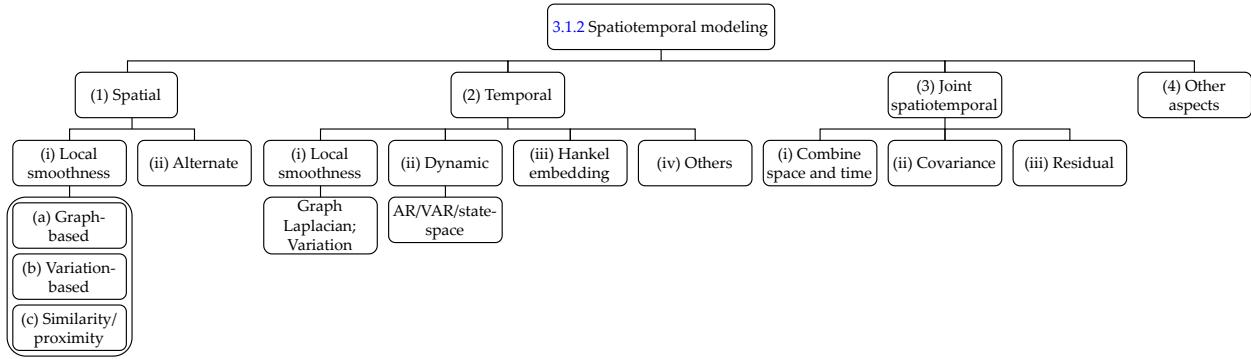


Figure 5: Outline of spatiotemporal modeling strategies in tensor decomposition (Section 3.1.2).

(TR) (Yu et al., 2025; Liu et al., 2025d), and tensor network decomposition (Zheng et al., 2025). Overall, low-rank factorizations capitalize on redundancy across space and time, a hypothesis aligned with temporal periodicity and spatial network patterns in traffic data (Li et al., 2015).

In general, regularization on latent variables is usually integrated into the objective functions in Eq. (3) to improve convergence and model performance (Sørensen et al., 2012; Lim and Comon, 2009; Veganzones et al., 2015; Goulart et al., 2015; Favier and de Almeida, 2014). Accurate modeling of spatiotemporal traffic data also requires incorporating domain structure, e.g., road network topology and diurnal/weekly dynamics, commonly introduced via corresponding regularizers (Wang et al., 2018) (see the following Section 3.1.2).

3.1.2. Spatiotemporal modeling

In this subsection, we review how spatial and temporal side information is integrated into low-rank decomposition models for traffic data, covering: (1) spatial consistencies; (2) temporal dependencies; (3) joint spatiotemporal modeling; and (4) other aspects, followed by (5) a short summary.

For simplicity, we arrange the traffic data as a spatiotemporal matrix $\mathbf{Y} \in \mathbb{R}^{M \times T}$ (e.g., M locations \times T time points). We seek a low-rank factorization $\mathbf{Y} \approx \mathbf{U}\mathbf{V}^\top$, with $\mathbf{U} \in \mathbb{R}^{M \times R}$ and $\mathbf{V} \in \mathbb{R}^{T \times R}$ being the spatial and temporal latent factors, respectively, and R the latent dimension (rank). We use $\mathcal{R}(\cdot)$ to denote the regularization term. The masked objective can be uniformly written as:

$$\min_{\mathbf{U}, \mathbf{V}} \frac{1}{2} \left\| (\mathbf{Y} - \mathbf{U}\mathbf{V}^\top)_{\Omega} \right\|_F^2 + \rho_s \mathcal{R}(\mathbf{U}) + \rho_t \mathcal{R}(\mathbf{V}), \quad (4)$$

where $\rho_s, \rho_t > 0$ are regularization parameters controlling the smoothness strengths. For higher-order data, one can replace $\mathbf{U}\mathbf{V}^\top$ with a CP or Tucker reconstruction and apply mode-specific regularizers to the corresponding factor matrices; the masked/weighted loss is unchanged. We next introduce commonly used choices of spatial regularizer $\mathcal{R}(\mathbf{U})$ and temporal regularizer $\mathcal{R}(\mathbf{V})$. The outline of introduced spatiotemporal modeling strategies for tensor decomposition is given in Figure 5.

(1) Incorporating spatial consistencies. Traffic observations over a road network exhibit strong spatial interdependence—measurements from nearby locations or connected road segments tend to co-vary. Vanilla CP/Tucker treat the location mode simply as an index and do not enforce that nearby locations have similar latent factors (Chen and Sun, 2021; Lei et al., 2022b). To encode spatial consistencies, tensor objectives introduce spatial regularizers that encourage local smoothness on the spatial factors so the latent representation varies gradually over the physical network. We group these approaches into two categories:

- (i) Local smoothness regularization**, enforcing spatial coherence among neighboring locations, which includes:
 - (a) Graph-based penalties.** Using graph Laplacian matrices derived from the road network topology (optionally weighted by connectivity or distance) to regularize the spatial factor (Baggag et al., 2019; Rao et al., 2015; Gong et al., 2025).
 - (b) Variation-based penalties.** Applying discrete difference operators that discourage abrupt changes between neighboring locations, e.g., total variation (TV) or quadratic variation (QV) (Yokota et al., 2016b).

- (c) **Similarity/proximity formulations.** Custom similarity or distance-based weight matrices that encode spatial proximity when an explicit road graph is unavailable.
- (ii) **Alternate spatial information** (beyond adjacency): leveraging side information such as additional covariates, or learned covariance/kernel structures to regularize the spatial mode and capture longer-range or functional similarity.

(i) Local smoothness regularization **(a) Graph-based regularization.** A common way to integrate spatial structure into low-rank decomposition is to penalize lack of smoothness over a road network graph via the graph Laplacian (Baggag et al., 2019). The idea, which is popular in recommendation systems and adapted to traffic, is to encourage entities connected in a graph to have similar latent representations, thereby enforcing local consistency (Rao et al., 2015). In traffic data, sensors/links/intersections are modeled as nodes, and edges encode topological proximity such as adjacent road segments. From the network adjacency matrix, one can derive the Laplacian, then add a quadratic Laplacian penalty on the spatial factor matrix to encourage nearby locations to have similar latent factors, preserving local spatial correlations (Wang et al., 2018). In practice, edge weights can be binary or distance/affinity-based, and normalized Laplacians are often used to improve conditioning. The penalty is computationally efficient and integrates seamlessly with CP/Tucker objectives.

Formally, let $\mathcal{G} = (S, E)$ be a graph with $|S| = M$ vertices (nodes, e.g., sensors, intersections, or locations). An edge $e_{ij} = (s_i, s_j) \in E$ connects two nodes s_i and s_j ; in traffic, edges typically reflect topological adjacency (e.g., adjacent road segments). Let $\mathbf{Adj}_s = [\alpha_{ij}] \in \mathbb{R}^{M \times M}$ denote the spatial adjacency matrix (binary or distance-weighted), and let $\mathbf{D} = \text{diag}(\mathbf{Adj}_s \mathbf{1}_{M \times 1})$ be the degree matrix, where $d_{ii} = \sum_j \alpha_{ij}$ and $\mathbf{1}_{M \times 1}$ is the $M \times 1$ matrix of all-ones. The unnormalized graph Laplacian is $\mathbf{Lap}_s = \mathbf{D} - \mathbf{Adj}_s$. One may use a normalized variant $\mathbf{Lap}'_s = \mathbf{D}^{-\frac{1}{2}} \mathbf{Lap}_s \mathbf{D}^{-\frac{1}{2}}$ for better conditioning. The graph Laplacian regularizer on spatial latent factors \mathbf{U} can be represented as:

$$\mathcal{R}_{\text{Lap}}(\mathbf{U}; \mathbf{Lap}_s) = \frac{1}{2} \text{tr}(\mathbf{U}^\top \mathbf{Lap}_s \mathbf{U}) = \frac{1}{2} \sum_{(s_i, s_j) \in E} \alpha_{ij} \|\mathbf{u}_{i:} - \mathbf{u}_{j:}\|_2^2, \quad (5)$$

which penalizes large differences between the latent factors of any two connected locations and thus preserving local spatial patterns. Computationally, the Laplacian regularizer is quadratic in \mathbf{U} and can leverage the sparsity of \mathbf{Lap}_s , enabling efficient ALS/gradient updates.

Graph Laplacian regularization has been extensively applied in collaborative filtering (Rao et al., 2015) and is effective for traffic applications requiring network spatial consistency such as kriging, i.e., spatial extrapolation (interpolation). Lots of studies apply graph regularized tensor decomposition models for traffic data: graph-regularized MF for traffic speed imputation (Wang et al., 2018); fused CP with a manifold (graph) constraint for flow imputation (Wu et al., 2018); CP with graph smoothing for imputation of speed/volume (Baggag et al., 2019; Bhanu et al., 2020); Tucker with graph regularization under high missing rates (Gong et al., 2025; Xie et al., 2024); and TR with graph constraints for traffic states imputation (Yu et al., 2025).

(b) Variation-based regularization. Local smoothness can also be imposed by penalizing discrete difference of the latent factors. A popular instance is the Smooth PARAFAC (SPC) approach, which adds either total variation (TV) or quadratic variation (QV) penalties to CP factors by operating on their discrete gradients (Yokota et al., 2016b). Define first-order difference operator $\mathbf{L}_{\text{diff}} \in \mathbb{R}^{(M-1) \times M}$, whose rows compute

consecutive differences $[\dots, 1, -1, \dots]$, i.e., $\mathbf{L}_{\text{diff}} = \begin{bmatrix} 1 & -1 \\ & 1 & -1 \\ & & \ddots & \ddots \\ & & & 1 & -1 \end{bmatrix}$. The variation regularizer on \mathbf{U} is

$$\mathcal{R}_{\text{var}(p)}(\mathbf{U}; \mathbf{L}_{\text{diff}}) = \frac{1}{2} \|\mathbf{L}_{\text{diff}} \mathbf{U}\|_p^p, \quad p \in \{1, 2\}, \quad (6)$$

where $p = 1$ yields a TV penalty (promotes piecewise-constant spatial profiles) and $p = 2$ obtains a QV penalty (promotes smoothly varying profiles). Intuitively, the variation regularizer discourages abrupt changes in the latent factors between neighboring locations. Computationally, with $p = 2$ (QV), the update is quadratic and fits neatly into ALS/gradient updates; with $p = 1$ (TV), proximal methods on differences can be used for model solving. These updates are standard within CP/Tucker and scale well as \mathbf{L}_{diff} is highly sparse.

In traffic applications, variation-based spatial regularization has proven effective. For example, the fused-penalty CP model in [Wu et al. \(2018\)](#) applied a TV penalty on spatial latent factors to enforce traffic states at adjacent sensors change gradually, improving the robustness under noise and missingness. Practically, when sensors/locations are not ordered along a line, L_{diff} can be replaced with an oriented incidence matrix of the road graph to generate a graph-TV variant ([Wang et al., 2016](#)). The same construction can be applied to the temporal factor with a temporal difference operator, enforcing short-lag continuity, and we discuss it in Section 3.1.2.

(c) Other local smoothness formulations. When an explicit road network graph is unavailable or insufficient, local spatial coherence can still be imposed using proximity/similarity formulations. Several alternatives have been explored:

- Neighbor regression constraints. Nearby locations are used to predict and regularize latent spatial factors for each site via a local linear regression term within CP, which effectively shrinks factors toward a weighted average of related neighbors ([Zhou et al., 2015](#)).
- Similarity matrix regularization. A precomputed similarity matrix, e.g., from euclidean distance, travel time, or functional similarity, penalizes discrepancies between factors of similar locations, encouraging locally consistent spatial representations ([Yang et al., 2019; Jia et al., 2020; Said and Erradi, 2021](#)).
- Transform domain regularization, e.g., graph-spectral regularizer ([Deng et al., 2021a](#)), hyper-Laplacian spectral regularization ([Li et al., 2024c](#)), spectral total variation ([Takemoto et al., 2025; Yang et al., 2025a](#)). In graph-tensor SVD (GT-SVD) introduced in [Deng et al. \(2021a\)](#), spatial factors are expanded in the Laplacian eigenbasis and constrained to be low-rank in the spectral domain, which captures both local adjacency effects and longer-range correlations such as upstream/downstream influences that may not appear as immediate neighbors in an adjacency graph.

In short, similarity/proximity- based schemes provide flexible ways to encode spatial domain knowledge even without an explicit network: by enforcing consistency among nearby or functionally similar locations. These methods improve fidelity of tensor decompositions for spatial interpolation (kriging), denoising, and robust imputation in traffic data.

(ii) Others spatial strategies In practice, it is often useful to design problem-specific regularization tailored to the dataset and application. There are studies leverage custom spatial similarity measures that encode contextual information other than physical proximity. For instance, [Said and Erradi \(2021\)](#) partitioned a city into regions and constructed urban feature vectors for each region based on non-geometric information such as demographics, land use, and road density. These features were used to derive an urban similarity matrix that captures functional similarities between regions, which was then incorporated into a modified CP decomposition as a spatial regularizer, allowing side information unrelated to road network adjacency to guide the spatial structure of the factorization.

Another line of work introduces structured covariance constraints on the spatial mode. Instead of a Laplacian or difference penalty, the spatial factors are assumed to follow a multivariate distribution with a covariance aligned to spatial relationships. For example, a Gaussian process (GP) prior with a certain kernel function, e.g., Matérn kernels depending on inter-location distance ([Rasmussen and Williams, 2006](#)) or graph kernels derived from the Laplacian ([Smola and Kondor, 2003](#)), can obtain a covariance matrix that encodes how much latent factors of two locations should co-vary ([Lei and Sun, 2024](#)). Penalizing deviations from the specified covariance structure (often via a *covariance norm* regularization or inverse covariance quadratic form), the model can incorporate complex spatial correlations beyond nearest-neighbor links ([Lei et al., 2024](#)). We will elaborate more on covariance-based regularization in the following joint spatiotemporal modeling part, as it provides a unified framework for space-time regularities.

Most decomposition models combine spatial regularization with additional constraints to better capture traffic dynamics and data imperfections. Common complements include temporal regularization to encode dynamics ([Zhou et al., 2015; Gong et al., 2025](#)); robustness penalties to handle noise/outliers ([Wu et al., 2018](#)), e.g., separating sparse anomalies from low-rank patterns ([Yu et al., 2025](#)); and nonnegativity to ensure interpretability for nonnegative measures such as flow ([Yang et al., 2019](#)). Overall, incorporating spatial correlations helps ensure that learned spatial factors reflect real geographic/network patterns rather than being driven solely by algebraic fit.

(2) *Incorporating temporal dependencies.* In addition to spatial correlations, real-world traffic data exhibit rich temporal structure. Patterns include long-term periodicities such as daily circadian rhythms, weekday-weekend cycles, and seasonal trends, as well as short-term dependencies, e.g., autocorrelation across consecutive intervals within a day (Li et al., 2015). For example, in a third-order tensor organized as *location* \times *time of day* \times *day*, the *time of day* factors typically capture diurnal profiles (morning/evening peaks, mid-day lulls), while the *day* mode factors separate weekdays from weekends or reflect gradual week-to-week changes (Lei et al., 2022a). However, standard decomposition models do not enforce such structure; they learn it only from observations. Under noisy or sparse sampling, basic decomposition models often fail to recover smooth daily cycles or clear weekly periodicity (Chen et al., 2021b; Lei et al., 2022b).

To better capture temporal dynamics, especially when data are scarce or noisy, various forms of temporal regularization have been introduced into tensor decomposition frameworks. We group common approaches into four classes:

- (i) **Local smoothness constraints:** penalize abrupt changes between consecutive time steps to enforce short-term continuity.
- (ii) **Dynamic regularization:** encodes time-evolving behavior using time-series models, e.g., AR, vector autoregressive (VAR), state-space, providing directed temporal dependence and supporting forecasting.
- (iii) **Structured Hankel embedding constraints:** exploit longer-term or seasonal patterns by forming Hankel (delay) embeddings of time series and imposing low-rank structure in the embedded domain.
- (iv) **Other constraints:** hybrid or domain-informed designs, e.g., frequency-domain/spectral regularization, entropy-based features, that incorporate known temporal priors.

In what follows, we review representative tensor decomposition methods within each category and how they enhance temporal modeling for spatiotemporal traffic data analysis. As defined, we use $\mathbf{V} \in \mathbb{R}^{T \times R}$ to denote the temporal latent factor matrix, with row vectors $\{\mathbf{v}_{t:} \in \mathbb{R}^R\}_{t=1}^T$ and column vectors $\{\mathbf{v}_{:r} \in \mathbb{R}^T\}_{r=1}^R$, and $\mathcal{R}(\mathbf{V})$ to denote the temporal regularizer.

(i) Local smoothness constraints To encourage short-term temporal continuity in low-rank factorization, it is common to regularize the time-mode factor matrix $\mathbf{V} \in \mathbb{R}^{T \times R}$, analogous to spatial smoothness.

Temporal Laplacian smoothness. A straightforward way is to construct a linear path-graph along the time axis, where each time index t ($1 < t < T$) connects to its neighbors $t - 1$ and $t + 1$. With unit edge weights, the unnormalized Laplacian is

$$\mathbf{Lap}_t = \begin{bmatrix} 1 & -1 & \cdots & 0 & 0 \\ -1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 2 & -1 \\ 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \in \mathbb{R}^{T \times T}.$$

The associated quadratic penalty (i.e., temporal Laplacian regularizer)

$$\mathcal{R}_{\text{Lap}}(\mathbf{V}; \mathbf{Lap}_t) = \frac{1}{2} \text{tr}(\mathbf{V}^\top \mathbf{Lap}_t \mathbf{V}) = \frac{1}{2} \sum_{t=1}^{T-1} \|\mathbf{v}_{t:} - \mathbf{v}_{t+1:}\|_2^2 \quad (7)$$

penalizes pairwise differences across graph edges, making adjacent time-factor rows similar and smoothing short-lag fluctuations (Bhanu et al., 2020). It is also possible to encode longer-term or periodic correlations by augmenting the adjacency matrix based on temporal periodicity, e.g., adding a wrap-around edge between the first and the last time points for diurnal cycles or connecting the same time-of-day across days. In this way, the Laplacian functions as an undirected AR-style regularizer (Bhanu et al., 2020; Chen et al., 2024d; Wu et al., 2025).

Toeplitz/variation smoothness. An alternative to Laplacian smoothing is to penalize first-order temporal differences with a $(T - 1) \times T$ Toeplitz operator \mathbf{T}_{diff} whose rows compute $\mathbf{v}_{t:} - \mathbf{v}_{t+1:}$ (Zhou et al., 2015;

[Gong et al., 2025](#)):

$$\mathbf{T}_{\text{diff}} = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix}.$$

This has the same structure as the spatial difference operator \mathbf{L}_{diff} used in SPC ([Yokota et al., 2016b](#)) (cf. Eq. (6)). The associated variation penalty is

$$\mathcal{R}_{\text{var}(p)}(\mathbf{V}; \mathbf{T}_{\text{diff}}) = \frac{1}{2} \|\mathbf{T}_{\text{diff}} \mathbf{V}\|_p^p, \quad p \in \{1, 2\}. \quad (8)$$

With $p = 2$ (quadratic variation, QV), the penalty smooths temporal profiles and is equivalent to the Laplacian penalty on a unit-weight, first-order (nearest-neighbor) path graph, as $\mathbf{T}_{\text{diff}}^\top \mathbf{T}_{\text{diff}} = \mathbf{Lap}_t$. With $p = 1$ (total variation, TV), the term preserves piecewise-constant profiles ([Yokota et al., 2016b](#)) and sharp change points such as onsets/offsets of rush hours ([Wu et al., 2018](#)). Toeplitz-based penalties have been widely used to discourage interval-to-interval jumps in the time mode ([Zhou et al., 2015; Gong et al., 2025](#)).

These local temporal smoothers have been widely used in traffic applications ([Yang et al., 2024; Chen et al., 2024a](#)). For example, in CP models, [Baggag et al. \(2019\)](#) and [Bhanu et al. \(2020\)](#) applied temporal graph-Laplacian regularization, encouraging hour-to-hour or day-to-day latent factors to evolve gradually; [Zhou et al. \(2015\)](#) incorporated a Toeplitz-based difference penalty to enforce continuity in daily traffic speed data; [Wu et al. \(2018\)](#) and [Zhang et al. \(2022\)](#) used TV regularization along the time mode to obtain piecewise-smooth temporal factors that capture steady periods separated by change points, e.g., before and after rush hour. For Tucker decomposition, [Gong et al. \(2025\)](#) introduced temporal smoothness by penalizing the norm of the tensor reconstructed after applying a first-order differencing operator, i.e., the Toeplitz matrix \mathbf{T}_{diff} , along the time mode. This discouraged abrupt changes in the reconstructed tensor over time mode and improve imputation performance especially during periods of rapid transitions such as congestion onset. Other variants include weighted temporal regularizers based on time-of-day similarity ([Jia et al., 2020](#)) and linear regression constraints on recent time steps to capture short-term correlations ([Yan et al., 2021](#)), etc.

While local temporal smoothness effectively captures short-scale temporal dependencies, e.g., ensuring the traffic state at 8:05 a.m. is close to those at 8:00 a.m. and 8:10 a.m., it does not inherently model evolving trends or longer-range patterns. Local smoothing alone is not sufficient for long-range dynamics and forecasting. To handle directed temporal evolution, more structured dynamic modeling is needed. Many low-rank tensor studies pair smoothness with additional mechanisms that account for sequential dependence when predicting forward, such as the work in [Baggag et al. \(2019\)](#) and [Bhanu et al. \(2020\)](#).

(ii) Dynamic regularization To explicitly model time-evolving behavior in traffic data, low-rank factorization models often integrate time-series structure—e.g., AR, VAR, state-space dynamics ([Schmidt and Lopes, 2019; Xiong et al., 2010; Yu et al., 2016](#))—into the decomposition. Beyond local smoothness penalties, dynamic regularizers enforce that temporal factors follow a generative process and can capture directional and long-range dependencies.

A representative example is temporal regularized matrix factorization (TRMF) ([Yu et al., 2016](#)), proposed for time-series imputation and forecasting. Let $\mathcal{L} = \{\ell_1, \dots, \ell_d\}$ be an AR(d) lag set. TRMF assumes that for each $t \notin \mathcal{L}$,

$$\mathbf{v}_{t:} \approx \sum_{k=1}^d \mathbf{A}_k \mathbf{v}_{t-\ell_k:},$$

with $\mathbf{A}_k \in \mathbb{R}^{R \times R}$ the lag- ℓ_k coefficient matrices (learned or pre-specified). The AR regularizer writes:

$$\mathcal{R}_{\text{AR}(d)}(\mathbf{V}; \{\mathbf{A}_k\}_{k=1}^d) = \frac{1}{2} \sum_{t \notin \mathcal{L}} \left\| \mathbf{v}_{t:} - \sum_{k=1}^d \mathbf{A}_k \mathbf{v}_{t-\ell_k:} \right\|_2^2. \quad (9)$$

The encoded AR dynamics on \mathbf{V} enable extrapolation for forecasting. In traffic data analysis, [Chen et al. \(2023a\)](#) adapted AR regularizer to discover recurring temporal patterns, and [Chen et al. \(2025d\)](#) extended it to a VAR regularization for multi-step traffic state prediction, capturing interactions among latent dimensions over time. It is also possible to apply transition matrix on the spatial Laplacian road network matrices to define dynamic Laplacians for real-time traffic prediction, such as the work in [Deng et al. \(2016\)](#).

Within tensor-based models, [Baggag et al. \(2019\)](#) proposed temporal regularized tensor factorization (TRTF) under a CP paradigm. TRTF factorizes a third-order traffic tensor, e.g., *origin* \times *destination* \times *time*, and fits an AR(1) state-space model on the temporal factor matrix. Still using $\mathbf{V} \in \mathbb{R}^{T \times R}$ to denote the temporal factor matrix, TRTF imposes $\mathbf{v}_{t+1} \approx \mathbf{A}\mathbf{v}_t$, where $\mathbf{A} \in \mathbb{R}^{R \times R}$ is a learned transition matrix capturing directional dependencies among latent temporal components. Thus, the latent factor at time $t + 1$ is a linear transform of the factor at time t , enabling simulation and forecasting of traffic conditions by iterating the latent state forward. Empirically, TRTF outperformed standard factorization for traffic prediction under high missing rates, underscoring the benefit of embedding time-series dynamics. Similarly, [Wang and Sun \(2021\)](#) proposed a low-rank model regularized by a VAR process aimed at traffic anomaly detection—by modeling expected temporal evolution, anomalies are flagged as deviations the VAR-regularized low-rank approximation cannot explain.

Overall, dynamic regularization improves low-rank decomposition models by explicitly embedding temporal evolution mechanisms. While local smoothness captures short-term continuity, AR/VAR/state-space penalties encode directionality and long-range dependence, enabling trend/seasonality anticipation and more reliable forecasting or pattern identification under noise and missing data ([Baggag et al., 2019](#); [Wang and Sun, 2021](#); [Liao et al., 2025](#)).

(iii) Hankel structured regularization One limitation of AR/VAR-type temporal regularization is the assumption on relatively low-order linear dynamics and (often) stationarity. Real-world traffic time series are frequently nonstationary and multi-periodic, showing gradual growth/decline trends, multi-scale seasonality, and occasional regime changes that AR models may miss ([Chen et al., 2024f](#)). To uncover such dynamics in a data-driven way, researchers have turned to Hankel (time-delay) embedding techniques within low-rank models ([Yokota et al., 2018](#); [Wang et al., 2023b](#)). The idea is to represent a time series by a Hankel matrix (or high-order Hankel tensor) in which each row/column is a lagged segment of the original series. For a univariate series $\mathbf{y} = [y_1, \dots, y_T]^\top \in \mathbb{R}^T$, a Hankel matrix of window length (embedding dimension) τ is

$$\mathbf{Y}_{\text{Hankel}} = \begin{bmatrix} y_1 & y_2 & \cdots & y_{T-\tau+1} \\ y_2 & y_3 & \cdots & y_{T-\tau+2} \\ \vdots & \vdots & \ddots & \vdots \\ y_\tau & y_{\tau+1} & \cdots & y_T \end{bmatrix},$$

which has size $\tau \times (T - \tau + 1)$. Each row advances the window by one time step. We define this Hankel embedding operator as $\mathcal{H}_\tau(\cdot) : \mathbb{R}^T \rightarrow \mathbb{R}^{\tau \times (T - \tau + 1)}$, and $\mathbf{Y}_{\text{Hankel}} = \mathcal{H}_\tau(\mathbf{y})$. If the original series satisfies a linear recurrence of order $\leq \tau$, the Hankel matrix is low-rank. This means that low-rankness in a Hankel representation reveals inherent temporal cycles or dependencies—a classic result consistent with the fact that any linear time-invariant system of order R yields a Hankel matrix of rank R ([Sarkar et al., 2021](#); [Golyandina et al., 2018](#)).

Applying the idea of Hankel embedded time series modeling, [Yokota et al. \(2018\)](#) showed that performing low-rank CP on a Hankelized tensor, where each mode is constructed by a Hankel embedding, can capture complex dynamics not apparent in the original data format. [Chen et al. \(2024f\)](#) recently proposed Hankel temporal MF (HTMF) for traffic prediction. HTMF factorizes the data as $\mathbf{Y} \approx \mathbf{U}\mathbf{V}^\top$ (spatial and temporal factors), then constructs a Hankel matrix from the temporal factor matrix \mathbf{V} by Hankel embedding each column $\mathbf{v}_{:,r}$ and enforces a low-rank constraint on this Hankel matrix as an additional constraint. For each $\mathbf{v}_{:,r} \in \mathbb{R}^T$ ($r = 1, \dots, R$), let $\mathcal{H}_\tau(\mathbf{v}_{:,r}) \in \mathbb{R}^{\tau \times (T - \tau + 1)}$ denote its Hankel embedded matrix. Stacking $\{\mathcal{H}_\tau(\mathbf{v}_{:,r})\}_{r=1}^R$ along rows to form a embedded matrix $\mathbf{V}_{\text{Hankel}} = [\mathcal{H}_\tau(\mathbf{v}_{:,1}); \dots; \mathcal{H}_\tau(\mathbf{v}_{:,R})] \in \mathbb{R}^{R\tau \times (T - \tau + 1)}$. HTMF enforces a low-rank constraint on $\mathbf{V}_{\text{Hankel}}$: $\text{rank}(\mathbf{V}_{\text{Hankel}}) = R$, with the rank $R \ll T - \tau + 1$. The embedding operation on a matrix is then denoted as $\mathcal{H}_\tau(\cdot) : \mathbb{R}^{T \times R} \rightarrow \mathbb{R}^{R\tau \times (T - \tau + 1)}$, and $\mathbf{V}_{\text{Hankel}} = \mathcal{H}_\tau(\mathbf{V})$. The overall problem can be expressed as:

$$\min_{\mathbf{U}, \mathbf{V}} \frac{1}{2} \left\| (\mathbf{Y} - \mathbf{U}\mathbf{V}^\top) \right\|_F^2, \text{ s.t. } \text{rank}(\mathcal{H}_\tau(\mathbf{V})) = R. \quad (10)$$

We denote the corresponding Hankel regularization as:

$$\mathcal{R}_{\text{Hankel}}(\mathbf{V}; \tau) : \text{rank}(\mathcal{H}_\tau(\mathbf{V})) = R. \quad (11)$$

This encourages the latent temporal factors to exhibit structured low-rank patterns indicative of recurring cycles or seasonality, without assuming a specific period or stationarity a priori. Empirical results show

that HTMF outperformed AR-regularized MF on traffic datasets with strong seasonal trends, since Hankel regularization captures longer-range periodicity that lower-order AR may overlook (Chen et al., 2024f).

Hankel-structured embeddings have also been explored in other factorization frameworks. For example, Shi et al. (2020b) introduced a Tucker-based technique—block Hankel tensor ARIMA (BHT-ARIMA)—which integrates delay embedding and time series modeling. The traffic time series is first converted into a block Hankel tensor via multi-way delay embedding, transforming sequential dynamics into a structured low-rank format. Tucker decomposition is then applied to the Hankel-transformed tensor, followed by fitting an ARIMA model on the core for forecasting. By encoding seasonal/periodic dependencies in a low-rank tensor space, BHT-ARIMA effectively captures latent temporal dynamics and shows strong performance on short time series across road segments. Additionally, Hankel-based ideas have been used in rank-surrogate based completion models (Wang et al., 2021, 2023b), where the low-rank assumption on the embedded tensor is enforced through nuclear norm minimization, improving recovery for high incomplete or irregularly sampled data (see Section 3.2). Overall, Hankel-structured regularization provides a non-parametric, flexible way to model complex temporal patterns in LRTL, often improving upon AR-type regularizers when traffic data are highly nonstationary and multi-periodic.

(iv) Other temporal constraints Beyond the categories above, researchers have explored additional ways to incorporate temporal knowledge into tensor decomposition, often in hybrid or innovative fashions.

One approach is to incorporate frequency-domain and information-theoretic features. For example, Said and Erradi (2021) introduced entropy-based features and fast Fourier transform (FFT) components within a CP model to characterize and regularize temporal dynamics. Their method explicitly captures dominant frequency components such as daily or weekly cycles in the latent temporal factors via the Fourier terms, effectively embedding known periodic behaviors directly into the factorization. This can be seen as a form of spectral regularization, which ensures the latent representation accounts for observed frequency peaks in the traffic time series data.

Another approach involves covariance-based temporal regularization, analogous to the spatial covariance strategies mentioned earlier. Lei and Sun (2024) proposed using a Matérn covariance function to define a prior covariance matrix for the temporal factors, thereby encouraging the latent time profiles to follow a certain smoothness and correlation structure in time. This method falls under a general framework of using kernel functions to impose structured priors on factor matrices. We will discuss a unified view of such covariance norm regularization in the joint spatiotemporal section.

There are also methods that combine external dynamic modules or machine learning components into the tensor decomposition framework. For example, Zhang et al. (2022) integrated dynamic mode decomposition (DMD)—a technique from dynamic systems that identifies spatiotemporal modes of evolution—with CP decomposition. They applied CP with temporal TV smoothing and performed DMD to ensure the extracted temporal patterns are consistent with linear dynamic modes, which improved forecasting performance of traffic speed data. On a different front, Bhanu et al. (2020) combined CP decomposition with components from neural networks, effectively creating a hybrid model where the tensor factorization captures global patterns and a neural network captures more complex nonlinear temporal behavior. This allowed their model to handle both imputation and prediction tasks, benefiting from the expressiveness of neural nets while retaining the structure of a low-rank decomposition. Such approaches go beyond explicit regularization terms, instead they learn temporal dynamics through additional model structure or modules.

Lastly, an effective yet conceptually simple strategy for temporal modeling is to reformat the data tensor to encode known periodic structures. Rather than treating time as one mode, one can unfold it into multiple modes that correspond to different granularities of time. For instance, Tan et al. (2013) constructed a fourth-order tensor with modes *location* \times *time of day* \times *day of week* \times *week of month* for traffic flow data, then applied Tucker decomposition. In this setup, daily and weekly periodicities are explicitly separated into different modes. Specifically, the factor matrix for *time of day* captures the diurnal cycle, while the factor for *day of week* captures weekday/weekend patterns. Such approach effectively builds the periodicity into the model without requiring an extra regularizer to enforce it, and can improve modeling of recurrent patterns.

(3) Joint spatiotemporal modeling. Real-world traffic phenomena intertwine spatial and temporal dependencies, therefore it is often beneficial to model these dimensions jointly rather than in isolation (Shekhar et al., 2015). In tensor decomposition, joint spatiotemporal modeling means enforcing structure or regularization across both spatial and temporal modes in an integrated manner. This encourages the learned latent factors to capture coupled space-time dynamics—e.g., congestion waves that propagate through both space and

time (Lei et al., 2022b)—instead of treating spatial and temporal patterns separately. A unified view also helps interpretation, such as region-specific rush hours (defined by a conjunction of location and time) or how a local traffic incident influences downstream traffic over the next hour via a coupled space-time effect (Kargas et al., 2021; Saberi et al., 2020; Wang et al., 2024b).

A straightforward route to joint modeling is to include both spatial and temporal regularizers in the same objective: penalize the spatial factor matrix with a spatial smoothness term and the temporal factor matrix with a temporal smoothness/dynamic term, simultaneously during factorization. Many works follow this approach—regularizing each mode independently but within one optimization problem—thereby each mode respects its own structure while the global low-rank model ties them together. More recent research explored more sophisticated couplings that go beyond independent penalties.

We group joint spatiotemporal modeling strategies into three categories:

- (i) **Combined space and time regularization:** apply spatial and temporal regularizers together within a single model to enforce consistency along both dimensions.
- (ii) **Covariance-based regularization:** use structured covariance matrices that simultaneously encode spatial and temporal correlations, providing a unified regularization framework across modes.
- (iii) **Complementary constraints** (i.e., complementary residual modeling): impose spatiotemporal structure not only on the latent factors but also on residual/error terms to capture fine-scale patterns that the low-rank factors might miss, improving robustness and generalization.

These approaches offer varying degrees of integration between spatial and temporal modeling, and have been shown to improve imputation and forecasting performance relative to using spatial or temporal information alone.

(i) Combine space and time Many studies directly incorporate both spatial and temporal regularization in tandem to build a joint model. In low-rank decomposition frameworks for multidimensional data, representative models include the graph Laplacian regularized matrix factorization (MF) by Rao et al. (2015), for which the optimization problem for an incomplete spatiotemporal matrix $\mathbf{Y} \in \mathbb{R}^{M \times T}$ (Ω denoting set of observed indices) is formulated as:

$$\min_{\mathbf{U}, \mathbf{V}} \frac{1}{2} \left\| (\mathbf{Y} - \mathbf{U}\mathbf{V}^\top)_{\Omega} \right\|_F^2 + \frac{\rho_s}{2} \text{tr}(\mathbf{U}^\top \mathbf{L}_U \mathbf{U}) + \frac{\rho_t}{2} \text{tr}(\mathbf{V}^\top \mathbf{L}_V \mathbf{V}), \quad (12)$$

where $\mathbf{L}_U = \mathbf{Lap}_s + \eta_s \mathbf{I}_M$ and $\mathbf{L}_V = \mathbf{Lap}_t + \eta_t \mathbf{I}_T$, with $\{\mathbf{Lap}_s, \mathbf{Lap}_t\}$ being the Laplacian matrices for the respective dimensions and $\{\eta_s, \eta_t\}$ regularization weights. In a tenor setting, a typical model is the SPC by Yokota et al. (2016b), for which the objective function of the constrained CP decomposition on a D th-order tensor $\mathbf{Y} \in \mathbb{R}^{I_1 \times \dots \times I_D}$ can be written as:

$$\min_{\boldsymbol{\lambda}, \{\mathbf{A}^{(d)}\}_{d=1}^D} \frac{1}{2} \|\mathbf{Y}_{\Omega} - \mathbf{M}_{\Omega}\|_F^2 + \sum_{r=1}^R \frac{\lambda_r}{2} \sum_{d=1}^D \rho_d \left\| \mathbf{L}_{\text{diff}}^{(d)} \mathbf{a}_r^{(d)} \right\|_p^p, \quad (13)$$

where $\mathbf{M} = \sum_{r=1}^R \lambda_r \mathbf{a}_r^{(1)} \circ \dots \circ \mathbf{a}_r^{(D)}$ and for each mode d , $\mathbf{L}_{\text{diff}}^{(d)} \in \mathbb{R}^{(I_d-1) \times I_d}$ is a difference operator defined as before, ρ_d is the smoothness parameter, and $p \in \{1, 2\}$.

As for traffic data analysis, Zhou et al. (2015) in their CP decomposition used a combination of spatial linear regression constraints (to enforce that nearby locations have similar factors) and temporal Toeplitz smoothness constraints (to enforce day-to-day continuity) in the latent space. Wu et al. (2018) applied TV (total variation) penalties on each mode of a three-way decomposition (so both the spatial mode and the temporal mode were penalized for piecewise smoothness), in addition to a graph Laplacian penalty on the spatial factors—ensuring simultaneously that in latent space the spatial and temporal patterns are smooth. Bhanu et al. (2020) likewise adopted graph Laplacian regularizers for both spatial and temporal factor matrices in a CP model, effectively coupling the neighborhood consistency across space with the continuity across time. Gong et al. (2025) proposed a spatiotemporal-regularized Tucker model that combined a manifold smoothness prior on the spatial factor \mathbf{U} (through a graph Laplacian, as described in Section 3.1.2) and a first-order differencing prior on the temporal factor \mathbf{V} (as in Section 3.1.2). In their ablation study, removing either the spatial or the temporal regularizer led to worse imputation performance under high missing data rates, confirming that regularizations in both dimension were crucial to accurately capture

traffic dynamics. These examples illustrate the straightforward approach of applying separate spatial and temporal penalties within one model—by doing so, the decomposition is informed by network topology and temporal continuity at once, leading to latent factors that reflect patterns such as “nearby sensors at nearby times have similar traffic states” (Xie et al., 2024).

Some methods go a step further by explicitly coupling spatial and temporal effects instead of adding their penalties together. For example, in the TRTF (temporal regularized tensor factorization) model mentioned earlier (Baggag et al., 2019), the spatial and temporal regularizations (AR dynamics on time, graph smoothness on space) interact through the shared latent tensor. The AR regularizer effectively models how traffic conditions propagate temporally through the spatial network—e.g., the effect of an incident moves along connected roads over time—which is a joint space-time phenomenon. Another example is the work by Zhou et al. (2019), who developed a coupled matrix-tensor factorization to fuse traffic data at different temporal resolutions. They linked a daily-scale traffic matrix with an hourly-scale traffic tensor via shared spatial factors, meaning the spatial patterns are common but the temporal patterns are at different granularities. This coupling ensured that information flows between the two temporal scales (the daily and hourly patterns inform each other through the spatial coupling), which is a practical form of spatiotemporal integration for multi-resolution data. In addition, domain knowledge can motivate designing latent factors that inherently capture specific spatiotemporal behaviors. For instance, Kargas et al. (2021) introduced a model that incorporated epidemiological SIR dynamics into a tensor factorization (originally for disease spread modeling). Analogously, the principle can be translated to traffic modeling by using traffic flow equations or diffusion models as a regularization on the latent factors—for example, requiring that the evolution of the latent traffic state follows a physical diffusion process across the network. This kind of physics-informed regularization would tightly couple space and time by enforcing known mechanistic relationships. Although it is not yet common in traffic tensor literature, it represents an interesting future direction as hinted by Kargas et al. (2021).

In general, joint spatiotemporal models help overcome limitations of treating space and time separately. A purely spatially-regularized model might over-smooth across locations and fail to capture sharp, localized disturbances, e.g., a sudden jam on one road. A purely temporally-regularized model might smooth out or ignore spatial heterogeneity, e.g., differences between neighborhoods or routes. By enforcing both spatial and temporal consistency, joint models achieve a balance: for example, if an entire region of sensors has missing data for a period, a joint model can fill it in by simultaneously leveraging data from other regions at the same time (spatial neighbors) and data from other times at the same region (temporal neighbors). Empirically, this often yields more accurate and realistic imputations. In anomaly detection, a joint approach can distinguish true anomalies (which might be isolated in both space and time) from noise, by expecting a smooth background in both dimensions (Sofuoğlu and Aviyente, 2022). Although considering both space and time increases model complexity and adds hyperparameters, the payoff is a higher-fidelity understanding of traffic phenomena. Many works report that the combination leads to significant improvements in tasks such as data reconstruction and prediction, especially under challenging conditions—e.g., high missing rates, non-recurrent events—where one dimension alone might not provide enough information. By incorporating road network structure and temporal dynamics into one unified low-rank decomposition framework, joint spatiotemporal tensor models can capture patterns such as congestion propagation or peak traffic waves much more effectively than models that treat space or time in isolation.

(ii) Covariance based regularization Beyond specified smoothness, a more general framework for spatiotemporal regularization is to use structured covariance matrices to encode prior knowledge about dependencies among spatial locations and time points and incorporate them into the latent factorization process (Zhou et al., 2012; Lei et al., 2022b; Lei and Sun, 2024). In this formulation, one assumes that the columns of a factor matrix are drawn from a multivariate Gaussian distribution with a certain covariance function. By choosing an appropriate covariance matrix, we can simultaneously encode complex spatial and temporal correlations. For example, a Gaussian process (GP) prior on the temporal factor matrix $\mathbf{V} \in \mathbb{R}^{T \times R}$ might assume that each column of \mathbf{V} (which corresponds to evolution over time of a temporal latent component) is an T -dimensional Gaussian with covariance matrix $\mathbf{K}_t \in \mathbb{R}^{T \times T}$; if \mathbf{K}_t is constructed using, for example, a squared-exponential (SE) kernel, this implies a smoothness and perhaps periodicity in time (depending on the kernel parameters). Similarly, a GP prior on the spatial factor matrix $\mathbf{U} \in \mathbb{R}^{M \times R}$ with covariance $\mathbf{K}_s \in \mathbb{R}^{M \times M}$ could enforce that each column of \mathbf{U} (a latent pattern across locations) has correlations defined by \mathbf{K}_s (which might be derived from physical distance or connectivity). This is a probabilistic regularization view: instead of a hard penalty, we impose a soft constraint that factors should lie in a function space defined by those kernels (Rasmussen and Williams, 2006; Zhou et al., 2012). The result is that the

optimization will favor factor matrices whose structure (covariance) matches the prescribed ones, thereby unifying many types of regularization in a single framework.

A notable example is the kernelized probabilistic MF (KPMF) model by Zhou et al. (2012). In KPMF, the authors assume that the factor matrix columns are random draws from a Gaussian with a given covariance/kernel matrix. In matrix completion terms, if we have a spatiotemporal data matrix (with locations as rows and times as columns), one can place a kernel-based prior on rows and another on columns. This effectively integrates side information: e.g., using a spatial kernel for rows and a temporal kernel for columns means the model “knows” which locations are similar and which time points are similar a priori, and it uses that in factorization. The optimization problem in KPMF on a spatiotemporal matrix $\mathbf{Y} \in \mathbb{R}^{M \times T}$ with Ω being observed index set can be formed as:

$$\min_{\mathbf{U}, \mathbf{V}} \frac{1}{2} \left\| (\mathbf{Y} - \mathbf{U}\mathbf{V}^\top)_{\Omega} \right\|_F^2 + \frac{\rho_s}{2} \sum_{r=1}^R \mathbf{u}_{:r}^\top \mathbf{K}_s^{-1} \mathbf{u}_{:r} + \frac{\rho_t}{2} \sum_{r=1}^R \mathbf{v}_{:r}^\top \mathbf{K}_t^{-1} \mathbf{v}_{:r}, \quad (14)$$

where $\mathbf{K}_s \in \mathbb{R}^{M \times M}$ and $\mathbf{K}_t \in \mathbb{R}^{T \times T}$ denote the covariance matrices for factors \mathbf{U} and \mathbf{V} respectively, and ρ_s and ρ_t are the weight parameters. When extended to a tensor, one can use multiple covariance priors—one for each mode (Lei and Sun, 2024). Lei et al. (2022b, 2024) take this further in a Bayesian tensor factorization context, where they place GP priors on both spatial and temporal factor matrices. By selecting covariance functions (such as Matérn kernels) that capture desired smoothness or periodicity for space and time, they embed those properties into the model. These Bayesian models not only enforce the regularization but also provide uncertainty quantification (UQ) for the estimated factors and missing entries. We will discuss more about this in Section 4.

In a deterministic setting, a similar effect can be achieved by adding a covariance norm penalty. This means if we want the factor \mathbf{U} to have covariance approximately \mathbf{K}_s , we could penalize deviations, e.g., $\left\| \mathbf{U}^\top \mathbf{U} - \mathbf{K}_s \right\|_F^2$ or use $\mathbf{U}^\top \mathbf{K}_s^{-1} \mathbf{U}$ in the objective (Allen et al., 2014; Larsen et al., 2024). Allen et al. (2014) proposed a generalized least squares matrix decomposition (GLS-MD), which integrates known covariance structures into matrix decomposition through a GLS loss formulation. In low-rank models for traffic data, such ideas have emerged as well. Lei and Sun (2024) introduced an additive decomposition where one component of the model is globally low-rank and constrained by covariance norm to capture broad trends with structured correlations, while another component is designed to capture local variations. We will further discuss this approach in the following part on residual modeling.

The key point here is that covariance-based regularization provides a unifying umbrella: many commonly used regularizers can be interpreted as special cases of assuming a specific covariance structure. For instance, a graph Laplacian smoothness penalty is related to assuming the factor matrix has a precision matrix (i.e., the inverse of the covariance matrix) proportional to the Laplacian, and a TV (total variation) penalty corresponds to assuming a certain banded covariance structure. Lei et al. (2022b) explicitly note that both graph-based and AR-based regularizers can be seen as particular instances of the general covariance-based framework. By directly modeling covariances, one can flexibly encode not only local smoothness but also long-range dependencies and periodic structures in a principled manner. Moreover, this approach naturally connects to Bayesian interpretations. For example, a regularization term of the form $\text{tr}(\mathbf{U}^\top \mathbf{L}_u \mathbf{U})$ can be viewed as imposing a Gaussian prior on \mathbf{U} with precision matrix \mathbf{L}_u .

In summary, covariance-driven regularization enhances low-rank models by allowing them to better reflect realistic spatial-temporal dependency structures in traffic data. It often leads to improved completion/imputation performance and interpretability. This framework also lays the foundation for fully probabilistic models that combine the strengths of tensor factorization with GP modeling, which we discuss in detail in Section 4.

(iii) Complementary constraints (residual regularization) While factor-based regularization captures global spatial and temporal patterns, complex traffic data may still contain localized, high-frequency variations or anomalies that a single low-rank model (even with smoothness constraints) might not represent well (Lei and Sun, 2024). Trying to fit these fine-scale details by increasing the rank could overfit and exceed the computational budget (Lei and Sun, 2024). To address the limitation of single regularized low-rank models, recent studies have explored the idea of complementary residual regularization, where structured constraints are not only placed on the latent factors but also on the residuals or reconstruction errors. These residual terms are often used to model localized deviations from the global low-rank structure, enabling

the model to better capture fine-grained patterns while preserving computational efficiency and interpretability (Lei and Sun, 2024; Lei et al., 2022a).

A representative work of this idea is the generalized least squares kernelized tensor factorization (GLSKF) model proposed by Lei and Sun (2024). GLSKF combines a GP covariance regularized CP decomposition for global correlations with a local residual process that is also constrained by a covariance structure. They define the covariance norm for regularization, and formally for a vector $\mathbf{x} \in \mathbb{R}^M$ and a given covariance matrix $\mathbf{K}_x \in \mathbb{R}^{M \times M}$, the covariance norm on \mathbf{x} is defined as:

$$\|\mathbf{x}\|_{\mathbf{K}_x} := \sqrt{\mathbf{x}^\top \mathbf{K}_x^{-1} \mathbf{x}}. \quad (15)$$

The covariance norm regularization on \mathbf{x} is

$$\mathcal{R}_{\text{cov}}(\mathbf{x}; \mathbf{K}_x) = \frac{1}{2} \|\mathbf{x}\|_{\mathbf{K}_x}^2 = \frac{1}{2} \mathbf{x}^\top \mathbf{K}_x^{-1} \mathbf{x}. \quad (16)$$

Consider a third order tensor $\mathbf{Y} \in \mathbb{R}^{M \times T \times P}$ with Ω denoting the observed indices, the optimization problem in GLSKF can be written as:

$$\begin{aligned} & \min_{\mathbf{U}, \mathbf{V}, \mathbf{W}, \mathbf{R}} \frac{1}{2} \left\| \mathbf{Y}_\Omega - \left(\sum_{r=1}^R \mathbf{u}_{:,r} \circ \mathbf{v}_{:,r} \circ \mathbf{w}_{:,r} \right)_\Omega - \mathbf{R}_\Omega \right\|_F^2 + \\ & \quad \frac{\rho_u}{2} \sum_{r=1}^R \|\mathbf{u}_{:,r}\|_{\mathbf{K}_r^u}^2 + \frac{\rho_v}{2} \sum_{r=1}^R \|\mathbf{v}_{:,r}\|_{\mathbf{K}_r^v}^2 + \frac{\rho_w}{2} \sum_{r=1}^R \|\mathbf{w}_{:,r}\|_{\mathbf{K}_r^w}^2 + \frac{\rho_r}{2} \|\text{vec}(\mathbf{R})\|_{\mathbf{K}_{\text{vec}(\mathbf{R})}}^2, \end{aligned} \quad (17)$$

where $\{\mathbf{K}_r^u \in \mathbb{R}^{M \times M}\}_{r=1}^R$, $\{\mathbf{K}_r^v \in \mathbb{R}^{T \times T}\}_{r=1}^R$, and $\{\mathbf{K}_r^w \in \mathbb{R}^{P \times P}\}_{r=1}^R$ are covariance matrices for columns of \mathbf{U} , \mathbf{V} , and \mathbf{W} , respectively, $\mathbf{R} \in \mathbb{R}^{M \times T \times P}$ is the tensor for residual process, for which a product kernel is assumed and thus obtaining a Kronecker covariance matrix $\mathbf{K}_{\text{vec}(\mathbf{R})} \in \mathbb{R}^{MTP \times MTP} = \mathbf{K}_3^{\text{local}} \otimes \mathbf{K}_2^{\text{local}} \otimes \mathbf{K}_1^{\text{local}}$. For simplicity, one can set columns of latent factors follow the same covariance function, e.g., $\mathbf{K}_1^u = \dots = \mathbf{K}_R^u$. For computation efficiency, compactly supported sparse covariance matrices are assumed for the local covariance structures $\{\mathbf{K}_1^{\text{local}} \in \mathbb{R}^{M \times M}, \mathbf{K}_2^{\text{local}} \in \mathbb{R}^{T \times T}, \mathbf{K}_3^{\text{local}} \in \mathbb{R}^{P \times P}\}$, which makes the model to be both efficient and effective for multidimensional data completion. From the model objective, we see that a residual regularization is introduced using generalized least squares (GLS), which penalizes the discrepancy between the observed data and the low-rank approximation under a structured covariance model. This allows the residual to capture local variations not modeled by the latent factors, improving imputation quality, particularly in high-resolution traffic data. Intuitively, this means any remaining discrepancy between the data and the low-rank approximation should be for example a plausible high-frequency variation, e.g., a localized spike that has a certain covariance signature, rather than arbitrary noise. By doing so, the model doubles up: the low-rank part handles the big picture, and the residual part with regularization handles the nuances.

Empirically, this complementary approach proved quite effective. In high-resolution traffic sensor data, there are often small-scale fluctuations due to incidents etc. that a smooth low-rank model might gloss over. GLSKF was able to capture those via the residual component, resulting in improved imputation accuracy compared to standard low-rank models, including the smooth CP model SPC (Yokota et al., 2016b), as demonstrated in experiments (Lei and Sun, 2024). The residual regularization ensured that adding this extra component did not simply overfit noise, but targeted meaningful local structure. This idea is related to the concept of “low-rank plus sparse” decomposition, e.g., separating anomalies as a sparse matrix from a low-rank background, but here the “sparse” part is given a smoothness GP prior too, acknowledging that even anomalies or high-frequency parts have spatial-temporal structure. As traffic datasets become larger and more granular, such multi-component models—global low-rank with local sparse spatiotemporal process—provide a promising way to maintain interpretability and efficiency while not sacrificing fine detail. The Bayesian formulation of this framework has also been introduced in the Bayesian complementary kernelized learning (BCKL) model (Lei et al., 2022a), which will be discussed in Section 4.1.2.

(4) *Other aspects.* In addition to spatial and temporal correlations, a variety of other structural regularizations and model extensions have been explored in low-rank tensor decomposition to better suit traffic data characteristics (Wu et al., 2018; Mukai et al., 2025). We highlight a few important aspects:

- **Orthogonality for interpretability:** To enhance factor interpretability and reduce redundancy, some works impose orthonormality on the latent factors (Sørensen et al., 2012). For example, Afshar et al.

(2017) proposed CP-ORT (orthogonal CP decomposition) for spatiotemporal data, which yields mutually orthogonal components, facilitating interpretation of each factor as an independent pattern. Similarly, Han et al. (2024) introduced an orthogonality-regularized CP model for taxi trip data, showing that enforcing orthogonality on spatial and temporal factors leads to clearer separation of dynamic patterns such as distinct travel hotspots and time-of-day trends, and improved pattern discovery. Imposing orthogonality on latent factor matrices (either as a hard constraint or via a regularizer encouraging $A^{(d)\top} A^{(d)} \approx I_R$) helps address the non-uniqueness of tensor decompositions and aligns the model structure with principles similar to principal component analysis (PCA) (Afshar et al., 2018; Yin et al., 2020).

- **Nonnegativity constraints:** In many traffic applications, the data are counts or non-negative measurements (e.g., traffic volumes, occupancy, travel times), for which negative values are not physically meaningful (Wu and Mi, 2025). Constraining factor matrices to be nonnegative improves interpretability (Sharma et al., 2025; Luo et al., 2019). For instance, Yang et al. (2019) employed nonnegative CP to cluster traffic flow patterns, ensuring each latent component represented a nonnegative intensity profile (e.g., a typical daily flow curve) that contributes positively to the observed data. Jin et al. (2024) further combined nonnegativity with sparsity in a CP model to analyze ride-hailing demand, enabling discovery of distinct, localized patterns—such as demand hotspots that appear only at specific times and never fall below zero. Nonnegativity can be enforced via constrained optimization (hard constraints) or adding penalties that discourage negative values, and it often aligns the factorization with human-interpretable features, since many traffic phenomena, e.g., morning or evening peaks, naturally manifest as nonnegative profiles.
- **Combined regularizer frameworks:** Traffic data are noisy, bursty, and heterogeneous, thus often multiple regularizers are applied simultaneously to capture different data characteristics (Mukai et al., 2025). Two widely used regularization terms are the ℓ_2 norm for smoothness and the ℓ_1 norm for sparsity. As mentioned earlier, Wu et al. (2018) presented a fused CP model that integrated several penalties on the factors: an ℓ_0/ℓ_1 term to promote sparsity and isolate anomalies or rare events; a TV (total variation) term for piecewise-constant behavior; an ℓ_2 (Frobenius) term for overall smoothness; and a graph Laplacian term for spatial manifold structure. This multi-regularizer approach allowed the model to detect sparse anomalies (via sparsity constraints), fit smooth underlying trends (via ℓ_2 and TV terms), and maintain spatial consistency across neighboring sensors (via the graph regularization). As a result, the model achieved a more robust decomposition than single-regularizer baselines, especially with noise and outliers.
- **Hybrid with neural networks:** Recent work embeds small neural networks or continuous functions into the factorization (Luo et al., 2022a; Xu et al., 2025b; Nie et al., 2025). For example, in tensor function factorization, each factor is generated by a learnable function of coordinates such as sensor location and time (Luo et al., 2023). These hybrids can preserve a low-rank skeleton while capturing localized nonlinearities, and combine naturally with classic regularizers e.g., Laplacian, ℓ_1/ℓ_2 , for stability and interpretability (Li et al., 2025b; Luo et al., 2025b, 2024). In addition, some methods attach lightweight neural components around a low-rank core, e.g., decoding residuals, modeling nonlinear interactions, or refining temporal dynamics. Such hybrids typically improve reconstruction/forecasting while preserving the inductive bias of low-rank (Peng et al., 2025a; Jin et al., 2025; Kim et al., 2024; Luo et al., 2024; Xu et al., 2024a)
- **Scalability considerations:** Traffic tensors can be extremely large, e.g., a city-wide traffic speed tensor may include hundreds of locations by thousands of time points, and possibly extra modes for covariates or contextual features (Song et al., 2019). Practical algorithms therefore need to scale in memory and time (Zhou et al., 2016; Phipps et al., 2023). One extension is the weighted/missing-data-aware factorization, which avoids unnecessary computation over unobserved entries and focuses work on Ω (the observed set) (Lei and Sun, 2024). A notable example is CP-WOPT (CP weighted optimization) (Acar et al., 2011), which performs weighted CP by assigning zero weights to missing entries and non-zero weights to observed ones, and solves the problem with efficient gradient-based optimization. CP-WOPT has been shown to scale to tensors with millions of entries—e.g., a $1000 \times 24 \times 365$ traffic tensor (*locations* \times *hours* \times *days*)—making it suitable for large-scale or near-real-time deployment. Additional scalable tactics include stochastic optimization, e.g., mini-batch or block-wise updates (Kolda and Hong, 2020; Yu and Li, 2024), distributed/parallel factorization (Kang et al., 2012; Choi and Vishwanathan, 2014; Smith et al., 2015), and structure-exploiting transforms. For example, methods based

on the tubal Fourier transform reduce large 3D tensor problems to smaller matrix problems via fast Fourier transforms (FFTs) (Chen et al., 2021a), as will be mentioned in Section 3.2.2.

These aspects reflect the evolving landscape of tensor-based modeling for transportation systems. Different forms of regularization serve different purposes: **orthogonality** improves identifiability and yields more interpretable components; **nonnegativity** aligns the model with physical realities of traffic counts and supports parts-based interpretations; **sparsity** and **robust** penalties improve resilience to outliers and noise in heterogeneous traffic environments; and **weighted** or **efficient** algorithms ensure computational scalability as data volumes increase. By thoughtfully selecting and combining such constraints and techniques, researchers can tailor low-rank tensor decomposition models to match the structural characteristics of their traffic data and meet application-specific goals—whether better interpretability, higher accuracy under missing data, or faster computation—thereby extending analytical capabilities beyond those of standard low-rank approximations.

(5) *Summary.* We summarize the introduced spatiotemporal modeling approaches for tensor decomposition in Figure 6, in which the commonly applied tasks have also been included. A comparison of representative tensor decomposition studies for traffic data—including CP, Tucker, and other variants—is provided in Table 3, in chronological order. As can be seen:

- CP is widely/generally paired with spatiotemporal regularizers, e.g., graph Laplacian/variation on space and difference/AR terms on time, as the CP paradigm is simple yet effective. Tucker offers more flexibility but also complexity given a core tensor, and works use unconstrained Tucker can perform well on simple tasks. Recently, many Tucker studies also add spatial/temporal or other forms of regularization such as orthogonality to curb non-uniqueness and improve performance.
- Decomposition models are not only used for imputation/completion and prediction; they are also strong tools for pattern discovery. The learned interpretable latent space/factors help explain underlying spatiotemporal structure, e.g., diurnal/weekly temporal cycles, region-specific spatial patterns, and support downstream tasks such as clustering, regression, anomaly identification/monitoring, and visualization, etc.
- Methodological evolution: There is a clear shift toward richer formats and hybrids. Recent traffic studies increasingly explore tensor ring (TR), tensor train (TT), dimension-preserved decompositions (DPD), and other structured variants, and it becomes more and more common to combine decomposition with complementary modules, e.g., transform spectral domain features, correlated/sparse residuals, and neural transform/networks. These trends typically target higher robustness/ efficiency/ effectiveness under noise/missingness, stronger expressiveness for complex dynamics and better scalability.

3.2. Rank-surrogate relaxations for spatiotemporal traffic data

Rank-surrogate relaxation constitutes another major family of deterministic LRTL methods for estimating unobserved or missing entries in incomplete tensors by exploiting inherent low-rank structure (Gandy et al., 2011; Zhang and Aeron, 2016; Long et al., 2019). Extending matrix-completion principles (Candes and Recht, 2012) to higher-order, multi-way data, these approaches avoid fixing the rank a priori and instead enforce global low-rankness by optimizing directly over the target tensor under surrogate penalties on rank. Typical choices include convex relaxations—most notably tensor nuclear norm minimization (Liu et al., 2012; Lu et al., 2019)—as well as nonconvex surrogates (Hu et al., 2012). For spatiotemporal traffic tensors, which are often sparse and irregular due to sensor outages or communication errors, such relaxations have emerged as a powerful tool for robust data imputation and reconstruction (Chen et al., 2020, 2021b; Nie et al., 2022, 2023). In what follows, we review representative rank-surrogate models and their applications in traffic data analysis.

3.2.1. Rank-surrogate models

Formulation. Let \mathcal{Y} denote an incomplete D th-order observed data tensor with Ω being the set of observed indices. Low-rank tensor completion aims to recover the missing entries in \mathcal{Y} by finding a completion that agrees with the observed data \mathcal{Y}_Ω with the smallest possible rank (Liu et al., 2012). The problem parallels matrix rank minimization for matrix completion: we seek a tensor \mathcal{X} that matches \mathcal{Y}_Ω and is as low-rank as possible. Exact rank minimization is, however, nonconvex and NP-hard (Liu et al., 2012). In addition, the

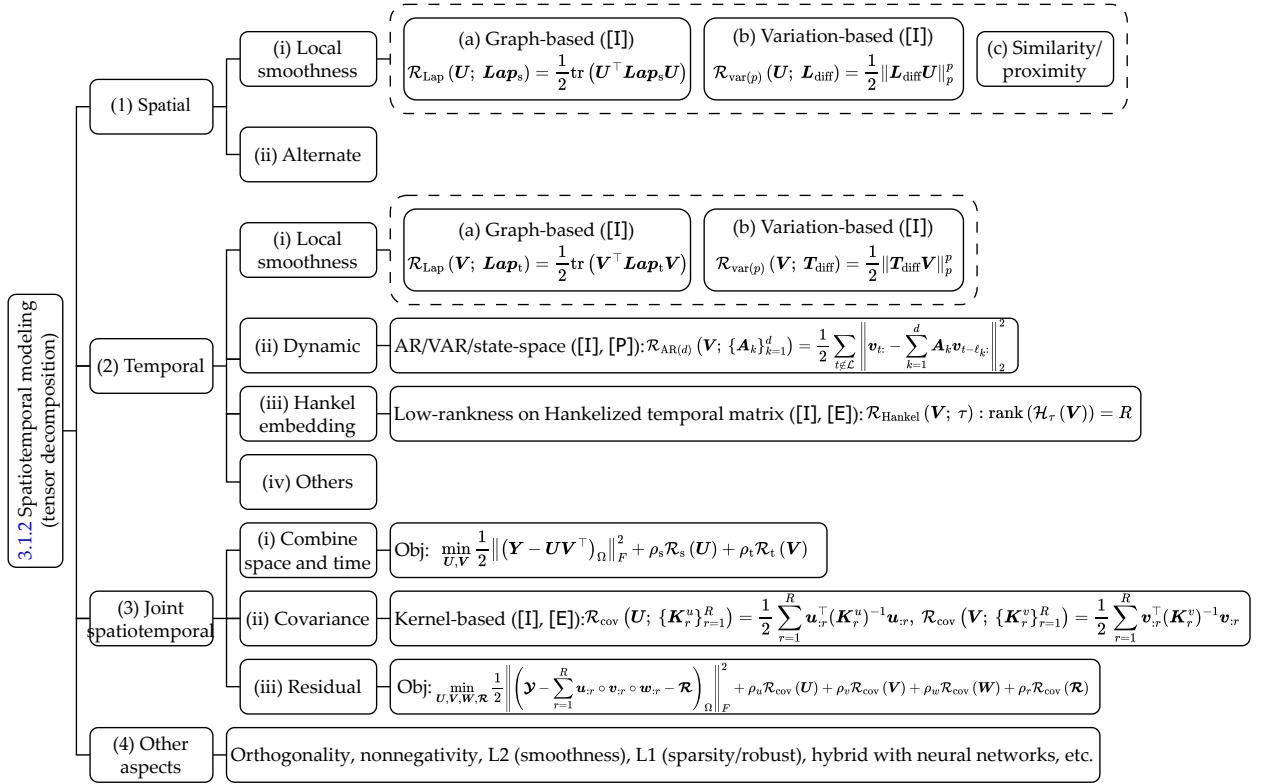


Figure 6: Summary of spatiotemporal modeling strategies/regularization in tensor decomposition (Section 3.1.2).

varying definitions of tensor rank, e.g., CP rank/Tucker multilinear rank and tubal rank, lead to different algorithms and guarantees for tensor completion (Liu et al., 2019b; Bahadori et al., 2014; Qiu et al., 2024; He et al., 2024; Zhang and Aeron, 2016; Zhao et al., 2021). In practice, rank-surrogate models replace the intractable rank objective with tractable proxies (convex or nonconvex penalties), while enforcing agreement with the observed entries (Liu et al., 2012; Chen et al., 2020; Qiu et al., 2024).

Despite the differing rank definitions, rank-surrogate methods follow the same core assumption: the fully observed spatiotemporal tensor lies on a low-dimensional manifold, i.e., it has far fewer degrees of freedom than its size (Liu et al., 2012; Chen et al., 2020). Based on this principle, rank-surrogate tensor completion imposes spatiotemporal consistency by uncovering a global low-dimensional structure that coherently couples all dimensions. This framework is powerful in traffic data analysis, where missing values might occur across entire time periods or spatial regions. Low-rank models can still recover these missing entries by borrowing information from correlated times and locations (Ran et al., 2016b; Nie et al., 2023). Over the past decade, rank-surrogate minimization has become a widely adopted approach for traffic data imputation and analysis, frequently outperforming traditional interpolation and time-series methods by utilizing the multi-way correlations inherent in spatiotemporal traffic data (Chen et al., 2020, 2021b, 2025b).

Computation. Solving the rank minimization problem is generally intractable due to the nonconvex nature of tensor rank functions, thus surrogate objectives are often used. A common approach is convex relaxation, where the tensor rank is replaced by a convex function that promotes low-rank solutions—typically the **nuclear norm** (Liu et al., 2012; Lu et al., 2019; Jiang et al., 2020b). For a matrix $\mathbf{X} \in \mathbb{R}^{M \times T}$, the nuclear norm is defined as:

$$\|\mathbf{X}\|_* = \sum_{i=1}^{\min(M,T)} \sigma_i(\mathbf{X}), \quad (18)$$

where $\sigma_i(\mathbf{X})$ is the i th singular value of \mathbf{X} in decreasing order. The nuclear norm is the convex envelope (i.e., the tightest convex lower bound) of the rank function (Recht et al., 2010). Liu et al. (2012) pioneered nuclear norm for tensors with their high-accuracy low-rank tensor completion (HaLRTC) method. They formulated a convex optimization problem that minimizes a weighted sum of the nuclear norms of all

Table 3: Comparison of tensor decomposition for spatiotemporal traffic data.

Tech.	Author (year)	Con. S	Con. T	Con. O	Solution	Data	Task
CP	Zhou et al. (2015)	Regression	Toeplitz	-	ALS	OD flow	[I]
	Afshar et al. (2017)	-	-	Orthogonal	Projected GD	Trip record	[PD]
	Wu et al. (2018)	Graph, TV	TV	ℓ_2, ℓ_1 norm	Projected GD	Traffic flow	[I]
	Xie et al. (2018a)	-	-	Reshape	ALS	OD flow	[I]
	Yang et al. (2019)	Similarity	-	Nonnegative	ALS	Traffic flow	[PD]
	Baggag et al. (2019)	Graph	Graph, AR	-	ALS	Traffic speed	[I] [P]
	Bhanu et al. (2020)	Graph	Graph	LSTM	ALS	Traffic volume	[P]
	Jia et al. (2020)	Weighted	Weighted	Network	GD	Vehicle GPS	[I]
	Yan et al. (2021)	-	Regression, spline	Residual	ALS	Traffic flow	[I] [P]
	Said and Erradi (2021)	Similarity	Entropy, FFT	-	ALS	Traffic flow	[I]
	Wang and Sun (2021)	-	VAR	ℓ_2 norm	ALS	Traffic speed	[AD]
	Zhang et al. (2022)	-	TV	DMD	ADMM	Traffic speed	[P]
	Xing et al. (2023b)	-	-	Data fusion	ALS	Traffic volume	[I]
	Lei and Sun (2024)	Covariance	Covariance	Residual	ALS	Traffic speed	[I]
Tucker	Tan et al. (2013)	-	-	-	Gradient	Traffic volume	[I]
	Chen et al. (2018)	-	-	ℓ_2 norm	GD	Traffic speed	[I]
	Yang et al. (2020a)	-	-	-	GD	Traffic flow	[I] [P]
	Gong and Zhang (2020)	-	-	Residual	SVD	Speed, flow	[I]
	Chen et al. (2023a)	-	VAR	-	Gradient	Trip record	[PD]
	Xu et al. (2024b)	-	-	-	KR	Traffic flow	[P]
	Xie et al. (2024)	Graph	Toeplitz	ℓ_2 norm	GD	Traffic speed	[I]
	Gong et al. (2025)	Manifold	Toeplitz	ℓ_1 on \mathcal{G}	Gradient	Speed, flow	[I]
Others	Tan et al. (2016)	-	AR	-	ALS	Traffic flow	[P]
	Deng et al. (2021a)	Graph spectral	Toeplitz	ℓ_2 norm	ALS	Traffic volume	[I]
	TR, Wu et al. (2023)	-	-	Nonlocal	Alternating	Traffic speed	[I]
	DPD, Lin et al. (2024a)	-	-	ℓ_2 norm	Convolution	Traffic speed	[P]
	TT, Zhang et al. (2024)	-	Toeplitz	ℓ_2 norm	SVD	Traffic state	[I]
	TD, Ming et al. (2024)	-	-	ℓ_2 norm	Gradient	OD pairs	[I]
	DPD, Chen et al. (2024a)	Toeplitz	-	Nonnegative	Gradient	Speed, flow	[I]
	TR, Yu et al. (2025)	Graph	Graph	ℓ_0 norm	ADMM	Speed, volume	[I]
	TT, Nguyen et al. (2025)	Graph	Graph	ℓ_2 norm	GD	Traffic flow	[I]
	Peng et al. (2025a)	GNN	Convolution	Residual	Gradient	Flow, speed	[P] [PD]

Abbreviations: Tech.: technique; Con. S / Con. T / Con. O: constraint for spatial mode, temporal mode, and other aspects; KR: kernel regression; GNN: graph neural network; [I]: imputation; [PD]: pattern discovery; [P]: prediction; [AD]: anomaly detection.

matricizations of the tensor. For \mathcal{Y} , the problem can be written as:

$$\min_{\mathbf{X}} \sum_{d=1}^D \beta_d \|\mathbf{X}_{(d)}\|_* \quad \text{s.t. } \mathbf{X}_\Omega = \mathcal{Y}_\Omega, \quad (19)$$

where $\mathbf{X}_{(d)}$ denotes the mode- d unfolding of \mathbf{X} , and $\beta_d > 0$ are weight parameters. This optimization is usually solved using iterative algorithms such as alternating direction method of multipliers (ADMM), where each subproblem applies singular value thresholding (SVT) to enforce low-rankness (Liu et al., 2012). Conceptually, this formulation encourages each unfolding of \mathbf{X} to be low-rank, which is a tractable proxy for overall low-rank structure in a tensor case (Wimalawarne and Mamitsuka, 2021; Peng et al., 2025b; Peng et al.).

Convex completion methods have the advantage of not requiring the rank to be specified—the nuclear norm adaptively finds an appropriate rank—and they guarantee convergence to a global optimum. In traffic applications, convex tensor completion has shown good performance when data sizes are moderate and missing rates are not extremely high. However, these methods can be computationally expensive for very large tensors due to repeated SVD computations on large matrices. It can also introduce biased reconstructions and potentially underestimates true values due to the over-shrinkage of large singular values, particularly when dominant components are critical, e.g., rush hour peaks in traffic data.

To mitigate the bias introduced by nuclear norm penalization, various nonconvex and factorization-based have been developed. One effective strategy of nonconvex surrogates is the **truncated nuclear norm** (TNN) (Zhang et al., 2012; Xue et al., 2018), which avoids penalizing the largest k singular values. This allows the model to preserve primary components while still denoising the remaining structure. For a

$M \times T$ matrix \mathbf{X} , the TNN can be defined as (Hu et al., 2012):

$$\|\mathbf{X}\|_{*,k} = \sum_{i=k+1}^{\min(M,T)} \sigma_i(\mathbf{X}), \quad (20)$$

where k is a user-defined parameter, typically the expected rank of the matrix. The underlying assumption is that the top k singular values represent the principal low-rank signal, while the rest may be attributed to noise or error. The TNN-based tensor completion problem on \mathcal{Y} is formulated as:

$$\min_{\mathbf{X}} \sum_{d=1}^D \beta_d \|\mathbf{X}_{(d)}\|_{*,k}, \quad \text{s.t. } \mathbf{X}_\Omega = \mathcal{Y}_\Omega. \quad (21)$$

These nonconvex optimization problems are often solved with iterative reweighting or alternating minimization schemes (Xu and Yin, 2013). Chen et al. (2020) applied TNN to traffic speed data and demonstrated that tuning the truncation threshold parameter leads to more accurate imputation by retaining meaningful patterns, such as daily cycles.

Another family of nonconvex penalties is based on the **Schatten- p norm** with $0 < p < 1$, which generalizes the nuclear norm (Schatten-1, i.e., a special case with $p = 1$) and penalizes singular values more aggressively (Nie et al., 2012; Xie et al., 2016; Gao et al., 2020). For a matrix $\mathbf{X} \in \mathbb{R}^{M \times T}$, the Schatten- p norm is defined as:

$$\|\mathbf{X}\|_{S_p} = \left(\sum_{i=1}^{\min(M,T)} \sigma_i^p(\mathbf{X}) \right)^{\frac{1}{p}}, \quad 0 < p < 1, \quad (22)$$

where smaller values of p encourage stronger low-rank regularization by more heavily penalizing smaller singular values. Extending to tensors, the completion problem is:

$$\min_{\mathbf{X}} \left(\sum_{d=1}^D \beta_d \|\mathbf{X}_{(d)}\|_{S_p}^p \right)^{\frac{1}{p}}, \quad \text{s.t. } \mathbf{X}_\Omega = \mathcal{Y}_\Omega, \quad (23)$$

which is also typically solved by iterating or alternating minimization algorithms. Empirical results suggest that Schatten- p norms can achieve higher-quality recovery than convex nuclear norm approaches, although they can introduce challenges such as sensitivity to local minima and increased hyperparameter tuning complexity.

In the context of traffic data, several studies have also explored surrogate norms that extend the Schatten- p family. For example, Nie et al. (2022) introduced truncated Schatten- p norm for traffic imputation, Hu et al. (2023) applied the Schatten capped p norm, while Wu et al. (2022a) employed a p -shrinkage operator (Wu et al., 2024b), both showing enhanced performance in capturing complex spatiotemporal traffic patterns.

In parallel to norm-based approaches, factorization-based methods assume an explicit CP/Tucker or other forms of decomposition for the estimated tensor \mathbf{X} and also forms a type of rank surrogate for completion (Cai et al., 2022; Luo et al., 2025a; Pan et al., 2024; Gong et al., 2023; Zhou et al., 2017; Yang et al., 2020b; Yu et al., 2023). It turns the tensor completion problem into the task of learning low-dimensional factor matrices. As introduced in Section 3.1, these approaches optimize the latent factors to fit the observed data. By fixing a rank R , factorization-based models effectively parametrize the low-rank constraint and reduce the optimization problem to a much smaller set of parameters corresponding to the factor matrices. Examples include the method by Acar et al. (2011), which performs rank minimization via CP decomposition, and the work in Xu et al. (2023a), which applies factorization techniques for matrix completion. Factorization-based approaches avoid the manipulation of large unfolded matrices and can handle large-scale datasets efficiently. However, they require the rank R to be specified in advance, a task that can be nontrivial and have a significant impact on model performance and reconstruction quality.

We summarize commonly used rank-surrogate models and representative studies for tensor completion in Table 4. Overall, solving rank-surrogate based tensor completion effectively remains an active area of research. Convex formulations such as nuclear norm minimization offer theoretical guarantees and can be scaled via techniques such as parallel computing; however, their computational cost grows rapidly with tensor size. Nonconvex and factorization-based methods tend to be more efficient and faster per iteration, but they come with their own challenges: factorization-based approaches require careful tuning of the rank, while nonconvex methods often demand well-designed initialization to avoid local minima. To address

Table 4: Rank-surrogate-regularized models for tensor completion (the problem formulation subject to $\mathcal{X}_\Omega = \mathcal{Y}_\Omega$).

Rank surrogate	Author (year)
Nuclear norm: $\min_{\mathcal{X}} \sum_{d=1}^D \beta_d \ \mathcal{X}_{(d)}\ _*$ (Convex)	HaLRTC, Liu et al. (2012) Semerci et al. (2014) Weighted NN, Gu et al. (2014) Zhang and Aeron (2016) Weighted NN, Gu et al. (2017) Lu et al. (2019) Jiang et al. (2020a) Liu et al. (2024)
Truncated nuclear norm: $\min_{\mathcal{X}} \sum_{d=1}^D \beta_d \ \mathcal{X}_{(d)}\ _{*,k}$ (Nonconvex)	Matrix TNN, Zhang et al. (2012) Matrix TNN, Hu et al. (2012) Xue et al. (2018) Li et al. (2020a)
Schatten- p norm: $\min_{\mathcal{X}} \left(\sum_{d=1}^D \beta_d \ \mathcal{X}_{(d)}\ _{S_p}^p \right)^{\frac{1}{p}}$ (Nonconvex)	Matrix, Nie et al. (2012) Weighted, Xie et al. (2016) Shang et al. (2017) t-Schatten- p , Kong et al. (2018) Weighted, Gao et al. (2020)
Others	TF, Zhou et al. (2017) TT, Yang et al. (2020b) TT, Cai et al. (2022) Tucker, Gong et al. (2023) TR, Yu et al. (2023) Tucker, Pan et al. (2024) TW, Luo et al. (2025a)

Abbreviations: NN: nuclear norm; TNN: truncated nuclear norm; TF: tensor factorization; TT: tensor train; TR: tensor ring; TW: tensor wheel.

these limitations, recent studies have focused on making such methods adaptive or parameter-free. For example, [He et al. \(2022\)](#) proposed a nonconvex tensor completion model that adjusts the rank penalty during optimization, thereby reducing the need to predefined the rank or regularization parameters. In practice, a common strategy is to set aside a small validation set of observed entries and perform cross-validation to select hyperparameters such as rank or regularization weights.

Application to traffic data. Rank-surrogate models have shown particular effectiveness for traffic data analysis given the strong repetitive and structured patterns inherent in real-world traffic data ([Lu et al., 2025](#); [Zeng et al., 2024b](#)). Similar to the representation used in tensor decomposition, traffic speed data can be organized as a third-order tensor with dimensions *location* \times *time of day* \times *day*. Rank-surrogate methods can then be applied to recover missing entries by borrowing information from: (i) other times of the same day (temporal similarity), (ii) the same time across different days (daily periodicity), and (iii) other locations with similar traffic profiles (spatial correlation). Early works applied matrix completion to either individual time series (per location) or time snapshot (per network state), tensor-based methods exploit all spatiotemporal dependencies simultaneously. Related studies show that tensor methods often outperform matrix-based completion in traffic imputation ([Ran et al., 2016b](#)). For example, [Ran et al. \(2016b\)](#) applied a forth-order tensor representation and yielded more accurate imputation than flattening weekly data into a single dimension. The key advantage is the ability of rank-surrogate minimization to capture day-to-day similarities—for instance, if data from a specific Monday are missing, the model can infer it by combining patterns from other Mondays and trends earlier in the same week. Rank-surrogate models have been applied to various traffic datasets, including freeway loop detector counts ([Chen et al., 2021a](#)), travel speeds ([Chen et al., 2021b, 2020](#)), and OD travel demands ([Xie et al., 2018b](#)), showcasing its versatility across multiple traffic data modalities.

Recent rank-surrogate relaxation models have been validated on large real-world datasets, demonstrating not only high accuracy but also computational efficiency. The output of tensor completion is an imputed tensor, which can serve as input for downstream applications that require fully observed traffic data, such as short-term forecasting ([Chen et al., 2021b](#)) or anomaly detection ([Wang et al., 2021](#)). In practice, tensor completion has become a fundamental preprocessing step in traffic data analysis pipelines, substantially alleviating data sparsity issues and enabling effective analysis in the presence of missing observations. Sim-

ilar to tensor decomposition, incorporating spatiotemporal dependencies is crucial within rank-surrogate completion frameworks for traffic data (Chen et al., 2021b). In the following subsections, we describe how spatiotemporal consistencies are integrated into surrogate-based completion models.

3.2.2. Spatiotemporal modeling

(1) *Spatial consistencies.* As in tensor decomposition, leveraging spatial correlations is essential for effective tensor completion in traffic data analysis (Nie et al., 2023; Chen et al., 2024d). In rank-surrogate based completion algorithms, spatial structures can be incorporated in various ways, depending on the specific modeling approach (Li et al., 2017; Wu and Fan, 2024). In traffic data analysis, incorporating spatial consistencies aims to exploit the similarity or connectivity between traffic sensors or road segments to enhance modeling accuracy, especially in areas with sparse or missing data (Nie et al., 2023).

For singular-value-based rank-surrogates completion methods built on tensor unfolding and tensor-norm minimization, spatial correlations can be incorporated by unequal weighting of norms or adding structural constraints (Cheng et al., 2024; Ng et al., 2017; Zhao et al., 2024b). Some studies distinguish between unfoldings along the spatial and temporal modes (Cheng et al., 2024). For instance, a higher weight can be assigned to the unfolding with spatial locations along one axis, reflecting the intuition that spatial data may exhibit a low-rank structure distinct from temporal patterns. Imposing a stronger low-rank penalty on the spatial unfolding biases the model toward identifying spatial patterns such as clusters of sensors with similar temporal behavior. Nevertheless, many works on spatiotemporal traffic data adopt equal weighting across unfoldings (Chen et al., 2020, 2024e). To further incorporate road network topology, graph Laplacian constraints can be applied directly to the spatial unfolding—simile to approaches in tensor decomposition (Section 3.1.2), but imposed on the unfolding matrix rather than on spatial factor matrices—as demonstrated in Shu et al. (2024); Nie et al. (2023); Hu et al. (2024); Sofuoğlu and Aviyente (2022). These methods capture spatial redundancy and network-based dependencies, which is especially helpful when data are extremely sparse or contain block missing scenarios, e.g., an entire region of sensors failing simultaneously, as in kriging or spatial interpolation. In such cases, a vanilla low-rank model can struggle to infer values for completely missing regions, whereas incorporating road network topology/connectivity or typical flow propagation patterns can help bridge these gaps (Li et al., 2018b; Xia et al., 2025; Gürsun and Crovella, 2012). Furthermore, TV (total variation) based spatial regularization has also been used to encourage piecewise-smooth fields while preserving sharp boundaries (Li et al., 2017; Qiu et al., 2021; Liu et al., 2023).

For factorization-based completion methods, where the factor matrices are explicitly learned, spatial regularization terms—such as graph Laplacian or variation-based penalties—can be imposed either on the spatial factor matrix or the corresponding unfolding to ensure that the reconstructed tensor adheres to the road network smoothness (Li et al., 2017; Zheng et al., 2019; Yu et al., 2023, 2024). These methods are analogous to adding a Laplacian/variation term in tensor decomposition (Yokota et al., 2016b; Li et al., 2017), but adapted to rank-surrogate based completion.

Another approach involves hybrid models that combine rank-surrogate models with other spatial interpolation techniques. For example, Xia et al. (2025) integrate a low-rank completion objective with a graph-based network, adding a graph convolution/diffusion term that refines fine-grained local details using the road network. Such graph network fusion captures local spatial variations that purely low-rank structures may miss, resulting in more accurate and spatially coherent imputations.

In summary, spatial consistency is introduced into completion models through (i) tailored weighting of low-rank penalties across unfoldings, (ii) graph Laplacian regularizers that encode network topology, (iii) local variation smoothness, and (iv) fusion with network-based interpolation. These techniques make imputation geographically consistent—for example, if data of a certain road are missing, the model infers them from neighboring roads and overall traffic trends rather than arbitrary values. Empirically, adding spatial regularization markedly improves performance under sparsity or contiguous block-missing patterns, e.g., regional sensor outages, preventing implausible spatial discontinuities and helping recover phenomena such as congestion that typically span adjacent links.

(2) *Temporal dependencies.* Modeling temporal dependencies is equally important in rank-surrogate based completion for traffic data. A low-rank tensor structure on its own captures some temporal consistency. For instance, if a *day* mode or similar is included, the low-rank model naturally learns repeated daily or weekly traffic patterns, as shown in Ran et al. (2016b); Liu et al. (2019a). Specifically, Ran et al. (2016b) built a 4-way tensor (*location* \times *time of day* \times *day of week* \times *week*) to encode weekly periodicity, and Liu et al. (2019a) used a year/month organization to force annual seasonality. These formulations implicitly ensure that the

completion algorithm uses data from the same time-of-day on other days, or the same day-of-week in other weeks, to inform a missing entry. In other words, part of temporal consistency can be achieved by aligning and stacking the data appropriately along temporal modes before running completion. Additionally, local variations such as first-order differences have been incorporated into rank-surrogate-based completion methods for temporal smoothness as well (Sofuoğlu and Aviyente, 2022). For example, Chen et al. (2025b) arguments a tubal rank based spectral-norm minimization objective with a local variation penalty to encourage temporal smoothness and improve traffic imputation.

However, explicit temporal regularization is often needed, especially when data is missing over long continuous time intervals, e.g., a sensor not reporting for hours or days. In such cases, relying only on low-rank structure or local differencing may fail to capture the sequential nature of traffic evolution. Therefore, researchers have embedded time-series modeling inside the completion objective. A representative example is the low-rank autoregressive tensor completion (LATC) model (Chen et al., 2021b) which augments a standard tensor completion objective with an AR constraint on the time mode unfolding. LATC introduces a temporal-variation based regularizer that makes consecutive time steps in the unfolded matrix follow an AR process. With this design, even when long consecutive periods across all locations are missing, the model can interpolate through the gap by essentially simulating an AR process using dynamics learned from observed periods. Let $\mathcal{L} = \{\ell_1, \dots, \ell_d\}$ be a order- d time lag set and $\mathbf{A} \in \mathbb{R}^{M \times d}$ a coefficient matrix, the optimization problem of LATC on a third-order $location \times time\ of\ day \times day$ partially observed tensor $\mathbf{Y}_\Omega \in \mathbb{R}^{M \times T \times P}$ can be written as:

$$\min_{\mathbf{X}, \mathbf{Z}, \mathbf{A}} \|\mathbf{X}\|_{*,k} + \frac{\rho}{2} \|\mathbf{Z}\|_{\mathbf{A}, \mathcal{H}}, \quad \text{s.t. } \mathbf{X} = \mathcal{Q}(\mathbf{Z}), \quad \mathbf{Z}_\Omega = (\mathbf{Y}_{(1)})_\Omega, \quad (24)$$

where $\|\mathbf{Z}\|_{\mathbf{A}, \mathcal{H}} = \sum_{m,t} \left(z(m,t) - \sum_{i=1}^d a(m,i)z(m,t-\ell_i) \right)^2$ is the temporal-variation on $\mathbf{Z} \in \mathbb{R}^{M \times (TP)}$, $\mathcal{Q}(\cdot)$ denotes a forward tensorization operator that converts the multivariate time series matrix into a third-order tensor, and $\|\cdot\|_{*,k}$ is the TNN (truncated nuclear norm).

Beyond AR, other dynamic constraints have also been explored in completion. For example, Wang et al. (2023b) incorporated Hankel-based temporal regularization into a rank-surrogate tensor completion method (paralleling the Hankel discussions in Section 3.1.2). The idea is to enforce that the Hankel embedding transformed unfolding is low-rank, thereby capturing periodic patterns in the completed data and enabling long-gap filling across all locations by leveraging repeated structures in the time series. Another line of work operates in the transformed/spectral domain (Shu et al., 2024; Chen et al., 2025c; Liu et al., 2025c,b): Shu et al. (2024) apply a discrete cosine transform (DCT) along the time (and optionally spatial) modes of the incomplete data, perform low-rank completion in the DCT domain, then transform back. Since traffic time series often contain only a few dominant frequency components (daily/weekly cycles, etc.); recovering in the DCT domain can be easier than direct the time domain extrapolation when the data are periodic. In practice, the DCT approach assumes the traffic data are approximately periodic and that noise can be separated as higher-frequency components; zeroing or shrinking selected frequencies during low-rank approximation achieves a temporal smoothing effect once transformed back to the original domain.

In summary, temporal correlations are infused into rank-surrogate completion algorithms through mechanisms including: aligning data to exploit periodicity; adding AR or other dynamic penalties to ensure realistic temporal evolution; imposing Hankel structured low-rank constraints to capture long-range dependencies; and transforming data to the frequency domain to enforce smoothness or repetition. The net effect is that the imputed tensor not only has low-rank structure but also exhibits plausible traffic time series behavior at each location. When spatial and temporal constraints are combined, the completion problem essentially becomes a spatiotemporal interpolation guided by historical patterns in both dimensions. This often improves performance over using only spatial or temporal context. For example, a joint model can leverage that a missing reading at a specific sensor at 5 PM on Tuesday can be inferred from “Tuesdays at 5 PM follow a typical pattern” (temporal) and “a given sensor behaves similar to its neighboring sensors” (spatial). Empirically, integrated spatiotemporal models consistently outperform separate approaches, especially under very high missing rates or structured missing blocks. We next discuss methods that integrate both spatial and temporal modeling within the rank-surrogate based completion frameworks.

(3) *Joint spatiotemporal modeling and other aspects.* In tensor completion, spatial and temporal correlations are often modeled inherently given that a multi-dimensional low-rank structure ties together all modes. A standard low-rank completion already uses dependencies across any two modes to fill in the third. For instance, in a spatiotemporal matrix, if an entire row (a spatial location) is missing over a period, the model can still

reconstruct it by using temporal patterns learned from other times and spatial patterns from other locations at those times. [Ran et al. \(2016b\)](#) demonstrated this advantage: their fourth-order tensor (with a *day of week* mode) yielded better imputations than matrix completion by simultaneously leveraging weekly seasonality and spatial groupings of sensors. [Yang et al. \(2020a\)](#) similarly found that completing a third-order traffic flow tensor (*location* \times *time of day* \times *day*) produced more consistent estimates across neighboring detectors than completing time series at each location independently, underscoring the benefit of joint consideration.

Beyond the implicit coupling provided by the tensor format, many works add explicit joint spatiotemporal regularization to the completion objective. Most of the joint strategies from Section 3.1.2 carry over here ([Liu et al., 2025c,b](#)). A common design is to combine graph/variation-based spatial smoothness with temporal AR/variation penalties within the same rank-surrogate framework ([Li et al., 2024b; Yu et al., 2024; Zeng et al., 2024b](#)). For example, [Bhanu et al. \(2020\)](#) proposed a CP-based traffic completion model which uses graph Laplacians along both space and time. Residual modeling can likewise be adapted to completion ([Lu et al., 2019; Feng et al., 2024; Shu et al., 2025; Zhao et al., 2021; Yang et al., 2020b](#)). For instance, GLOSS (graph regularized low-rank plus temporally smooth sparse decomposition) ([Sofuo glu and Aviyente, 2022](#)) decomposes the incomplete tensor into a smooth low-rank global background and a sparse residual for traffic anomalies, applying graph Laplacian penalties to each unfolding of the low-rank part and ℓ_1 norm with TV (along the temporal mode) to the sparse local anomaly component. Similar frameworks have been used in [Shu et al. \(2025\); Wang et al. \(2023a\); Zeng et al. \(2025\); Lyu et al. \(2024\)](#) for traffic imputation. In addition, [Chen et al. \(2024b\)](#) used a neural network to model the local consistency within the completion model.

A practical challenge in joint modeling is parameter tuning—when we have multiple regularization terms, e.g., spatial, temporal, etc., we have multiple hyperparameters to set, e.g., the weight of spatial smoothness vs. temporal smoothness vs. rank penalty. In addition, convex methods have regularization parameters for nuclear norm weights, nonconvex methods have truncation levels (k parameter in TNN) or p values in Schatten- p norm. This can be addressed by cross-validation or automated methods as mentioned. [He et al. \(2022\)](#) adaptive rank penalty is one example of trying to reduce manual tuning. In many cases, researchers will do a grid search on a small validation set to find a good balance of regularization weights that minimizes imputation error on that set. Such procedure increases computational overhead but ensures the model is neither over nor under-fit and usually pays off in better performance.

When choosing a tensor completion method for a specific traffic application, one should consider factors including: the missing data pattern, the size of the dataset, and whether directional information is needed for the given task e.g., interpolation and prediction ([Chen et al., 2021b; Wang et al., 2023b](#)). Convex nuclear norm minimization ([Liu et al., 2012](#)) is mathematically elegant and simple in approach, but for large-scale problems its heavy SVD computations and shrinkage bias have led to more adoption of nonconvex or factorization based methods in recent years ([Chen et al., 2021a](#)). Nonconvex surrogates or factorization methods can achieve more accurate results that are closer to the true unobserved values and scale to larger tensors, but require careful handling as discussed. There is evidence that with proper initialization and parameter tuning, nonconvex methods consistently outperform convex ones in traffic data completion ([Chen et al., 2020](#)). In terms of scalability, methods such as the tubal-Fourier approach ([Chen et al., 2021a](#)) demonstrate that leveraging unitary transforms which convert a large 3D problem into small matrix problems can overcome the computational barrier, enabling completion on tensor cubes of dimension such as $1000 \times 24 \times 365$ on standard hardware, which can be promising for city-scale deployments.

To sum up, rank-surrogate relaxation techniques have matured to handle a wide variety of scenarios in spatiotemporal traffic data. They enforce global consistency via low-rank structure while allowing incorporation of domain-specific knowledge such as spatial network connectivity and temporal patterns, etc. through regularization. The end result—a completed data tensor—is an enabling asset for ITS analytics: it provides a cleaned, reconstructed view of the traffic state that can be used for tasks e.g., short-term forecasting, traffic control strategy development, anomaly detection, and pattern analysis, even in the face of originally incomplete or noisy data.

(4) *Summary.* To put the different approaches in perspective, we summarize a few comparisons from the literature:

- **Convex vs. nonconvex methods:** Convex tensor completion methods such as nuclear norm minimization as in HaLRTC ([Liu et al., 2012](#)) is mathematically elegant and avoids choosing the rank explicitly, but it can struggle with scalability and may introduce estimation bias due to uniformly shrinking singular values. In practice, nonconvex and factorization-based methods have become more popular for

large traffic problems, as they typically yield better reconstruction accuracy and converge faster when well-tuned.

- **CP vs. Tucker vs. transform-based completion:** For factorization-based completion methods, which tensor representation to use can depend on data characteristics. CP-based completion is simple and effective when the true data tensor has a very low rank and the missing data is not too extreme. Tucker-based completion, either via nuclear norm on unfoldings or direct core optimization, allow different rank settings for different modes and often leads to good performance without intensive rank tuning ([Chen et al., 2018](#)). Transform methods, e.g., tubal/Fourier transformation, treat one mode often time dimension in the frequency domain. When traffic data has clear periodic components such as daily or weekly cycles, these methods can be efficient and accurate by capturing the periodic low-rank structure ([Chen et al., 2021a](#)). Overall, CP is a special case with same low rank across all modes, Tucker covers the multilinear rank scenario, and transform methods convert tensor completion into structured matrix completion problems which can be advantageous. The choice should be guided by the data: if a handful of global patterns explain the data, CP or low multilinear-rank Tucker can work well; if the data has a complex mix of patterns or is better viewed in another basis e.g., frequency domain, transformed approaches or more heavily regularized models may perform better.
- **Regularized vs. unregularized models:** Incorporating spatial/temporal regularization almost always improves accuracy when the regularizers are appropriately set and properly tuned. For instance, [Gong et al. \(2025\)](#) showed that a spatiotemporal-regularized Tucker model significantly outperformed both a plain low-rank model and a pure time series model in imputing traffic data under high missing rates. The regularized model could make usage of data in challenging situations by leaning on the smoothness priors, whereas the unregularized methods either overfit noise or underfit the structure. The downside of heavy regularization is the need for additional hyperparameters. In scenarios with little prior information, a strategy is to start with an unregularized model to avoid the risk of imposing incorrect assumptions. When reliable side information is available, such as a road graph or known periodicity, using it in regularization can provide a boost in performance.
- **Handling different missing patterns:** Tensor completion tends to have a clear advantage when the missing pattern is structured, e.g., non-random. If entire days or entire sensor streams are missing, which is a common occurrence in traffic applications due to sensor outages or maintenance, per-sensor or matrix methods struggle, whereas a tensor method can still fill those gaps by using data from other modes. Studies have found that under such block-missing scenarios, low-rank tensor models vastly outperform simpler baselines because they effectively borrow strength from the other dimensions. For more general missing patterns, e.g., 5% of readings missing at random, many methods including simple models can perform well, but low-rank tensor completion approaches still often yield the lowest errors by leveraging the full multidimensional correlations. At extremely high missing rates, e.g., 90% missing, Bayesian methods start to shine, as they can avoid parameter tuning and automatically impose structure with prior assumption without overfitting. We discuss Bayesian low-rank tensor models in Section 4.

We summarize and compare rank-surrogate tensor-completion studies for traffic data in Table 5, organized by publication year. As can be seen,

- Rank-surrogate formulations (e.g., tensor nuclear norm and related surrogates) have been widely used for traffic data completion, particularly imputation/extrapolation and anomaly detection. A sparse anomaly (residual) tensor component is naturally incorporated in tensor completion via an ℓ_1 sparse/robust penalty. Compared with tensor decomposition, surrogate models operate directly on the completed/estimated tensor and avoids explicit rank selection, but they generally do not expose mode-wise latent factors for interpretation.
- The spatial/temporal regularization terms used in rank-surrogate completion models are similar as those used in tensor decomposition, such as local graph Laplacian or variation/difference (TV/QV) penalties and AR/VAR dynamics. Over time, studies increasingly layer multiple spatiotemporal constraints and hybrid priors to better capture structure and improve model performance/robustness.
- Completion frameworks for general spatiotemporal data can be straightforwardly adapted for traffic data, such as spectral-domain completion model that exploits Fourier/spectral features [Tu et al. \(2025\)](#). A recent trend is to augment rank-surrogates with learnable transform or neural components

Table 5: Comparison of rank-surrogate models based low-rank completion for spatiotemporal traffic data.

Tech.	Author (year)	Con. S	Con. T	Con. O	Solution	Data	Task
NN	Ran et al. (2016b)	-	-	-	ADMM	Traffic flow	[I]
	Chen et al. (2021a)	-	-	Tubal	ADMM	Traffic speed	[I]
	Wang et al. (2021)	-	Hankel	ℓ_1 norm	ADMM	Traffic flow	[AD]
	Sofuooglu and Aviyente (2022)	Graph	Graph	Residual	ADMM	Traffic speed	[AD]
	Chen et al. (2022)	-	-	ℓ_1 norm	ADMM	Volume, speed	[I]
	Li et al. (2023a)	-	-	Group	ADMM	Traffic speed	[I]
	Li et al. (2024b)	Graph	Variation	-	ADMM	OD flow	[I]
	Zeng et al. (2024b)	TV	TV	-	ADMM	Traffic speed	[I]
	Lu et al. (2025)	-	-	Transformed	ADMM	Traffic speed	[I]
	Zeng et al. (2025)	TV	TV	Nonlocal	ADMM	Traffic speed	[I]
TNN	Chen et al. (2020)	-	-	-	ADMM	Traffic speed	[I]
	Sure et al. (2021)	-	-	-	ALS	Traffic volume	[I]
	Chen et al. (2021b)	-	VAR	-	ADMM	Speed, flow	[I] [E]
	Wang et al. (2023b)	Hankel	Hankel	-	ADMM	Trajectory	[I]
	Shu et al. (2024)	Graph	Difference	Circulant	ADMM	Traffic speed	[I]
	Zhao et al. (2024c)	-	-	Nonnegative	ADMM	Traffic speed	[I]
	Li et al. (2024a)	-	-	Convolution	ADMM	Traffic speed	[I]
Schatten- p	Yu et al. (2020)	-	-	-	Alternating	Trajectory	[I]
	Nie et al. (2022)	-	-	-	ADMM	Speed, volume	[I]
	Zhao et al. (2023b)	Manifold	-	-	ADMM	Traffic speed	[I]
	Nie et al. (2023)	Graph	-	-	ADMM	Traffic speed	[I] [E]
Others	Xie et al. (2018b)	-	-	Sequential	Alternating	OD	[I]
	p -shrinkage, Wu et al. (2022a)	-	-	Weighting	ADMM	Traffic speed	[I]
	TF, Xu et al. (2023a)	Hessian	Variation	-	ADMM	Speed, volume	[I]
	Capped- p , Hu et al. (2023)	-	-	Residual	ADMM	Speed, volume	[I]
	Chen et al. (2024c)	Penalty	Penalty	-	ADMM	Volume, speed	[I]
	Circulant NN, Chen et al. (2024d)	Graph	Circulant	-	ADMM	Speed, volume	[I]
	Chen et al. (2024e)	-	-	Regression	ADMM	Traffic speed	[I]
	Chen et al. (2024b)	Penalty	Penalty	Deep PnP	ADMM	Traffic speed	[I]
	TT, Yu et al. (2024)	Toeplitz	Toeplitz	ℓ_2 norm	Alternating	Traffic speed	[I]
	Hu et al. (2024)	Laplacian	Laplacian	Sparse	ADMM	Traffic state	[I]
	Xia et al. (2025)	-	-	GNF	ADMM	Traffic speed	[I]
	$\ell_1 \ell_2$ norm, Shu et al. (2025)	-	-	Residual	ADMM	Traffic speed	[I]

Abbreviations: Tech.: technique; NN: nuclear norm; TNN: truncated nuclear norm; Con. S / Con. T / Con. O: constraint for spatial mode, temporal mode, and other aspects; PnP: plug-and-play; TT: tensor train; GNF: graph network fusion; [I]: imputation; [E]: extrapolation; [AD]: anomaly detection.

to capture complex dependencies (Wu et al., 2022b; Yang et al., 2022). Similar research have embedded deep neural networks (DNNs) into tensor SVD pipelines (Luo et al., 2022b) or introduce parameterized neural modules within low-rank completion (Xu et al., 2025a; Xie et al., 2025; Wu et al., 2026).

Overall, LRTL—through both decomposition and rank-surrogate techniques—provides a powerful toolkit for spatiotemporal traffic data analysis. By maintaining a balance between data fidelity and structural regularization (whether as constraints or objective penalties), these methods extract meaningful low-dimensional patterns, respect the inherent spatial and temporal organization of traffic, and effectively tackle challenges of missing or noisy data that pervade real-world ITS applications.

4. Bayesian Probabilistic LRTL for Spatiotemporal Traffic Data

Deterministic LRTL effectively uncovers latent low-dimensional structure but lacks mechanisms for quantifying uncertainty, which is essential for reliable decision-making in real-world transportation systems. In contrast, **Bayesian LRTL** offers a probabilistic framework that jointly learns latent structure and provides *uncertainty quantification (UQ)* (Shi and Shen, 2023). By placing prior distributions over model parameters in the latent space (latent factors) and inferring posterior and predictive distributions conditioned on the observed data, Bayesian LRTL methods deliver calibrate UQ (credible and predictive intervals) alongside point estimates, enabling robust, interpretable, and uncertainty-aware estimations for traffic datasets characterized by high variability, missingness, and measurement error.

In this section, we first review and introduce Bayesian LRTL for spatiotemporal traffic data analysis (Section 4.1); then bridge deterministic and Bayesian LRTL formulations (Section 4.2), clarifying how opti-

mization constraints map to Bayesian probabilistic priors; and lastly compare Bayesian tensor factorization (TF) with Gaussian process (GP)—a typical probabilistic framework for spatiotemporal modeling with UQ (Section 4.3), discussing their respective advantages in traffic applications.

4.1. Bayesian tensor factorization for spatiotemporal traffic data

This subsection reviews Bayesian probabilistic low-rank factorization approaches in the context of spatiotemporal traffic data analysis. We begin with core concepts and foundational Bayesian factorization models (Section 4.1.1), then explain how spatiotemporal dependencies are incorporated within these probabilistic frameworks and also discuss practical considerations such as rank determination and heterogeneous noise modeling (Section 4.1.2).

4.1.1. Basics and fundamental models

Bayesian probabilistic modeling provides a principled framework for jointly estimating model parameters and quantifying uncertainty. In low-rank learning, Bayesian (hierarchical) TF assigns priors on model parameters (latent factors etc.) and hyperpriors on related hyperparameters, specifies a likelihood for the observed data, and infers full posterior (predictive) distributions for unobserved data rather than point estimates as in deterministic LRTL, enabling data-driven parameter estimation and UQ (Chen and Sun, 2021; Lei et al., 2022a).

Usually, two key components are required to construct a Bayesian low-rank factorization model: **(1) model specification**, including

- (i) **priors** on model parameters, e.g., latent factors and model noise parameters, with **hyperpriors** on corresponding hyperparameters, and
- (ii) **likelihood** formulation for the observations;

and **(2) posterior inference algorithm** that used to approximate the intractable posterior distribution, such as Markov chain Monte Carlo (MCMC) (Salakhutdinov and Mnih, 2008; Lei et al., 2022b) and variational inference (VI) (Kong et al., 2023; Chen et al., 2019a). We next elaborate on these components, outlining how Bayesian low-rank tensor models are defined, estimated, and deployed in practice.

Model specification. Consider a spatiotemporal observation matrix $\mathbf{Y}_\Omega \in \mathbb{R}^{M \times T}$, where Ω is the index set of observed entries, rows correspond to M spatial locations, and columns to T time points. A probabilistic MF (matrix factorization) model expresses the observed entries \mathbf{Y}_Ω as:

$$\mathbf{Y}_\Omega = (\mathbf{U}\mathbf{V}^\top)_\Omega + \mathbf{E}_\Omega,$$

where $\mathbf{U} \in \mathbb{R}^{M \times R}$ and $\mathbf{V} \in \mathbb{R}^{T \times R}$ are latent factor matrices for the spatial and temporal dimensions (with latent rank R), respectively, and $\mathbf{E} \in \mathbb{R}^{M \times T}$ denotes the noise process matrix. The subscript $(\cdot)_\Omega$ indicates that the model is defined on the observed entries only.

A common prior for the latent factors, as used in Bayesian probabilistic MF (BPMF) (Salakhutdinov and Mnih, 2008), assumes each row vector of the latent factor matrices follow a multivariate Gaussian distribution:

$$\begin{aligned} \mathbf{u}_{m:} &\sim \mathcal{N}(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u^{-1}), \quad m = 1, \dots, M, \\ \mathbf{v}_{t:} &\sim \mathcal{N}(\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v^{-1}), \quad t = 1, \dots, T, \end{aligned} \tag{25}$$

where $\mathbf{u}_{m:} \in \mathbb{R}^R$ and $\mathbf{v}_{t:} \in \mathbb{R}^R$ represent the m th row of \mathbf{U} and t th row of \mathbf{V} , respectively, $\mathcal{N}(\cdot)$ denotes Normal distribution, with $\{\boldsymbol{\mu}_u \in \mathbb{R}^R, \boldsymbol{\Lambda}_u^{-1} \in \mathbb{R}^{R \times R}\}$ and $\{\boldsymbol{\mu}_v \in \mathbb{R}^R, \boldsymbol{\Lambda}_v^{-1} \in \mathbb{R}^{R \times R}\}$ as the mean and covariance matrix for rows of \mathbf{U} and rows of \mathbf{V} , respectively. We use *Normal* and *Gaussian* interchangeably throughout this paper. The residual noise matrix \mathbf{E} is typically modeled as white noise, with entries following i.i.d. (independent and identically distributed) Normal distribution:

$$e_{mt} \sim \mathcal{N}(0, \tau^{-1}), \tag{26}$$

where τ is the noise precision. With these assumptions, the likelihood for the observed elements is:

$$y_{mt} \sim \mathcal{N}(\mathbf{u}_{m:}\mathbf{v}_{t:}^\top, \tau^{-1}), \quad \forall(m, t) \in \Omega. \tag{27}$$

To complete a fully Bayesian hierarchical model, conjugate Normal-Wishart hyperpriors are often used for the row mean and precision hyperparameters $\Theta_u = \{\mu_u, \Lambda_u\}$ and $\Theta_v = \{\mu_v, \Lambda_v\}$:

$$\begin{aligned} p(\mu_u | \Lambda_u) p(\Lambda_u) &= \mathcal{N}(\mu_u | \mu_0, (\beta_0 \Lambda_u)^{-1}) \mathcal{W}(\Lambda_u | \mathbf{W}_0, \nu_0), \\ p(\mu_v | \Lambda_v) p(\Lambda_v) &= \mathcal{N}(\mu_v | \mu_0, (\beta_0 \Lambda_v)^{-1}) \mathcal{W}(\Lambda_v | \mathbf{W}_0, \nu_0), \end{aligned} \quad (28)$$

where $\mathcal{W}(\cdot)$ denotes Wishart distribution, with a positive-definite matrix $\mathbf{W}_0 \in \mathbb{R}^{R \times R}$ as the scale matrix and ν_0 specifying the degrees of freedom. Lastly, for the noise precision τ , usually a conjugate Gamma prior is assumed:

$$\tau \sim \text{Gamma}(a_0, b_0). \quad (29)$$

This hierarchical specification yields conditional conjugacy: the full conditionals for \mathbf{U} , \mathbf{V} , μ_u , Λ_u , μ_v , Λ_v , and τ remain in standard families, enabling Gibbs sampling with closed-form updates and efficient posterior inference with uncertainty estimation (Gelman et al., 2013).

It is straightforward to extend the two-dimensional BPMF model (Salakhutdinov and Mnih, 2008) to a higher-order tensor factorization settings (Chen et al., 2019a). For example, consider a third-order incomplete traffic speed observation tensor $\mathcal{Y}_\Omega \in \mathbb{R}^{M \times T \times P}$, where the three modes correspond to *location*, *time of day*, and *day*, respectively. A Bayesian CP factorization model approximates \mathcal{Y}_Ω as:

$$\mathcal{Y}_\Omega = \left(\sum_{r=1}^R \lambda_r \mathbf{u}_{:r} \circ \mathbf{v}_{:r} \circ \mathbf{w}_{:r} \right)_\Omega + \mathcal{E}_\Omega, \quad (30)$$

where $\mathbf{u}_{:r} \in \mathbb{R}^M$, $\mathbf{v}_{:r} \in \mathbb{R}^T$, and $\mathbf{w}_{:r} \in \mathbb{R}^P$ for $r = 1, \dots, R$ are the r th columns of the factor matrices $\mathbf{U} \in \mathbb{R}^{M \times R}$, $\mathbf{V} \in \mathbb{R}^{T \times R}$, and $\mathbf{W} \in \mathbb{R}^{P \times R}$, respectively, $\lambda = [\lambda_1, \dots, \lambda_R]^\top \in \mathbb{R}^R$ denotes the weight parameter, and $\mathcal{E} \in \mathbb{R}^{M \times T \times P}$ is the residual noise tensor.

Similar to BPMF, a standard modeling assumption is to place independent multivariate Normal priors on the rows of the latent factor matrices \mathbf{U} , \mathbf{V} , and \mathbf{W} . Additionally, a conjugate Normal prior is imposed on the component weights λ , and the residuals are modeled as i.i.d. Gaussian noise. For hierarchical modeling, conjugate Normal-Wishart hyperpriors are placed on the means and precision matrices of the row-wise priors. The noise precision τ is typically assigned a Gamma prior. The complete model specification is summarized below:

- Priors on latent factors:
 - For each location $m = 1, \dots, M$: $\mathbf{u}_{m:} \sim \mathcal{N}(\mu_u, \Lambda_u^{-1})$;
 - For each time-of-day point $t = 1, \dots, T$: $\mathbf{v}_{t:} \sim \mathcal{N}(\mu_v, \Lambda_v^{-1})$;
 - For each day $p = 1, \dots, P$: $\mathbf{w}_{p:} \sim \mathcal{N}(\mu_w, \Lambda_w^{-1})$.
- Hyperpriors on latent factor parameters:
 - $\Theta_u = \{\mu_u, \Lambda_u\}$: $\mu_u | \Lambda_u \sim \mathcal{N}(\mu_0, (\beta_0 \Lambda_u)^{-1})$, $\Lambda_u \sim \mathcal{W}(\mathbf{W}_0, \nu_0)$;
 - $\Theta_v = \{\mu_v, \Lambda_v\}$: $\mu_v | \Lambda_v \sim \mathcal{N}(\mu_0, (\beta_0 \Lambda_v)^{-1})$, $\Lambda_v \sim \mathcal{W}(\mathbf{W}_0, \nu_0)$;
 - $\Theta_w = \{\mu_w, \Lambda_w\}$: $\mu_w | \Lambda_w \sim \mathcal{N}(\mu_0, (\beta_0 \Lambda_w)^{-1})$, $\Lambda_w \sim \mathcal{W}(\mathbf{W}_0, \nu_0)$.
- Prior on component weights: $\lambda \sim \mathcal{N}(\mu_\lambda, \Sigma_\lambda)$.
- Hyperpriors on component weight parameters: $\Theta_\lambda = \{\mu_\lambda, \Sigma_\lambda\}$: $\mu_\lambda | \Sigma_\lambda \sim \mathcal{N}(\mu_0, \beta_0 \Sigma_\lambda)$, $\Sigma_\lambda^{-1} \sim \mathcal{W}(\mathbf{W}_0, \nu_0)$.
- Noise process: $e_{mtp} \sim \mathcal{N}(0, \tau^{-1})$. Prior: $\tau \sim \text{Gamma}(a_0, b_0)$.
- Likelihood: $y_{mtp} \sim \mathcal{N}\left(\sum_{r=1}^R u_{mr} v_{tr} w_{pr}, \tau^{-1}\right)$ for all observed indices $(m, t, p) \in \Omega$.

By specifying appropriate priors and hyperpriors over model parameters and hyperparameters, a fully Bayesian tensor factorization model is constructed. Such models allow for uncertainty-aware tensor decomposition, which can be especially useful for traffic data imputation under high missing rates.

The BPMF model introduced in [Salakhutdinov and Mnih \(2008\)](#) was among the first to fully formalize a fully Bayesian treatment of MF. This framework was later extended to higher-order tensors by [Zhao et al. \(2015\)](#), where a Bayesian CP factorization (BCPF) method was proposed. [Chen et al. \(2019a,b\)](#) applied similar Bayesian CP models to traffic datasets for traffic speed imputation. These models represent foundational approaches in Bayesian LRTL. We illustrate and compare the graphical model structures of representative Bayesian factorization models in Figure 7.

Model inference. Bayesian tensor factorization models are typically inferred using either Markov chain Monte Carlo (MCMC) methods or variational inference (VI) strategies ([Tzikas et al., 2008; Babacan et al., 2012; Ma et al., 2014](#)). Variational approaches, such as the Bayesian augmented tensor factorization (BATF) model proposed in [Chen et al. \(2019a\)](#), approximate the true posteriors by optimizing over a family of simplified tractable distributions. VI methods are often faster and more scalable compared with MCMC ([Salakhutdinov and Mnih, 2008; Luttinen, 2013](#)), and thus have been applied for online estimation/prediction ([Paliwal et al., 2021](#)). However, such computational efficiency comes at the cost of approximation bias, which may compromise the fidelity of UQ, particularly for unobserved data.

In contrast, MCMC methods, e.g., Gibbs sampling, provide asymptotically exact posterior estimates. For example, the BPMF ([Salakhutdinov and Mnih, 2008](#)) and Bayesian Gaussian CP decomposition (BGCP) model ([Chen et al., 2019b](#)) use Gibbs sampling to draw samples of model variables from the posteriors. In the cases that the likelihood is Gaussian and conjugate priors are employed, closed-form conditional posteriors can be derived for most of the model parameters in a factorization model. This enables the use of Gibbs sampling ([Gelman et al., 2013](#)), which iteratively samples parameters from the analytical conditionals. In practice, MCMC-based probabilistic factorization models can be computationally tractable and have been shown to converge efficiently even for large-scale traffic datasets ([Chen et al., 2019b; Lei et al., 2022b, 2024](#)). More importantly, MCMC yields more accurate posterior distributions and higher-quality uncertainty estimates than VI, which can be crucial for downstream decision-making tasks such as policy making and risk-sensitive forecasting.

Here we focus on the MCMC inference procedure. For details on variational posterior formulations for low-rank tensor factorization models, see [Chen et al. \(2019a\)](#). As an illustrative example, we describe the Gibbs sampling updates for the factorization model on a third-order traffic tensor as described in Eq. (30), with Gaussian likelihood and conjugate priors. For simplicity in this illustration, we assume the component weights λ are captured by the latent factors and fix $\lambda_r = 1$ to avoid identifiability issues. We refer to this basic model framework as **Bayesian probabilistic tensor factorization (BPTF)**. With the conjugate priors, the conditional posterior for each row of latent factors remains Gaussian. Specifically, for data tensor $\mathcal{Y} \in \mathbb{R}^{M \times T \times P}$ with observed entries \mathcal{Y}_Ω , the conditional posteriors for the m th row of $\mathbf{U} \in \mathbb{R}^{M \times R}$, the t th row of $\mathbf{V} \in \mathbb{R}^{T \times R}$, and the p th row of $\mathbf{W} \in \mathbb{R}^{P \times R}$ are:

$$p(\mathbf{u}_{m:} | \mathcal{Y}_\Omega, \mathbf{V}, \mathbf{W}, \tau, \Theta_u) = \mathcal{N}\left(\mathbf{u}_{m:} \mid \boldsymbol{\mu}_m^{u*}, (\boldsymbol{\Lambda}_m^{u*})^{-1}\right), m = 1, \dots, M, \\ \text{where } \boldsymbol{\mu}_m^{u*} = (\boldsymbol{\Lambda}_m^{u*})^{-1} \left(\tau \sum_{t,p:(m,t,p) \in \Omega} \mathbf{h}_{tp}^u y_{mtp} + \boldsymbol{\Lambda}_u \boldsymbol{\mu}_u \right), \boldsymbol{\Lambda}_m^{u*} = \tau \sum_{t,p:(m,t,p) \in \Omega} \mathbf{h}_{tp}^u (\mathbf{h}_{tp}^u)^\top + \boldsymbol{\Lambda}_u, \quad (31)$$

$$p(\mathbf{v}_{t:} | \mathcal{Y}_\Omega, \mathbf{U}, \mathbf{W}, \tau, \Theta_v) = \mathcal{N}\left(\mathbf{v}_{t:} \mid \boldsymbol{\mu}_t^{v*}, (\boldsymbol{\Lambda}_t^{v*})^{-1}\right), t = 1, \dots, T, \\ \text{where } \boldsymbol{\mu}_t^{v*} = (\boldsymbol{\Lambda}_t^{v*})^{-1} \left(\tau \sum_{m,p:(m,t,p) \in \Omega} \mathbf{h}_{mp}^v y_{mtp} + \boldsymbol{\Lambda}_v \boldsymbol{\mu}_v \right), \boldsymbol{\Lambda}_t^{v*} = \tau \sum_{m,p:(m,t,p) \in \Omega} \mathbf{h}_{mp}^v (\mathbf{h}_{mp}^v)^\top + \boldsymbol{\Lambda}_v, \quad (32)$$

$$p(\mathbf{w}_{p:} | \mathcal{Y}_\Omega, \mathbf{U}, \mathbf{V}, \tau, \Theta_w) = \mathcal{N}\left(\mathbf{w}_{p:} \mid \boldsymbol{\mu}_p^{w*}, (\boldsymbol{\Lambda}_p^{w*})^{-1}\right), p = 1, \dots, P, \\ \text{where } \boldsymbol{\mu}_p^{w*} = (\boldsymbol{\Lambda}_p^{w*})^{-1} \left(\tau \sum_{m,t:(m,t,p) \in \Omega} \mathbf{h}_{mt}^w y_{mtp} + \boldsymbol{\Lambda}_w \boldsymbol{\mu}_w \right), \boldsymbol{\Lambda}_p^{w*} = \tau \sum_{m,t:(m,t,p) \in \Omega} \mathbf{h}_{mt}^w (\mathbf{h}_{mt}^w)^\top + \boldsymbol{\Lambda}_w. \quad (33)$$

Here we define the Hadamard (element-wise) products: $\mathbf{h}_{tp}^u = \mathbf{v}_{t:} \circledast \mathbf{w}_{p:} \in \mathbb{R}^R$, $\mathbf{h}_{mp}^v = \mathbf{u}_{m:} \circledast \mathbf{w}_{p:} \in \mathbb{R}^R$, and $\mathbf{h}_{mt}^w = \mathbf{u}_{m:} \circledast \mathbf{v}_{t:} \in \mathbb{R}^R$. Intuitively, \mathbf{h}_{tp}^u is the element-wise product of the latent factors \mathbf{V} and \mathbf{W} associated

with the other two modes for entry (m, t, p) , which acts as a covariate when updating $\mathbf{u}_{m:}$, and similarly for \mathbf{h}_{mp}^v and \mathbf{h}_{mt}^w for updating \mathbf{v}_t and \mathbf{w}_p , respectively.

For convenience, let $\Theta_0 = \{\mu_0, \beta_0, \mathbf{W}_0, \nu_0\}$ denote the collection of hyper-hyperparameters. The conditional posteriors for model hyperparameters $\Theta_u = \{\mu_u, \Lambda_u\}$, $\Theta_v = \{\mu_v, \Lambda_v\}$, $\Theta_w = \{\mu_w, \Lambda_w\}$ (each consisting of a mean vector and precision matrix) remain conjugate Normal-Wishart distributions, summarized as below:

$$\Theta_u : p(\mu_u, \Lambda_u | \mathbf{U}, \Theta_0) = \mathcal{N}\left(\mu_u \mid \mu_0^{u*}, (\beta_0^{u*} \Lambda_u)^{-1}\right) \mathcal{W}(\Lambda_u | \mathbf{W}_0^{u*}, \nu_0^{u*}), \quad (34)$$

where

$$\begin{aligned} \mu_0^{u*} &= \frac{\beta_0 \mu_0 + M \bar{\mathbf{u}}}{\beta_0 + M}, \quad \beta_0^{u*} = \beta_0 + M, \quad \nu_0^{u*} = \nu_0 + M, \\ (\mathbf{W}_0^{u*})^{-1} &= \mathbf{W}_0^{-1} + M \mathbf{S}_u + \frac{\beta_0 M}{\beta_0 + M} (\bar{\mathbf{u}} - \mu_0) (\bar{\mathbf{u}} - \mu_0)^\top, \\ \bar{\mathbf{u}} &= \frac{1}{M} \sum_{m=1}^M \mathbf{u}_{m:}, \quad (\text{sample mean}) \\ \mathbf{S}_u &= \frac{1}{M} \sum_{m=1}^M (\mathbf{u}_{m:} - \bar{\mathbf{u}}) (\mathbf{u}_{m:} - \bar{\mathbf{u}})^\top, \quad (\text{sample covariance}) \\ \Theta_v : p(\mu_v, \Lambda_v | \mathbf{V}, \Theta_0) &= \mathcal{N}\left(\mu_v \mid \mu_0^{v*}, (\beta_0^{v*} \Lambda_v)^{-1}\right) \mathcal{W}(\Lambda_v | \mathbf{W}_0^{v*}, \nu_0^{v*}), \end{aligned} \quad (35)$$

where

$$\begin{aligned} \mu_0^{v*} &= \frac{\beta_0 \mu_0 + N \bar{\mathbf{v}}}{\beta_0 + N}, \quad \beta_0^{v*} = \beta_0 + T, \quad \nu_0^{v*} = \nu_0 + T, \\ (\mathbf{W}_0^{v*})^{-1} &= \mathbf{W}_0^{-1} + T \mathbf{S}_v + \frac{\beta_0 N}{\beta_0 + N} (\bar{\mathbf{v}} - \mu_0) (\bar{\mathbf{v}} - \mu_0)^\top, \\ \bar{\mathbf{v}} &= \frac{1}{T} \sum_{t=1}^T \mathbf{v}_{t:}, \quad \mathbf{S}_v = \frac{1}{T} \sum_{t=1}^T (\mathbf{v}_{t:} - \bar{\mathbf{v}}) (\mathbf{v}_{t:} - \bar{\mathbf{v}})^\top, \end{aligned}$$

and

$$\Theta_w : p(\mu_w, \Lambda_w | \mathbf{W}, \Theta_0) = \mathcal{N}\left(\mu_w \mid \mu_0^{w*}, (\beta_0^{w*} \Lambda_w)^{-1}\right) \mathcal{W}(\Lambda_w | \mathbf{W}_0^{w*}, \nu_0^{w*}), \quad (36)$$

where

$$\begin{aligned} \mu_0^{w*} &= \frac{\beta_0 \mu_0 + P \bar{\mathbf{w}}}{\beta_0 + P}, \quad \beta_0^{w*} = \beta_0 + P, \quad \nu_0^{w*} = \nu_0 + P, \\ (\mathbf{W}_0^{w*})^{-1} &= \mathbf{W}_0^{-1} + P \mathbf{S}_w + \frac{\beta_0 P}{\beta_0 + P} (\bar{\mathbf{w}} - \mu_0) (\bar{\mathbf{w}} - \mu_0)^\top, \\ \bar{\mathbf{w}} &= \frac{1}{P} \sum_{p=1}^P \mathbf{w}_{p:}, \quad \mathbf{S}_w = \frac{1}{P} \sum_{p=1}^P (\mathbf{w}_{p:} - \bar{\mathbf{w}}) (\mathbf{w}_{p:} - \bar{\mathbf{w}})^\top. \end{aligned}$$

For the noise precision τ , the posterior distribution is still a Gamma distribution:

$$p(\tau | \mathcal{Y}_\Omega, \mathbf{U}, \mathbf{V}, \mathbf{W}, \theta_\tau) = \text{Gamma}(a^*, b^*), \quad (37)$$

where we define $\theta_\tau = \{a_0, b_0\}$, $a^* = \frac{1}{2}|\Omega| + a_0$, and $b^* = \frac{1}{2} \sum_{(m,t,p) \in \Omega} (y_{mtp} - \sum_{r=1}^R u_{mr} v_{tr} w_{pr})^2 + b_0$.

Model implementation. The predictive distribution for y_{mtp} at unobserved locations, i.e., $(m, t, p) \in \Omega^c$, where Ω^c denotes the complementary set of Ω , can be obtained by marginalizing over model parameters and hyperparameters:

$$\begin{aligned} p(y_{mtp}^* | \mathcal{Y}_\Omega, \Theta_0, \theta_\tau) &= \int \int p(y_{mtp}^* | \mathbf{u}_{m:}, \mathbf{v}_{t:}, \mathbf{w}_{p:}, \tau) p(\mathbf{U}, \mathbf{V}, \mathbf{W}, \tau | \mathcal{Y}_\Omega, \Theta_u, \Theta_v, \Theta_w, \theta_\tau) \\ &\quad p(\Theta_u, \Theta_v, \Theta_w | \Theta_0) d\{\mathbf{U}, \mathbf{V}, \mathbf{W}, \tau\} d\{\Theta_u, \Theta_v, \Theta_w\}. \end{aligned} \quad (38)$$

The exact predictive distribution is analytically intractable. For MCMC inference based methods, the prediction distribution for unobserved data can be approximated with Monte Carlo estimation. Assuming one

Algorithm 1: Gibbs sampling for BPTF (Bayesian probabilistic tensor factorization).

Input: $\mathcal{Y}_\Omega, K_1, K_2$.

Output: Completed $\hat{\mathcal{Y}}$.

Initialize model parameters, including latent factor matrices $\{\mathbf{U}, \mathbf{V}, \mathbf{W}\}$ as normally distributed random variables, model noise precision $\tau = 1$, $\Theta_0 : \{\beta_0 = 1, \nu_0 = R, \mathbf{W}_0 = \mathbf{I}_R, \boldsymbol{\mu}_0 = \text{zeros}(R, 1)\}$, and $\theta_\tau : \{a_0 = 1e-6, b_0 = 1e-6\}$.

for $k = 1 : K_1 + K_2$ **do**

Draw model hyperparameters $\Theta_u^{(k)}, \Theta_v^{(k)}, \Theta_w^{(k)}$ following Eqs. (34), (35), (36), respectively;

for $m = 1 : M$ **do**

Draw the m th row of \mathbf{U} , i.e., $\mathbf{u}_{m:}^{(k)}$, following Eq. (31);

for $t = 1 : T$ **do**

Draw the t th row of \mathbf{V} , i.e., $\mathbf{v}_{t:}^{(k)}$, following Eq. (32);

for $p = 1 : P$ **do**

Draw the p th row of \mathbf{W} , i.e., $\mathbf{w}_{p:}^{(k)}$, following Eq. (33);

Draw model noise precision $\tau^{(k)}$ following Eq. (37);

if $k > K_1$ **then**

Compute $\hat{\mathcal{Y}}^{(k)}$ based on $\mathbf{U}^{(k)}, \mathbf{V}^{(k)}, \mathbf{W}^{(k)}, \tau^{(k)}$;

return $\{\hat{\mathcal{Y}}^{(k)}\}_{k=K_1+1}^{K_1+K_2}$.

perform an MCMC chain for $K_1 + K_2$ iterations, treating the first K_1 as burn-in and collecting the last K_2 samples for estimation. The predictive distribution for y_{mtp}^* can be estimated by averaging the predictions from the K_2 posterior samples:

$$p(y_{mtp}^* \mid \mathcal{Y}_\Omega, \Theta_0, \theta_\tau) \approx \frac{1}{K_2} \sum_{k=K_1+1}^{K_1+K_2} p(y_{mtp}^* \mid \mathbf{u}_{m:}^{(k)}, \mathbf{v}_{t:}^{(k)}, \mathbf{w}_{p:}^{(k)}, \tau^{(k)}). \quad (39)$$

where $k = 1 : K_1 + K_2$ denotes the number of MCMC iteration. The implementation algorithm for BPTF is summarized in Algorithm 1, including initialization, iterative updates for each row of $\mathbf{U}, \mathbf{V}, \mathbf{W}$, and accumulating the predictions $\hat{\mathcal{Y}}^{(k)}$ after burn-in.

Application to traffic data. Bayesian tensor factorization methods such as BGCP (Bayesian Gaussian CP decomposition) (Chen et al., 2019a,b) and its variants have been successfully applied to real traffic data tasks including missing data imputation, pattern recognition, and event (anomaly) detection (Zhu et al., 2022). For instance, Qi et al. (2023) apply BGCP to traffic tensors augmented with spatial and temporal features to detect traffic accidents. These studies underscore the practical impact of Bayesian low-rank modeling for ITS applications where uncertainty estimation is critical.

4.1.2. Spatiotemporal modeling

The fundamental Bayesian low-rank factorization methods described above can be effective for capturing core patterns of traffic data and performing tasks such as traffic data imputation. However, to model **temporal dynamics** and **spatial correlations**, and to tackle more complicated tasks such as prediction and spatial extrapolation, i.e., kriging, additional spatiotemporal modeling structure needs to be incorporated into the factorization models. We next introduce how to explicitly encode spatiotemporal correlations in Bayesian low-rank tensor factorization.

Bayesian temporal factorization models. To capture temporal dynamics, one way is to integrate time-series models into the tensor factorization. Xiong et al. (2010) proposed a Bayesian probabilistic temporal factorization (also known as Bayesian temporal collaborative filtering, TCF) model for time-evolving matrix/tensor data. In their model, a state-space model, i.e., an AR process with order being 1, is imposed on the time-mode latent factors. For example, consider a third-order user rating tensor $\mathcal{Y} \in \mathbb{R}^{M \times T \times P}$ with the three modes representing *user*, *time*, and *item*, respectively. To perform time-varying collaborative filtering, they approximate \mathcal{Y} with a CP factorization $\mathcal{Y} \approx \sum_{r=1}^R \mathbf{u}_{:r} \circ \mathbf{v}_{:r} \circ \mathbf{w}_{:r} + \mathcal{E}$ as usual, but instead of

independent priors on the temporal latent factors $\mathbf{V} \in \mathbb{R}^{T \times R}$, they assume an AR(1) process on each row vector of it, i.e., $\mathbf{v}_{t:}$:

$$\begin{aligned}\mathbf{v}_{t:} &\sim \mathcal{N}(\mathbf{v}_{t-1:}, \boldsymbol{\Lambda}_v^{-1}), \quad t = 2, \dots, T, \\ \mathbf{v}_{1:} &\sim \mathcal{N}(\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v^{-1}), \quad t = 1.\end{aligned}\tag{40}$$

Here $\mathbf{v}_{1:}$ denotes an initial prior, and $\boldsymbol{\Lambda}_v$ is a precision matrix governing the auto-regressive transition. The observation model and priors on \mathbf{U} and \mathbf{V} are similar to BPTF, i.e., Gaussian, and conjugate priors are placed on all hyperparameters. TCF effectively couples consecutive time steps in the latent space, allowing the latent factors to evolve smoothly over time. Based on the conjugate prior settings, a Gibbs sampling algorithm is derived for model inference.

[Chen and Sun \(2021\)](#) further extended this idea in their **Bayesian temporal tensor factorization (BTTF)** model for multidimensional traffic data, which uses a higher-order AR process, i.e., a vector AR (VAR) model, on the temporal latent factors. They applied BTTF to traffic speed and flow data for imputation and short-term forecasting. For example, consider a third-order traffic tensor $\mathbf{Y} \in \mathbb{R}^{M \times T \times P}$ for e.g., M origin locations, T time intervals, and P destination locations. BTTF factorizes \mathbf{Y} as $\mathbf{Y} \approx \sum_{r=1}^R \mathbf{u}_{:r} \circ \mathbf{v}_{:r} \circ \mathbf{w}_{:r} + \mathbf{\epsilon}$ similarly, but assumes each row vector of the temporal factor matrix \mathbf{V} follows a VAR(d) process. Defining a set of time lags $\mathcal{L} = \{\ell_1, \dots, \ell_d\}$, the t th row vector of \mathbf{V} , i.e., $\mathbf{v}_{t:} \in \mathbb{R}^R$, is modeled with:

$$\mathbf{v}_{t:} = \sum_{k=1}^d \mathbf{A}_k \mathbf{v}_{t-\ell_k:} + \mathbf{g}_t, \quad t \notin \mathcal{L},\tag{41}$$

where $\mathbf{A}_k \in \mathbb{R}^{R \times R}$ is coefficient matrix for lag ℓ_k ($k = 1, \dots, d$), and $\mathbf{g}_t \in \mathbb{R}^R$ is a noise term that assumed following $\mathbf{g}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_g)$. Letting $\mathbf{B} = [\mathbf{A}_1, \dots, \mathbf{A}_d]^\top \in \mathbb{R}^{(Rd) \times R}$ and $\mathbf{x}_t = [\mathbf{v}_{t-\ell_1:}^\top, \dots, \mathbf{v}_{t-\ell_d:}^\top]^\top \in \mathbb{R}^{(Rd) \times 1}$, we can write $\mathbf{v}_{t:}$ by $\mathbf{v}_{t:} = \mathbf{B}^\top \mathbf{x}_t + \mathbf{g}_t$. The VAR process on the temporal factor matrix \mathbf{V} can then be described as:

$$\mathbf{v}_{t:} \sim \begin{cases} \mathcal{N}(\mathbf{0}, \mathbf{I}_R), & t \in \mathcal{L} \text{ (initial lagged time steps)} \\ \mathcal{N}(\mathbf{B}^\top \mathbf{x}_t, \boldsymbol{\Sigma}_g), & \text{otherwise.} \end{cases}\tag{42}$$

They place conjugate matrix-normal-inverse-Wishart (MNIW) hyperprior on the VAR coefficient matrix \mathbf{B} and the noise covariance matrix $\boldsymbol{\Sigma}_g$ as below:

$$\mathbf{B} \sim \mathcal{MN}_{(Rd) \times R}(\mathbf{M}_0, \boldsymbol{\Psi}_0, \boldsymbol{\Sigma}_g), \quad \boldsymbol{\Sigma}_g \sim \mathcal{IW}(\mathbf{S}_0, \nu_0),\tag{43}$$

where $\mathcal{MN}(\cdot)$ is the matrix-normal distribution and \mathcal{IW} is the inverse-Wishart, $\boldsymbol{\Psi}_0 \in \mathbb{R}^{(Rd) \times (Rd)}$ and $\boldsymbol{\Sigma}_g \in \mathbb{R}^{R \times R}$ are covariance matrices, $\mathbf{M}_0 \in \mathbb{R}^{(Rd) \times R}$ is the prior mean matrix, and $\mathbf{S}_0 \in \mathbb{R}^{R \times R}$ is the scale matrix of the Wishart distribution.

They derive a Gibbs sampling algorithm for BTTF, where the posteriors for all model parameters except the VAR related variables resemble those in BPTF, and the VAR parameters $(\mathbf{B}, \boldsymbol{\Sigma}_g)$ have a conjugate posterior. Given the MNIW prior in Eq. (43), the posterior remains MNIW with updated hyperparameters:

$$p(\mathbf{B}, \boldsymbol{\Sigma}_g | -) = \mathcal{MN}(\mathbf{M}^*, \boldsymbol{\Psi}^*, \boldsymbol{\Sigma}_g) \times \mathcal{IW}(\mathbf{S}^*, \nu^*),\tag{44}$$

where

$$\begin{aligned}\boldsymbol{\Psi}^* &= (\boldsymbol{\Psi}_0^{-1} + \mathbf{Z}^\top \mathbf{Z})^{-1}, \\ \mathbf{M}^* &= \boldsymbol{\Psi}^* (\boldsymbol{\Psi}_0^{-1} \mathbf{M}_0 + \mathbf{Z}^\top \mathbf{X}), \\ \mathbf{S}^* &= \mathbf{S}_0 + \mathbf{X}^\top \mathbf{X} + \mathbf{M}_0^\top \boldsymbol{\Psi}_0^{-1} \mathbf{M}_0 - (\mathbf{M}^*)^\top (\boldsymbol{\Psi}^*)^{-1} \mathbf{M}^*, \\ \nu^* &= \nu_0 + T - \ell_d,\end{aligned}\tag{45}$$

with

$$\mathbf{Z} = \begin{bmatrix} \mathbf{v}_{\ell_d+1:}^\top \\ \vdots \\ \mathbf{v}_{T:}^\top \end{bmatrix} \in \mathbb{R}^{(T-\ell_d) \times R}, \text{ and } \mathbf{X} = \begin{bmatrix} \mathbf{x}_{\ell_d+1}^\top \\ \vdots \\ \mathbf{x}_T^\top \end{bmatrix} \in \mathbb{R}^{(T-\ell_d) \times (Rd)}.$$

In summary, incorporating an AR or VAR structure on the temporal latent factors, these Bayesian models can capture temporal dependencies in traffic data more explicitly, facilitating improved temporal predictions with quantified uncertainty. Similar VAR/dynamic prior regularized Bayesian temporal factorization has been used for multi-step crime prediction ([Liang et al., 2025](#)) and time series analysis ([Zhang, 2024](#)), among others. Furthermore, BTTF has been applied with dynamic moving/sliding windows for near-real-time forecasting ([Ren et al., 2021](#)).

Bayesian kernelized low-rank models. Another way to introduce structured correlations into low-rank models is by placing Gaussian process (GP) priors on columns of the latent factors, rather than using simple Gaussian priors on rows of them. This type of methods correspond to the covariance norm regularized tensor decomposition approaches (Lei and Sun, 2024) introduced in Section 3.1.2. GPs are widely used in statistics and machine learning for modeling spatial, temporal, and other structured dependencies (Rasmussen and Williams, 2006; Banerjee et al., 2008, 2014). In geostatistics, for example, low-rank factor models with GP priors have been explored for modeling spatiotemporal data (Ren and Banerjee, 2013). In machine learning, Zhou et al. (2012) proposed the kernelized probabilistic MF (KPMF) model, which uses GP priors on column vectors of latent factors to capture smooth correlations (we have discussed this model in Section 3.1.2, see Eq. (14)). Additionally, Luttinen and Ilin (2009) developed a VI (variational inference) solution for a similar GP-regularized MF model. Such GP-enhanced low-rank approaches were introduced to traffic data analysis in the Bayesian kernelized MF/tensor regression models by Lei et al. (2022b, 2024), where fully Bayesian inference algorithms based on MCMC are developed.

To illustrate, consider again a third-order *location* \times *time of day* \times *day* traffic speed tensor \mathcal{Y} of size $M \times T \times P$. One can extend the Bayesian kernelized MF (BKMF) model in Lei et al. (2022b) to a tensor form and formulate a **Bayesian kernelized tensor factorization** (BKTf) model (also refer to the global component modeling in Lei et al. (2022a)), where the tensor is approximated as $\mathcal{Y} \approx \sum_{r=1}^R \mathbf{u}_{::r} \circ \mathbf{v}_{::r} \circ \mathbf{w}_{::r} + \mathcal{E}$ with \mathcal{E} being i.i.d. noise as before. The key difference is in the priors for \mathbf{U} and \mathbf{V} : instead of independent Gaussian priors on each row, each column of \mathbf{U} and \mathbf{V} , i.e., each latent factor vector across one mode, is drawn from a zero-mean GP. The model specification for the latent factors \mathbf{U} and \mathbf{V} can be described as:

$$\begin{aligned}\mathbf{u}_{::r} &\sim \mathcal{GP}(\mathbf{0}, \mathbf{K}_r^u), \quad r = 1, \dots, R, \\ \mathbf{v}_{::r} &\sim \mathcal{GP}(\mathbf{0}, \mathbf{K}_r^v), \quad r = 1, \dots, R,\end{aligned}\tag{46}$$

where $\mathbf{u}_{::r} \in \mathbb{R}^M$ and $\mathbf{v}_{::r} \in \mathbb{R}^T$ are the r th column of latent factors \mathbf{U} and \mathbf{V} , respectively, $\mathbf{K}_r^u \in \mathbb{R}^{M \times M}$ and $\mathbf{K}_r^v \in \mathbb{R}^{T \times T}$ are positive semi-definite (PSD) covariance matrices computed from chosen kernel functions for the spatial and temporal dimensions, respectively. For instance, one can select $\mathbf{K}_{ij}^u = k_s(s_i, s_j; \phi_u)$ for $i, j \in [1, M]$ and $\mathbf{K}_{ij}^v = k_t(t_i, t_j; \phi_v)$ for $i, j \in [1, T]$, where k_s can be a graph diffusion kernel defined on a road network (Smola and Kondor, 2003) or a Matérn kernel over spatial coordinates (s_i, s_j) with kernel hyperparameters ϕ_u , and k_t can be a Matérn or periodic kernel for daily/weekly temporal patterns with inputs (t_i, t_j) and hyperparameters ϕ_v (Rasmussen and Williams, 2006). In practice, one can assume the same form of kernel function with the same kernel hyperparameters for all R latent components for better identifiability and simplicity. These covariance functions encode prior knowledge of smoothness, spatial network structure, or temporal periodicity into the factorization. For the third dimension, i.e., the *day* mode, which may not have strong correlations and a proper kernel function may be lacked, one can retain a Gaussian prior: $\mathbf{w}_{::r} \sim \mathcal{N}(\mathbf{0}_{P \times 1}, \Lambda_w'^{-1} \in \mathbb{R}^{P \times P})$, where $r = 1, \dots, R$.

To complete the Bayesian specification, hyperpriors are also placed on kernel hyperparameters such as kernel length-scales and variances. For better identifiability, one can assume the weights are absorbed in \mathbf{W} and fix the kernel variances as 1, then the kernel hyperparameters become length-scales of kernel functions for $\mathbf{u}_{::r}$ and $\mathbf{v}_{::r}$, denoting by $\{\phi_r^u, \phi_r^v\}_{r=1}^R$ (Lei et al., 2022a, 2024). In the models by Lei et al. (2022b,a), they assume the logarithm of kernel hyperparameters follow a Normal distribution to ensure the positivity:

$$\begin{aligned}\log(\phi_r^u) &\sim \mathcal{N}(\mu_\phi, \tau_\phi^{-1}), \\ \log(\phi_r^v) &\sim \mathcal{N}(\mu_\phi, \tau_\phi^{-1}).\end{aligned}\tag{47}$$

With these priors in place and conjugate priors for any other parameters as before, full Bayesian inference algorithms are developed.

The conditional posteriors of the latent factor vectors $\mathbf{u}_{::r}, \mathbf{v}_{::r}, \mathbf{w}_{::r}$ are still Gaussian and MCMC sampling can update each column as in a regression problem conditioned on the other parameters. Defining a binary mask tensor $\mathcal{O} \in \mathbb{R}^{M \times T \times P}$ for the observed entries in the data tensor \mathcal{Y} , where $o_{mtp} = 1$ if $(m, t, p) \in \Omega$ and 0 otherwise. For $r = 1, \dots, R$, define $\mathcal{Y}_r = \mathcal{Y} - \sum_{h=1, h \neq r}^R \mathbf{u}_{::h} \circ \mathbf{v}_{::h} \circ \mathbf{w}_{::h}$. For $d = 1, \dots, 3$, let $\mathcal{O}_{(d)}$ be the mode- d unfolding of \mathcal{O} , $\mathcal{O}_d \in \mathbb{R}^{|\Omega| \times (MTP)}$ be a binary selection matrix formed by removing the rows corresponding to zero values in $\text{vec}(\mathcal{O}_{(d)})$ from a $MTP \times MTP$ identity matrix, and $\mathbf{Y}_{r,(d)}$ denote the mode- d unfolding of \mathcal{Y}_r . The column-wise conditional posterior updates of the latent factor matrices

Algorithm 2: Slice sampling for kernel hyperparameter ϕ_r^u .

Input: $\phi_r^u, \mathbf{V}, \mathbf{W}, \mathbf{y}_\Omega, \tau$.

Output: next ϕ_r^u .

Initialize slice sampling scale $\gamma = \log(10)$.

Compute $\mathbf{K}_r^u, \Sigma_{\mathbf{y}_{\Omega,r}|\phi_r^u}, \log p(\phi_r^u | \mathbf{y}_{\Omega,r}, \tau)$ corresponding to ϕ_r^u based on Eqs. (49) and (50);

Calculate the sampling range: $\delta \sim \text{Uniform}(0, \gamma)$, $\phi_{\min} = \phi_r^u - \delta$, $\phi_{\max} = \phi_{\min} + \gamma$;

Draw $\kappa \sim \text{Uniform}(0, 1)$;

while True **do**

Draw proposal $\phi' \sim \text{Uniform}(\phi_{\min}, \phi_{\max})$;

Compute $(\mathbf{K}_r^u)', \Sigma_{\mathbf{y}_{\Omega,r}|\phi'}, \log p(\phi' | \mathbf{y}_{\Omega,r}, \tau)$ corresponding to ϕ' based on Eqs. (49) and (50);

if $\exp(\log p(\phi' | \mathbf{y}_{\Omega,r}, \tau) - \log p(\phi_r^u | \mathbf{y}_{\Omega,r}, \tau)) > \kappa$ **then**

return ϕ' as next ϕ_r^u ;

break;

else if $\phi' < \phi$ **then**

$\phi_{\min} = \phi'$;

else

$\phi_{\max} = \phi'$.

for $r = 1, \dots, R$ are summarized as below:

$$\begin{aligned}
 p(\mathbf{u}_{:r} | -) &= \mathcal{N}\left(\mathbf{u}_{:r} \mid \boldsymbol{\mu}_r^{u*}, (\Lambda_r^{u*})^{-1}\right), \\
 \text{where } \boldsymbol{\mu}_r^{u*} &= \tau(\Lambda_r^{u*})^{-1} (\mathbf{Y}_{r,(1)} \circledast \mathbf{O}_{(1)}) (\mathbf{w}_{:r} \otimes \mathbf{v}_{:r}), \\
 \Lambda_r^{u*} &= \tau(\mathbf{H}_r^u)^\top \mathbf{H}_r^u + (\mathbf{K}_r^u)^{-1}, \text{ with } \mathbf{H}_r^u = \mathbf{O}_1 ((\mathbf{w}_{:r} \otimes \mathbf{v}_{:r}) \otimes \mathbf{I}_M), \\
 p(\mathbf{v}_{:r} | -) &= \mathcal{N}\left(\mathbf{v}_{:r} \mid \boldsymbol{\mu}_r^{v*}, (\Lambda_r^{v*})^{-1}\right), \\
 \text{where } \boldsymbol{\mu}_r^{v*} &= \tau(\Lambda_r^{v*})^{-1} (\mathbf{Y}_{r,(2)} \circledast \mathbf{O}_{(2)}) (\mathbf{w}_{:r} \otimes \mathbf{u}_{:r}), \\
 \Lambda_r^{v*} &= \tau(\mathbf{H}_r^v)^\top \mathbf{H}_r^v + (\mathbf{K}_r^v)^{-1}, \text{ with } \mathbf{H}_r^v = \mathbf{O}_2 ((\mathbf{w}_{:r} \otimes \mathbf{u}_{:r}) \otimes \mathbf{I}_N), \\
 p(\mathbf{w}_{:r} | -) &= \mathcal{N}\left(\mathbf{w}_{:r} \mid \boldsymbol{\mu}_r^{w*}, (\Lambda_r^{w*})^{-1}\right), \\
 \text{where } \boldsymbol{\mu}_r^{w*} &= \tau(\Lambda_r^{w*})^{-1} (\mathbf{Y}_{r,(3)} \circledast \mathbf{O}_{(3)}) (\mathbf{v}_{:r} \otimes \mathbf{u}_{:r}), \\
 \Lambda_r^{w*} &= \tau(\mathbf{H}_r^w)^\top \mathbf{H}_r^w + \Lambda_w'^{-1}, \text{ with } \mathbf{H}_r^w = \mathbf{O}_3 ((\mathbf{v}_{:r} \otimes \mathbf{u}_{:r}) \otimes \mathbf{I}_P).
 \end{aligned} \tag{48}$$

In terms of model hyperparameters, note that the kernel hyperparameters do not have analytical solutions, and in BKMF/BKTR models (Lei et al., 2022b, 2024), they use slice sampling to update kernel hyperparameters based on the marginal posteriors. For example, the marginal likelihood of ϕ_r^u can be written as:

$$\log p(\mathbf{y}_{\Omega,r} | \phi_r^u) \propto -\frac{1}{2} \mathbf{y}_{\Omega,r}^\top \Sigma_{\mathbf{y}_{\Omega,r}|\phi_r^u}^{-1} \mathbf{y}_{\Omega,r} - \frac{1}{2} \log |\Sigma_{\mathbf{y}_{\Omega,r}|\phi_r^u}|, \tag{49}$$

where $\mathbf{y}_{\Omega,r} = \mathbf{O}_1 \text{vec}(\mathbf{Y}_r) \in \mathbb{R}^{|\Omega|}$ and $\Sigma_{\mathbf{y}_{\Omega,r}|\phi_r^u} = \mathbf{H}_r^u \mathbf{K}_r^u (\mathbf{H}_r^u)^\top + \tau^{-1} \mathbf{I}_{|\Omega|}$. The logarithm of marginal posterior of ϕ_r^u becomes:

$$\log p(\phi_r^u | \mathbf{y}_{\Omega,r}, \tau) \propto \log p(\phi_r^u) + \log p(\mathbf{y}_{\Omega,r} | \phi_r^u). \tag{50}$$

The slice sampling algorithm for ϕ_r^u is provided in Algorithm 2. The full MCMC sampling procedure and estimation processes refer to Lei et al. (2022a, 2024).

Overall, Bayesian kernelized factorization methods integrate tensor factorization with GP modeling in a fully probabilistic Bayesian setting. This combination enables the models to capture complex correlation structures via kernel functions, while still benefiting from the data efficiency of low-rank representations. Covariance functions serve as a flexible tool for modeling spatiotemporal dependencies, and Bayesian kernelized factorization can be viewed as a generalization of various Bayesian regularized factorization approaches through the use of specific kernel choices (Lei et al., 2022b). BKTF has been successfully applied to traffic data imputation and extrapolation tasks, such as kriging with missing values (Lei et al., 2022b,

2024; Lei, 2024; Lanthier et al., 2023). In addition, recently other formats of factorization models have also incorporated GP as priors for smoothness and capturing data correlations, paralytically tensor ring decomposition (Huang et al., 2025b).

Bayesian complementary kernelized learning. While Bayesian tensor factorization models with GP priors (e.g., BKMF, BKTf (Lei et al., 2022b,a)) effectively capture global spatiotemporal structures, they may require a relatively large rank R to model local, high-frequency variations in traffic data. In practice, the residuals from a low-rank model can exhibit structured patterns and spatiotemporal correlations that violate the i.i.d. noise assumption, especially for real-world traffic data where abrupt events such as accidents etc. can cause spikes that a smooth low-rank model cannot represent. Simply increasing the rank to fit these fine-scale effects risks overfitting and high computational costs (Lei et al., 2022a; Lei and Sun, 2024).

To address this issue, Lei et al. (2022a) proposed the **Bayesian complementary kernelized learning (BCKL)** framework. BCKL enhances the expressiveness of Bayesian tensor factorization by decomposing the data tensor into an additive combination of a global low-rank component and a local residual component. Formally, for the third-order traffic tensor $\mathcal{Y} \in \mathbb{R}^{M \times T \times P}$, BCKL estimates it with:

$$\mathcal{Y} \approx \underbrace{\sum_{r=1}^R \mathbf{u}_{:,r} \circ \mathbf{v}_{:,r} \circ \mathbf{w}_{:,r}}_{\mathcal{M}} + \mathcal{R} + \mathcal{E}, \quad (51)$$

where \mathcal{M} is a low-rank tensor capturing global trends with GP priors on factors as in BKTf, \mathcal{R} is a structured residual tensor capturing localized deviations, and \mathcal{E} denotes a remaining white noise process. The residual component \mathcal{R} is modeled as the sum of Q sparse GPs, each constructed using a product kernel composed of compactly supported kernel functions defined over each dimension. This formulation ensures computational efficiency for large-scale data, meanwhile effectively accounts for the structured high-frequency residual patterns. By explicitly modeling \mathcal{R} , BCKL can capture fine-grained traffic variations that would otherwise require a very high-rank tensor decomposition, thereby improving accuracy without sacrificing efficiency or interpretability.

This plug-and-play two-component approach allows decoupling of global and local effects: \mathcal{M} explains the major low-rank signal such as the overall diurnal patterns across the road network, while \mathcal{R} explains localized phenomena, e.g., a traffic spike in a specific road segment due to an accident). As a result, BCKL can achieve better overall reconstruction accuracy and provide well-calibrated uncertainty estimates for both the global trend and local variations (Lei et al., 2022a). Similar ideas of modeling structured residuals have been applied in other works for traffic data analysis. For example, Li et al. (2022) introduced a Bayesian neural tensor factorization that includes an auxiliary neural network module to learn complicated residual patterns beyond the core low-rank structure.

BCKL has been applied to traffic speed data imputation under extremely high missing rates, showing consistent improvements over standard Bayesian tensor models and standalone GP baselines in both mean and uncertainty estimations (Lei et al., 2022a). By jointly capturing low-rank spatiotemporal trends and structured local noise, BCKL is particularly effective in real-world scenarios where disruptions/variations such as accidents induce complex, localized, and spatiotemporal correlated effects in the data. The inclusion of the GP-based residual component yields improved predictive uncertainties for these irregular events, enhancing the model utility for risk-aware decision-making in ITS.

To illustrate the conceptual differences across Bayesian tensor models, we compare the assumptions and graphical models of representative Bayesian factorization approaches in Figure 7, including: (a) **BPTF** (Bayesian Probabilistic Tensor Factorization) with independent priors, (b) **BTTF** (Bayesian Temporal Tensor Factorization) with tempral AR priors, (c) **BKTf** (Bayesian Kernelized Tensor Factorization) with GP priors on factors, and (d) **BCKL** (Bayesian Complementary Kernelized Learning) with an added GP residual component.

(4) *Other aspects.* Beyond the choices of factor priors, several additional aspects have been explored in Bayesian LRTL for traffic data:

- **Automatic rank determination:** Rather than fixing the tensor rank R a priori, hierarchical shrinkage priors can be used to infer an appropriate rank from data. One popular approach is to use the multiplicative gamma process (MGP) prior, which can adaptively shrink unnecessary components. For example, Huang et al. (2024a) developed a Bayesian automatic CP decomposition (BACP) model that employs an MGP prior to gradually prune redundant factors, effectively performing automatic rank

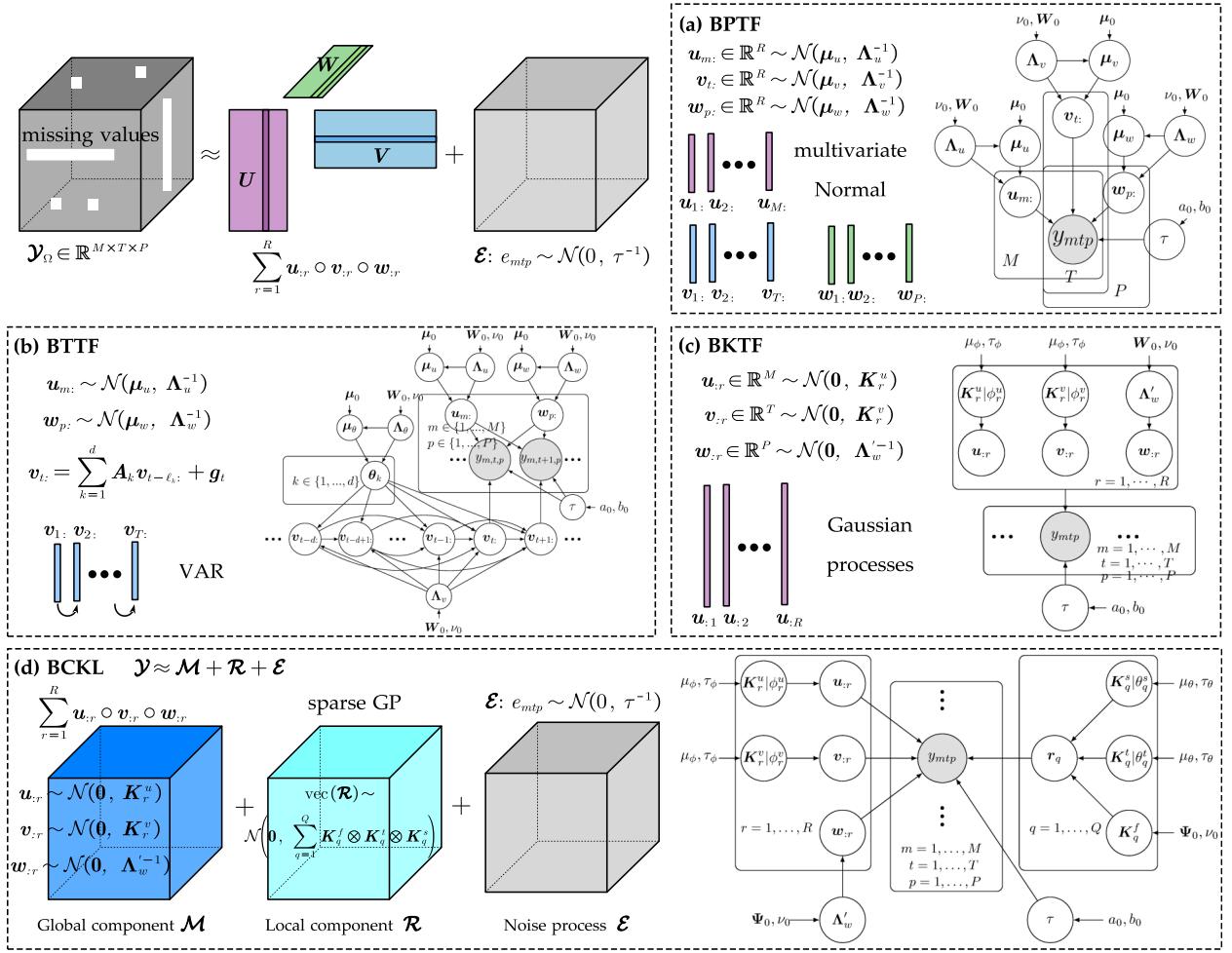


Figure 7: Comparison of Bayesian probabilistic factorization methods on a third-order tensor $\mathbf{Y}_\Omega \in \mathbb{R}^{M \times T \times P}$. We illustrate the prior assumptions and graphical models of (a) BPTF; (b) BTTF; (c) BKTF; and (d) BCKL.

selection. In addition, another work by Zhao et al. (2015) used ARD-type mechanisms for adaptive rank determination in a Bayesian CP model.

- **Modeling heterogeneous noise:** Most factorization models assume homoskedastic, i.e., constant-variance, Gaussian noise. However, traffic data can exhibit heteroscedasticity, meaning that the noise variance/precision can differ across time, location, or other contexts. Recent approaches (Chen and Sun, 2021; Lei et al., 2022b) allow for non-identically distributed noise, for example by introducing a pre-entry noise precision τ_{mtp} in the model for a $M \times T \times P$ data. These per-entry precisions can be assumed hierarchical priors and inferred along with the factors, providing a flexible and robust modeling of varying noise levels.
- **Neural network integration:** To increase model flexibility, some studies integrate deep neural network components into the Bayesian tensor framework. For instance, Deng et al. (2021b) applied a recurrent neural network (RNN) to model complex temporal dynamics within a generative tensor model. Li et al. (2022) combined a low-rank Bayesian tensor factorization with a neural residual module, thereby capturing nonlinear relationships that a linear low-rank tensor structure might overlook. These hybrid models aim to improve the performance of tensor methods with the expressiveness of neural nets.
- **Extended modeling tasks:** Bayesian low-rank tensor factorization methods have been extended beyond imputation and prediction/forecasting to supervised, decision-making, and other settings. Examples include tensor regression, which incorporates covariates into factor models for supervised learning (Lei et al., 2024), Bayesian optimization (BO) using BKTF as surrogate (Lei and Sun, 2023), and causal discovery (Zhang et al., 2021; Fan et al., 2022). A comprehensive discussion of Bayesian

kernelized low-rank modeling including BKMF (Lei et al., 2022b), BKTR (Lei et al., 2024), BCKL (Lei et al., 2022a), and BKTf for BO (Lei and Sun, 2023) can be found in Lei (2024). Some works also incorporated other structural constraints in probabilistic low-rank frameworks to enhance identifiability, interpretability, and model performance, such as orthogonal probabilistic factor models (Cheng et al., 2016; Zhou and Zhang, 2025) and Bayesian nonnegative factorization (Schmidt et al., 2009; Cemgil, 2009; Yang et al., 2025c).

Moreover, several Bayesian tensor methods developed for general spatiotemporal data are directly applicable to traffic data analysis, even though the original applications were outside transportation. For example, Bayesian low-rank factorization with graph smoothness (Li et al., 2023c; Chen et al., 2023b), Bayesian Tucker decomposition with core modeling (Tong et al., 2023), and Bayesian TR formulations (Long et al., 2021; Huang et al., 2024b).

(5) *Summary.* A summary and comparison of representative Bayesian tensor factorization models applied to traffic data are presented in Table 6, organized chronologically by publication year. We see that

- Most Bayesian LRTL methods use CP factorization since it is simple yet expressive and makes it relatively easy to incorporate spatiotemporal correlations/structure through priors on latent factors. For forecasting, adding temporal dynamics, e.g., AR/VAR priors, is often necessary. To support extrapolation/interpolation, covariance-based (GP regularized) smoothness priors can be effective. Recently, more complex (higher-parameter) decompositions have been explored, such as tubal rank models (Zhao et al., 2024a) and tensor ring (TR) factorizations (Liu et al., 2025a), in which sparse priors are commonly incorporated to control overparameterization.
- The factorized representation yields interpretable latent factors that are well-suited to pattern discovery and supervised tasks requiring interpretability, e.g., regression. In Bayesian formulations, posterior distributions over latent factors quantify latent-space uncertainty, enabling explanations of mode-wise patterns with credible intervals. For example, in tensor regression (e.g., BKTR (Lei et al., 2024)), one can attribute spatiotemporal effects to each covariate (constitute feature dimension) with quantified uncertainty. In addition, at the data level, posterior uncertainty combine with robust residuals make anomaly/outlier modeling explicit and explainable, for example, robust Bayesian factorization has been used for congestion detection (Li et al., 2021).
- Building scalable probabilistic frameworks with reliable UQ remains a challenging research problem in traffic settings, especially at city/network-scale (large-scale) and under heterogeneous noise. Open issues include efficient inference such as stochastic VI, distributed/blocked MCMC, Kronecker-structured updates, robust and non-Gaussian likelihoods for counts or heavy tails, e.g., poisson/negative-binomial, student-t, zero-inflated distributions, and handling heteroscedasitic or nonstationary noise.

4.2. Connection with deterministic TF

Low-rank tensor factorization has been developed under both deterministic optimization problems and Bayesian probabilistic frameworks. While the objectives differ—the former typically seeks point estimates under explicit regularization, and the latter infers a full posterior distribution under priors and likelihoods—they often share conceptual parallels. In this subsection, we outline the connections between common regularization techniques used in deterministic TF models and prior specifications in Bayesian models, particularly for spatiotemporal traffic applications. Understanding these links can help translate insights and innovations from one paradigm to the other.

Conceptual analogy. In optimization-based models, various constraints and regularization terms, including ℓ_1, ℓ_2 norms, graph smoothness penalties, etc., are used to encode prior knowledge and improve generalization. In Bayesian models, similar structural assumptions are encoded as prior distributions over latent variables. For example:

- An ℓ_2 -norm ridge regularization on factor entries is equivalent to placing a Gaussian prior. Specifically, zero-mean Gaussian encourages small weights, analogous to Tikhonov regularization.
- An ℓ_1 -norm or sparsity penalty corresponds to a sparsity-inducing prior, such as a Laplace (double-exponential) prior or a spike-and-slab prior, which encourages most of the entries to be zero.

Table 6: Comparison of Bayesian probabilistic LRTL for spatiotemporal traffic data.

Tech.	Author (year)	Pri. S	Pri. T	Pri. O	Lik.	Inf.	Data	Task
CP-B	BATF, Chen et al. (2019a)	-	-	-	\mathcal{N}	VI	Traffic speed	[I]
	BGCP, Chen et al. (2019b)	-	-	-	\mathcal{N}	Gibbs	Traffic speed	[I]
	Han and He (2020)	-	-	DP	\mathcal{N}	VI	Smart card	[I] [PD]
	Li et al. (2021)	-	-	MGP, robust	\mathcal{N}	Gibbs	Flow, speed	[PD]
	BTTF, Chen and Sun (2021)	-	VAR	-	\mathcal{N}	Gibbs	Speed, flow	[I] [P]
	Deng et al. (2021b)	-	-	RNN	\mathcal{N}	Gibbs	Traffic volume	[P]
	Li et al. (2022)	-	-	Residual	\mathcal{N}	Gibbs	Traffic speed	[I]
	Zhu et al. (2022)	-	-	Residual	\mathcal{N}	VI	Traffic speed	[I]
	BKMF, Lei et al. (2022b)	Graph kernel	Matérn kernel	-	\mathcal{N}	MCMC	Traffic speed	[I] [E]
	BCKL, Lei et al. (2022a)	Graph kernel	Matérn kernel	Residual	\mathcal{N}	MCMC	Traffic speed	[I]
	Huang et al. (2024a)	-	-	MGP	\mathcal{N}	VI	Traffic speed	[I]
	Zhang and Wei (2024)	-	-	Residual	\mathcal{N}	Gibbs	Traffic speed	[I]
	BKTR, Lei et al. (2024)	Graph kernel	Matérn kernel	-	\mathcal{N}	MCMC	Travel demand	[R]
	Huang et al. (2025a)	-	-	Residual	\mathcal{N}	Gibbs	Traffic speed	[I]
	Li et al. (2025a)	-	-	-	\mathcal{N}	VI	EV operational	[I]
	Yang et al. (2025b)	-	-	Factor sharing	\mathcal{N}	Gibbs	Traffic speed	[P]
Others	PCA-B, Qu et al. (2008)	-	-	-	\mathcal{N}	Gibbs	Traffic volume	[I]
	MF-B, Paliwal et al. (2021)	-	state-space	-	\mathcal{N}	VI	Traffic speed	[I] [P]
	MF-B, Wu et al. (2024c)	-	AR	-	\mathcal{N}	Gibbs	Transfer time	[P]
	Tubal, Zhao et al. (2024a)	-	-	t-SVD	\mathcal{N}	VI	Traffic speed	[I] [E]
	TR-B, Liu et al. (2025a)	-	-	Sparse \mathcal{G}	\mathcal{N}	VI	Speed, flow	[I]

Abbreviations: Tech.: technique; TR: tensor ring, “-B” indicates Bayesian model; Pri. S / Pri. T / Pri. O: prior for spatial mode, temporal mode, and other aspects; Lik.: likelihood; \mathcal{N} : Gaussian; Inf.: inference approach; VI: variational inference; [I]: imputation; [P]: prediction/forecasting; [K]: kriging; [PD]: pattern discovery; [R]: regression; DP: Dirichlet process; MGP: multiplicative gamma process; RNN: recurrent neural network; EV: electrical vehicle; MF: matrix factorization.

- A variation regularizer, such as QV (quadratic variation), which is often used to enforce local smoothness by penalizing $\frac{1}{2} \|\mathbf{L}_{\text{diff}} \mathbf{U}\|_F^2$ with a differentiation operator matrix \mathbf{L}_{diff} is equivalent to placing a GP prior on the latent factors with a precision matrix being $\mathbf{L}_{\text{diff}}^\top \mathbf{L}_{\text{diff}}$.

The key difference is that probabilistic models provide UQ in addition to point estimations, whereas optimization frameworks yield only point estimates without measures of uncertainty. We next discuss specific correspondences between notable deterministic models and their Bayesian counterparts.

Specific comparisons. **TRMF vs. BTTF.** Temporal regularized MF (TRMF) (Yu et al., 2016) is a deterministic model that incorporates time-series AR regularization into matrix/tensor factorization. We have discussed this model in Section 3.1.2 and the regularization term is given in Eq. (9). The regularizer on the temporal latent factor matrix encourages rows of it to follow an AR model. In contrast, the Bayesian temporal tensor factorization (BTTF) (Chen and Sun, 2021) treats temporal dynamics probabilistically by placing a VAR prior on the time-mode factors, as described in Section 4.1.2. Rather than minimizing a single objective, BTTF infers a posterior distribution over both the latent factors and the AR coefficients, i.e., coefficient matrices become random variables with priors. In addition, one can obtain not only point estimates for unobserved data but also confidence intervals for predictions.

Connection: TRMF can be viewed as producing the MAP (maximum a posteriori) estimate of BTTF. Both models can obtain a equivalent objective function if they use the same AR structure on the temporal factors and with an appropriate weight setting. BTTF generalizes TRMF by providing full posterior distributions for the temporal factors and AR coefficients, thus quantifying uncertainty in the temporal dynamics.

GRMF vs. BKMF. Graph regularized MF (GRMF) (Rao et al., 2015) is deterministic method that imposes graph Laplacian smoothness penalties to MF matrix factorization. For example, consider a spatiotemporal traffic matrix \mathbf{Y} , which is decomposed as a spatial factor matrix \mathbf{U} and temporal factor matrix \mathbf{V} . GMRF incorporates regularization term $\text{tr}(\mathbf{U}^\top \mathbf{Lap}_u \mathbf{U})$ into the cost, where \mathbf{Lap}_u is a graph Laplacian matrix encoding similarity between sensors, thus enforcing that rows of \mathbf{U} vary smoothly over the graph. In Bayesian kernelized tensor factorization (BKTF) (Lei et al., 2022b,a), these ideas are lifted into a probabilistic setting by using GP priors with kernels that enforce the same smoothness. For instance, one might use a graph diffusion kernel $\mathbf{K}^u = (\mathbf{Lap}_u + \eta \mathbf{I}_M)^{-1}$ as the covariance for columns of \mathbf{U} , or a Matérn kernel over spatial coordinates—both choices lead to latent factors that vary smoothly with respect to the underlying graph or space. Similarly, temporal smoothness penalties can be replaced with GP priors on columns of \mathbf{V} , as we saw with BKTF.

Connection: A graph Laplacian regularization term $\mathbf{u}_{\cdot r}^\top \mathbf{Lap}_u \mathbf{u}_{\cdot r}$ in an optimization is equivalent to a Gaussian prior on $\mathbf{u}_{\cdot r}$ with precision (inverse covariance) matrix proportional to \mathbf{Lap}_u . Thus, penalty-based smoothing in GMRF is analogous to GP prior-based smoothing in BKMF. The primary difference is that BKMF, being Bayesian, yields a distribution over \mathbf{U}, \mathbf{V} and can quantify uncertainty in imputed values, whereas GRMF yields only a single estimate for the latent factors that have been smoothness-regularized.

GLSKF vs. BCKL. Generalized Least Squares kernelized tensor factorization (GLSKF) (Lei and Sun, 2024) is an optimization-based method that augments covariance norm regularized low-rank matrix/tensor factorization with GP structured residual modeling. The objective function of GLSKF is given in Eq. (17) (see Section 3.1.2), in which $\{\mathbf{K}_r^u, \mathbf{K}_r^v, \mathbf{K}_{\text{vec}(\mathcal{R})}\}$ are covariance matrices capturing the structures of the latent factors for the data dimension and the residuals. Bayesian complementary kernelized learning (BCKL) (Lei et al., 2022a), which we discussed in Section 4.1.2, achieves a similar goal in a fully Bayesian way. BCKL models the data tensor \mathcal{Y} as $\mathcal{Y} \approx \mathcal{M} + \mathcal{R} + \mathcal{E}$, with priors on both the low-rank part \mathcal{M} and the GP residual part \mathcal{R} . BCKL learns both the GP prior on the latent factors and the residual covariance structure via kernel hyperparameters and yields a posterior distribution over both \mathcal{M} and \mathcal{R} .

Connection: GLSKF can be viewed as the fixed-covariance limit of BCKL. In other words, if BCKL were conducted with a Gaussian likelihood assuming a known residual covariance, e.g., the covariance $\mathbf{K}_{\text{vec}(\mathcal{R})}$ used in GLSKF, and with tight priors, such that point estimation dominates, it would reduce to solving a GLS-weighted factorization equivalent to GLSKF. BCKL extends beyond GLSKF by learning the residual covariance rather than requiring it as input, and by placing priors on all components $\mathbf{U}, \mathbf{V}, \mathbf{W}$, as well as the residual process, to fully capture uncertainty.

Summary. These examples illustrate that many optimization-based tensor models can be interpreted as special cases or MAP estimates of Bayesian models under suitable priors, and conversely, Bayesian models often have an optimization analogue with an appropriate regularizer. Recognizing this correspondence helps in transferring insights between the two domains. For instance, a regularizer designed for an optimization model can inspire a corresponding prior in a Bayesian setting and bring the benefit of uncertainty estimation. Overall, linking regularization and prior assumptions provides a deeper understanding of how structural assumptions influence model behavior in both frameworks. We systematically summarize and compare both forms for representative spatiotemporal constraints used in LRTL in Section 6.

Compared with deterministic LRTL, Bayesian probabilistic LRTL typically sacrifices some computational efficiency and involves more complex inference, but it provides calibrated uncertainty, which is crucial for risk-sensitive traffic applications. On the other hand, compared with common UQ approaches for spatiotemporal data such as Gaussian processes, Bayesian LRTL is often more scalable in practice and can handle data sizes that standard GP methods struggle with. Designing Bayesian LRTL frameworks that remain scalable on large, heterogeneous spatiotemporal datasets therefore remains a challenging but promising research direction.

4.3. Comparison with Gaussian processes

Tensor factorization (TF) and Gaussian processes (GPs) represent two powerful but fundamentally different paradigms for modeling spatiotemporal traffic data with UQ. TF methods excel in scalability and in discovering latent low-dimensional structure from multi-way data. In contrast, GPs offer a flexible nonparametric approach to function approximation with built-in UQ (Azarhoosh and Ghazaan, 2025). In this section, we compare these approaches and discuss their complementary advantages for traffic data analysis.

GP for spatiotemporal data. GPs are nonparametric probabilistic models widely used in spatial statistics and machine learning for interpolation and smoothing of data (Banerjee et al., 2008, 2014; Rasmussen and Williams, 2006). A GP defines a distribution over functions, characterized by a mean function and a covariance/kernel function. It means that any finite set of function values is jointly Gaussian, with covariance given by the kernel evaluated at those input points. In spatiotemporal traffic modeling, one can define a GP over the traffic field such that for any two points (\mathbf{s}_i, t_i) and (\mathbf{s}_j, t_j) , with \mathbf{s} denoting location and t representing time, one can write the covariance between the data points as:

$$\text{cov}(y(\mathbf{s}_i, t_i), y(\mathbf{s}_j, t_j)) = k_{\text{space}}(\mathbf{s}_i, \mathbf{s}_j) \times k_{\text{time}}(t_i, t_j), \quad (52)$$

i.e., assuming a separable kernel that is the product of a spatial kernel and a temporal kernel. For example, k_{space} could be a graph diffusion kernel on a road network or a Matérn kernel in a continuous space, and k_{time} can be a Matérn or periodic kernel to capture daily cycles. Kriging, a classical geostatistical interpolation method, is essentially GP regression with a preset covariance (variogram) (Oliver et al., 2015); it has

been widely used for spatial interpolation of traffic sensor data. The key strength of GPs is that they provide not only predictions but also an estimate of prediction uncertainty via the variance of the posterior GP, which is crucial for risk-aware traffic applications, e.g., alerting if uncertainty in travel time is high due to sparse data.

In traffic data analysis, GPs have also been directly applied to a variety of transportation tasks, such as Bayesian calibration of parameters for fundamental-diagrams etc. (Kidando et al., 2024; Wang et al., 2024c), trajectory estimation (Wu et al., 2024a), and other estimation problems (Yuan et al., 2021).

TF as an alternative. Instead of modeling traffic as a continuous function, tensor factorization treats the data as a multi-dimensional array and attempts to decompose it into a set of latent factors. For instance, given a traffic tensor $\mathcal{Y} \in \mathbb{R}^{M \times T \times P}$ for M locations, T time-of-day slots, and P days, a CP or Tucker decomposition finds low-rank factors U, V, W for each mode, such that $\mathcal{Y} \approx \sum_{r=1}^R u_{:,r} \circ v_{:,r} \circ w_{:,r}$. This approach implicitly captures complex multi-way interactions while representing the data in a compressed form, since storing U, V, W is much cheaper than the full tensor for $R \ll \min\{M, T, P\}$. Tensor models are typically very scalable: for sparse observed data, their complexity often grows linearly with the number of known entries, making them suitable for large-scale traffic datasets. They can be fitted either by ALS / gradient-based optimization or via Bayesian inference, as discussed. In practice, TF methods have achieved impressive computational gains. We summarize representative datasets used in LRTL for traffic data in Table 11, which lists the data size applied in each study.

Comparison and connection. We here discuss the comparison and connection between low-rank TF and GP from three aspects: (i) Scalability vs. flexibility; (ii) Kernel and tensor connections; and (iii) Nonstationarity and interpretability.

(i) Scalability vs. flexibility A key comparison between TF and GPs can be summarized as scalability vs. modeling flexibility:

- **Gaussian processes:**

- *Strengths:* Principled uncertainty quantification; kernel-based modeling of spatial and temporal correlations; excellent for interpolation tasks when data is sparse but highly correlated.
- *Limitations:* In their basic form, computing a GP model needs cubic time complexity $\mathcal{O}(N^3)$ for N data points, which becomes infeasible for very large data with large N . Moreover, classical GPs do not naively handle multi-dimensional array structure or multi-way relationships beyond treating one dimension as a single long index.
- *Remedies:* Extensive research has focused on scaling up GPs, e.g., through sparse GP approximations that use inducing points or Nyström methods to reduce complexity (Banerjee et al., 2014), and through low-rank kernel approximations. These techniques sacrifice certain accuracy for computational tractability, effectively approximating the covariance matrix to make large problems manageable.

- **Tensor factorization:**

- *Strengths:* Scales to large multi-way datasets (leveraging sparsity and distributed computations); yields interpretable latent factors (e.g., one factor might capture a typical “morning rush hour” pattern across sensors); can easily incorporate side information or constraints (graph structure, temporal periodicity, etc.) through regularization or structured priors.
- *Limitations:* Standard TF does not provide predictive uncertainty out-of-the-box—it produces a point estimate reconstruction. Bayesian methods are required to develop probabilistic models for UQ. Also, it usually requires the data to be structured on a predefined grid or categories (e.g., specific time slots), meaning continuous space or time must be discretized. Furthermore, pure low-rank models might struggle with highly nonlinear or non-stationary patterns that do not lie in a low-dimensional linear subspace; while GPs with nonlinear kernels might capture these better.

(ii) Kernel and tensor connections Recent studies have drawn connections between these two paradigms. [Yu et al. \(2018\)](#) showed that a low-rank tensor regression model can be interpreted as a GP with a specially structured kernel: essentially a multilinear kernel with limited degrees of freedom. This implies that fitting a low-rank model is akin to learning a GP with a covariance constrained to have low-rank structure in its cross-covariance across modes. In another line of work, [Lei et al. \(2024\)](#) demonstrated that a structured Bayesian tensor factorization/tensor regression model corresponds to a GP whose covariance matrix is the Kronecker product of the covariance matrices of the latent factors. In other words, the Bayesian tensor model implicitly defines a multi-dimensional GP with a covariance that factorizes by mode. These observations suggest that tensor factorization can be seen as learning a constrained GP, which is more scalable benefiting from the lower-dimensional latent modeling, but potentially less flexible than a full GP as it restricts the covariance form.

Moreover, the hybrid models discussed earlier, such as BKTF ([Lei et al., 2022b](#)) and BCKL ([Lei et al., 2022a](#)), explicitly combine the two approaches: they use GP priors within a Bayesian TF framework. Such models blend the advantages—they inherit scalability from the tensor decomposition and the flexibility from the GP prior to describe correlations. Specifically, BKTF ([Lei et al., 2022b](#)) uses GP priors on factors to enforce smoothness, and BCKL ([Lei et al., 2022a](#)) combines BKTF with a local sparse GP to model residuals, resulting in uncertainty estimates for both global and local structure. These approaches are actively pushing the frontier to handle large-scale, uncertain traffic data.

(iii) Nonstationarity and interpretability A practical challenge in traffic modeling is non-stationarity, e.g., the statistical properties of traffic can change over time such as daily patterns, seasonal trends, or vary by region. GPs typically assume stationarity, unless one designs non-stationary kernels or uses techniques such as input warping, which can limit their ability to handle long-term trends or regime changes without significantly increasing model complexity. TF is able to handle non-stationary effects naturally by adding extra dimensions or factors; for instance, one can include a “day-of-week” mode to capture weekly pattern shifts, or allow factor weights to evolve over time. In BTTF ([Chen and Sun, 2021](#)), we saw an example of incorporating non-stationary dynamics via a VAR process on factors.

Another point is interpretability: The latent factors in tensor models often have clear interpretations; for example, spatial factors might correspond to clusters of sensors or regions, temporal factors might correspond to typical daily profiles like morning vs. evening traffic patterns, etc. This interpretability is valuable for diagnosing models and understanding underlying phenomena. GPs, on the other hand, encode their understanding of the data through the kernel function, which can be a more abstract object—it may be harder to directly interpret a learned kernel or the GP predictions corresponding to the exact data points. Therefore, while GPs provide smooth function estimates, the low-rank factors can sometimes offer more tangible insights, especially to domain experts.

Summary. In conclusion, tensor factorization/decomposition vs. Gaussian processes is not an either-or choice but rather a complementary toolkit. GPs are often favored for fine-grained probabilistic modeling and interpolation in relatively smaller-scale or continuous-index settings, where their computational cost is manageable and their flexibility shines. TF is often the choice for scalable analysis of large, structured spatiotemporal datasets where one seeks dominant patterns and can sacrifice some flexibility for efficiency. Ongoing research on hybrid models, such as GP-regularized tensor factorizations and GP-tensor product kernels, aims to bridge the gap between these approaches and leverage the best of both worlds. Ultimately, the choice may depend on the specific traffic application: for example, if UQ in every prediction is paramount and data volume is moderate, GPs or GP-hybrid models might be preferred; if the dataset is massive and one needs fast, interpretable decompositions, e.g., for anomaly detection or compression, tensor methods might take the lead—and for many cases, a combination will yield the most powerful solution.

5. LRTL for Transportation Tasks

In this section, we review how LRTL methods are applied across core transportation data analysis tasks. Real-world ITS generate spatiotemporal datasets that are often incomplete, large-scale, and exhibit complex spatial and temporal dynamics/dependencies ([Lone and Akhtar, 2024](#); [Ahmad Jan et al., 2025](#)). These characteristics necessitate task-specific modeling strategies to effectively extract patterns and support informed decisions. We categorize the primary transportation applications into four objective-driven groups:

- **Spatiotemporal imputation** [I] (Section 5.1)—recovering missing or corrupted sensor readings in traffic data;

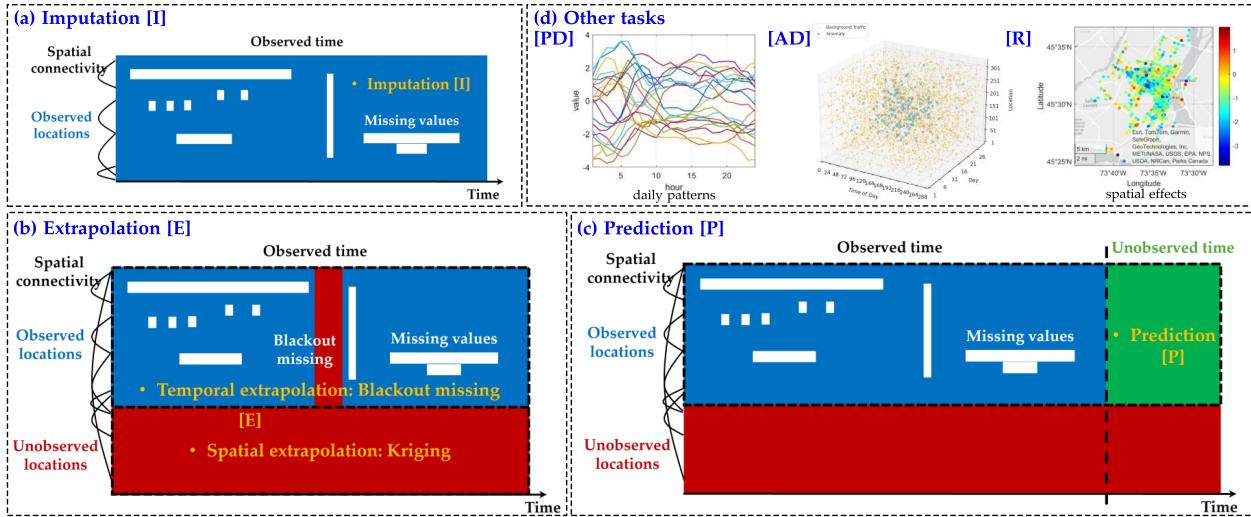


Figure 8: Illustration of LRTL for transportation tasks on a conceptional $space \times time$ spatiotemporal traffic data: (a) Imputation [I]; (b) Extrapolation [E]; (c) Prediction [P]; and (d) Other tasks, including pattern discovery [PD], anomaly detection [AD], and regression [R].

- **Extrapolation in space and time** [E] (Section 5.2)—estimating traffic states beyond observed spatial or temporal ranges;
- **Temporal prediction (forecasting)** [P] (Section 5.3)—forecasting future traffic conditions given historical data; and
- **Other tasks** (Section 5.4)—such as pattern discovery [PD], anomaly detection [AD], regression [R], etc.

For each category, we discuss representative LRTL formulations, highlight their algorithmic features (e.g., spatial/temporal regularization, probabilistic modeling), and summarize their empirical advantages in addressing domain-specific challenges. Figure 8 provides a conceptual illustration of these tasks and their interrelations on a spatiotemporal traffic data matrix ($space \times time$). The final subsection (Section 5.5) presents a cross-task comparison and summary of representative studies.

5.1. LRTL for spatiotemporal traffic data imputation

Traffic data imputation [I] is one of the most extensively studied and practically important applications of LRTL in transportation. Due to sensor malfunctions, communication issues, or limited sampling such as sparse GPS probe data, traffic datasets frequently suffer from missing or corrupted values. LRTL techniques provide a principled way to recover these missing entries by leveraging the inherent multidimensional correlations in traffic data. By modeling the data as a low-rank tensor, LRTL methods exploit spatial and temporal patterns to fill in gaps more accurately than univariate interpolation methods. Effective imputation is crucial as it forms the foundation for downstream ITS applications, e.g., reliable travel-time estimation and congestion detection depend on having complete, consistent sensor data.

Missing data patterns. In real-world scenarios, traffic data missingness can occur in different patterns (Zeng et al., 2024b; Li et al., 2024a). It is useful to classify these missing data patterns according to tensor structure (illustrated in Figure 9 using a Seattle traffic speed dataset¹ as example):

- **RE (Random Element-wise missing):** Individual entries are missing independently across the tensor. This pattern is commonly assumed in benchmarks. Many standard LRTL methods—even without specific regularization terms—perform well under moderate random missing rates, e.g., HaLRTC (Liu et al., 2012) for 10–50% RE missing. In traffic applications, early work such as Qu et al. (2009) used a probabilistic PCA (a special case of CP decomposition) for traffic flow imputation under random missing data.

¹<https://github.com/zhiyongc/Seattle-Loop-Data>

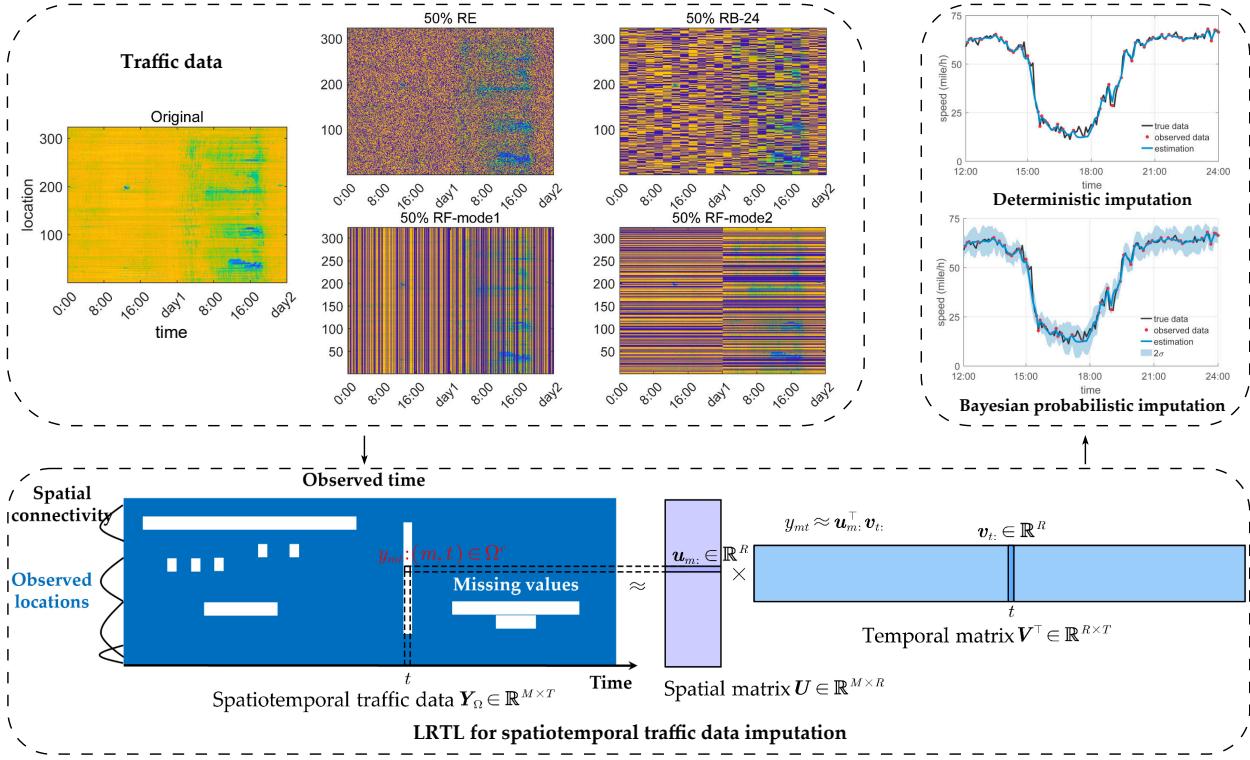


Figure 9: Illustration of LRTL for spatiotemporal traffic data imputation. The figure shows missingness patterns on the mode-1 unfolding for two days of the Seattle traffic speed dataset under 50% missing: RE (Random Element missing), RB (Random Block missing) with block size of 24 time points, RF-mode1 (Random mode-1 Fiber missing), and RF-mode2 (Random mode-2 Fiber missing).

- RB (Random Block missing): Entire blocks of data are missing due to sensor outages or communication failures. For example, an entire region of sensors may go offline simultaneously, creating a contiguous spatiotemporal block of missing entries. Block missingness is more challenging because local context is absent. LRTL models that incorporate spatial smoothness such as graph Laplacian regularization, or temporal dynamics such as AR/VAR priors, are better suited for this scenario, as they can borrow information from neighboring locations or times to infer the missing block.
- RF (Random Fiber missing): A fiber is a one-dimensional slice of the tensor along one mode, e.g., all time steps for a given road segment, or a full day of data across all sensors, see Figure 3. Missing fibers represent continuous outages along an entire dimension, such as a single sensor being down for a long period or a complete time slice missing for all sensors. This pattern requires strong inter-sensor or temporal dependencies for accurate recovery. For instance, if all data at a particular time are missing, i.e., a complete time-slice blackout, the completion task essentially becomes an extrapolation problem along the temporal dimension. Here for imputation, we consider time-slice missing randomly without continuous blackout and discuss blackout missing in extrapolation. Methods that explicitly model spatial or temporal structures can handle such cases. For example, graph kernel regularized low-rank approaches can infer missing sensors from neighbors (as done by BKMF; [Lei et al. \(2022b\)](#)), or time-series incorporated methods can forecast/interpolate a missing time-slice/interval, e.g., VAR constrained tensor completion model LATC ([Chen et al., 2021b](#)), or AR regularized TRMF ([Yu et al., 2016](#)).

These three missingness patterns (RE, RB, RF) often co-exist in practice, and robust LRTL models aim to handle complicated combinations. Figure 9 illustrates the imputation task via a spatiotemporal factorization scheme, showing these missingness patterns on an unfolded traffic matrix for clarity.

LRTL imputation methods. A variety of LRTL techniques have been developed to impute missing traffic data by minimizing reconstruction error under a low-rank assumption. Following Section 3-4, existing methods broadly fall into three groups (deterministic and Bayesian probabilistic):

- **Decomposition-based deterministic LRTL:** These methods directly factorize the incomplete tensor

into latent factors, e.g., via CP, Tucker, tensor train (TT), tensor ring (TR) decompositions, and reconstruct the data from a limited number of components. Traditional tensor decomposition methods, possibly enhanced with regularization, approximate the traffic tensor with low-rank structures capturing its salient patterns. Such approaches (including early matrix factorization (MF) methods) have been widely used for traffic imputation. Typically, they can achieve good accuracy under moderate random missing rates by exploiting local multi-way consistency.

- **Rank-surrogate relaxation based deterministic LRTL:** These approaches formulate imputation as an optimization problem, often minimizing a convex or nonconvex surrogate of the tensor rank. Examples include tensor nuclear norm ([Liu et al., 2012](#)), TNIN (truncated nuclear norm) ([Xue et al., 2018](#)), and Schatten- p norm ([Nie et al., 2012](#)) minimization methods, etc. (refer to Table 4). Such methods treat imputation as finding the lowest-rank tensor consistent with observed entries, and have shown strong performance on traffic speed/flow data—for instance, [Chen et al. \(2021b\)](#) and [Nie et al. \(2022\)](#) report high accuracy even under high missing rates by using advanced nonconvex surrogates. These relaxations, originally developed in computer vision and signal processing, have emerged as powerful tools for robust traffic data imputation and reconstruction (refer to Table 5).
- **Bayesian probabilistic LRTL:** These methods leverage probabilistic tensor factorization to model incomplete data, inferring a distribution over the latent factors. Generally missing data imputation is the basic and benchmark task for Bayesian low-rank models, such as Bayesian PCA ([Qu et al., 2008](#)), BGCP ([Chen et al., 2019b](#)), BKMF ([Lei et al., 2022b](#)), BTFC ([Xiong et al., 2010](#)), BTTF ([Chen and Sun, 2021](#)), etc. The Bayesian approach can naturally incorporate prior knowledge by imposing prior assumptions on the latent factors and quantify uncertainty in the imputed values (providing uncertainty quantification (UQ) along with predictions). This is particularly valuable for critical applications such as anomaly detection, where knowing the confidence of an imputed traffic value is as important as the imputation itself.

Compared to traditional interpolation or imputation, LRTL methods offer several advantages. First, by exploiting multi-way spatiotemporal correlations, tensor models achieve higher accuracy and more consistent imputations, especially when missing data are pervasive. Second, domain knowledge can be integrated via regularization or priors, e.g., spatial network Laplacians or temporal smoothness constraints, improving robustness/imputation performance to structured missingness. Third, Bayesian LRTL methods have shown superior performance under extreme missing conditions, maintaining reliability even when over half the data are missing. They provide uncertainty estimates for imputations, which is valuable for real-time ITS applications under data sparsity. Finally, effective imputation is foundational for downstream tasks: by recovering a complete data tensor, LRTL imputation enables subsequent analyses such as short-term forecasting and anomaly detection to operate on filled data. In summary, low-rank tensor imputation techniques form a crucial first step in many transportation analytics pipelines, transforming noisy, incomplete measurements into high-quality data for further decision-making.

5.2. LRTL for spatiotemporal traffic data extrapolation

Traffic data extrapolation [E] refers to estimating traffic conditions in unobserved spatial or temporal regions beyond the range of available data. This situation differs from imputation, which assumes that each dimension has some observations. Even for RF scenarios, we assume there are not continuous time-slice missing, which is different from the extrapolation cases. In extrapolation, certain locations or time periods have no observations at all, and we seek to predict those entirely unobserved entries. Two major forms of extrapolation arise in practice:

- **Spatial extrapolation** (kriging, K): Estimating traffic states at uninstrumented road segments or locations using data from other observed locations. For example, inferring conditions on a road segment with no sensor by leveraging data from nearby sensors. This is analogous to spatial interpolation (kriging) in geostatistics ([Oliver and Webster, 2014](#)), extended to tensor-based context where space and time are jointly modeled.
- **Temporal extrapolation** (blackout missing, BM, recovery): Predicting traffic values over contiguous time windows with no observations collected at any location, such as a complete sensor-network or communications outage/blackout. This requires forecasting beyond the observed time horizon rather than imputing within it. Unlike RF along the temporal mode (random, non-consecutive time-slice gaps), BM entails continuously missing time intervals; accordingly, we treat it under extrapolation rather than imputation.

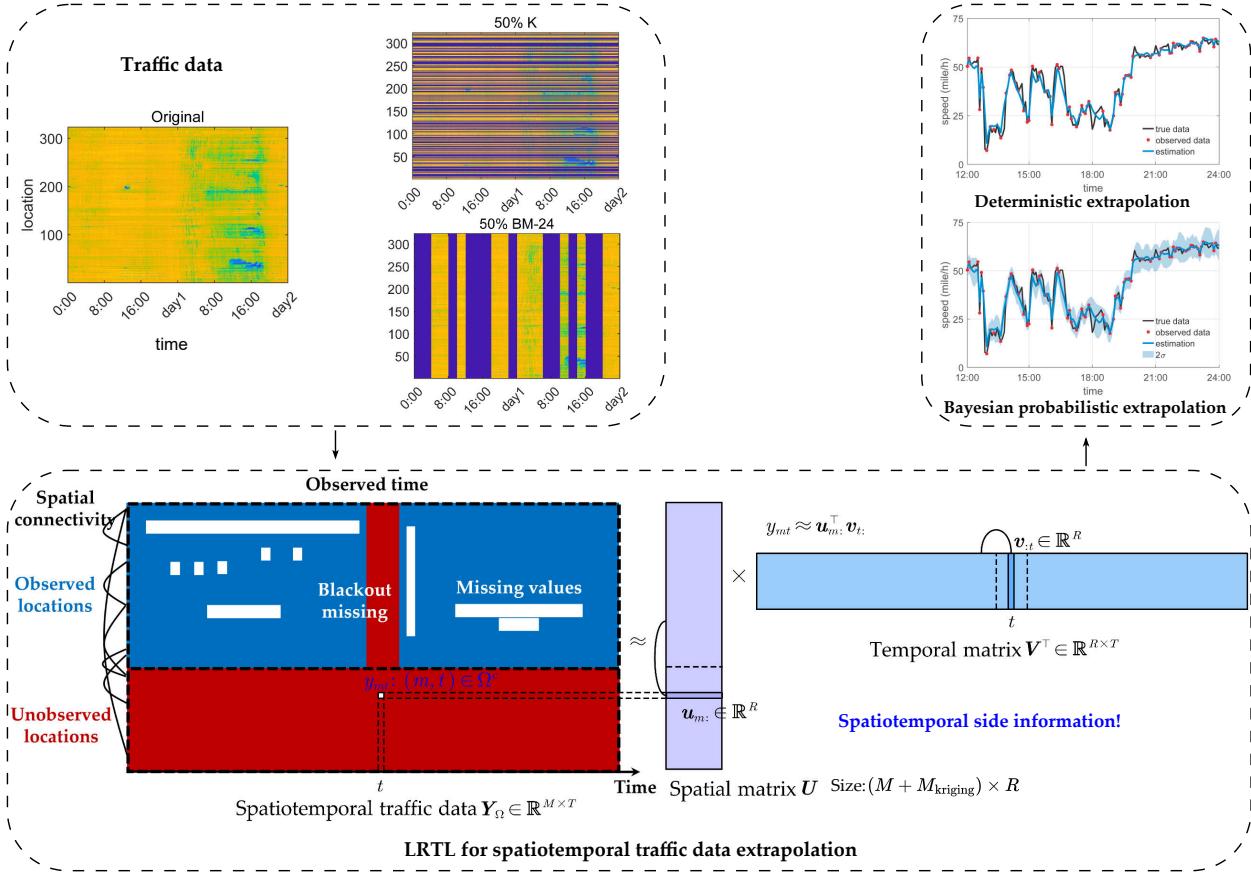


Figure 10: Illustration of LRTL for spatiotemporal traffic data extrapolation. The figure shows kriging (K) and blackout missing (BM) scenarios under 50% missing with BM block size being 24 time points.

Importance of side information. Extrapolation is an inherently under-determined problem: without additional information, predicting traffic at an entirely unobserved location or time is essentially guessing. Therefore, incorporating prior knowledge (side information) is critical for meaningful extrapolation. LRTL models facilitate this by embedding domain structures into the latent space. Common strategies include:

- **Graph-based spatial regularization:** Imposing smoothness or correlation among latent factors of road segments that are connected in the road network (Nie et al., 2023). This is achieved via graph Laplacian penalties or graph-based kernels/covariances, so that nearby or connected locations have similar latent representations. By adding a Laplacian regularizer to LRTL or using a graph kernel, unobserved locations can be inferred from observed ones in a manner consistent with network topology.
- **Temporal dynamics and priors:** Incorporating time-series models into the tensor framework, such as AR/VAR processes on latent temporal factors/temporal mode (Chen and Sun, 2021; Chen et al., 2021b), or using Hankel/delay embedding techniques to capture seasonal and trend components (Yokota et al., 2018). These constraints ensure that extrapolations follow realistic temporal patterns.
- **Kernelized similarity constraints:** Using kernel functions or covariance matrices, e.g., from GPs, to encode similarities across space or time. For instance, one can define a kernel based on spatial distance or historical correlation; this guides the extrapolation by constraining the latent factor relationships according to known similarities.

We illustrate factorization-based LRTL for spatiotemporal traffic data extrapolation in Figure 10, showing two settings on the Seattle traffic speed dataset: kriging K (spatially random) with 50% of spatial locations missing, and blackout missing BM with a contiguous 24-time-point blackout comprising 50% of the time horizon.

(1) *Spatial kriging.* For spatial extrapolation, LRTL methods generalize the idea of kriging by jointly modeling all sensor data in a tensor, rather than interpolating independently at each time step (Bae et al., 2018; Wu

[et al., 2021](#)). By leveraging spatiotemporal patterns, the tensor approach can estimate entire missing spatial slices (locations with no sensors) in a way that respects both spatial and temporal structure. Typically, LRTL models for spatial extrapolation will:

- Embed road network structure via a graph Laplacian penalty or graph kernel, to enforce that nearby locations have similar profiles.
- Learn a shared latent factor space for locations, such that an unobserved location can be assigned latent coordinates inferred from similar observed locations.
- Integrate distance-based weights or spatial kernel similarities to bias the model toward classical kriging behavior, effectively performing a low-rank interpolation that used known spatial correlations.

For instance, BKMF ([Lei et al., 2022b](#)) introduces graph kernel priors on the spatial latent factors and temporal Matérn kernel on temporal factors in a factorization model to estimate traffic speeds on unsensed roads, guided by the road network connectivity and historical traffic patterns. Another example is [Nie et al. \(2023\)](#), which combines a tensor completion model with Laplacian smoothness to enforce that the extrapolated values vary smoothly over the network topology. In essence, these approaches perform a form of multi-dimensional kriging where spatial predictions honor both topological proximity and the shared temporal dynamics learned from the data tensor.

Classical geostatistical kriging can be applied to traffic data by estimating unknown locations for each time using a variogram model. However, traditional kriging has limitations in this context: 1) it does not leverage temporal redundancy, which means that each time is interpolated independently; 2) it struggles with scalability for large, high-dimensional datasets; and 3) it assumes stationarity and requires careful variogram fitting which may not hold in dynamic traffic environments. In contrast, tensor-based extrapolation jointly models space and time, capturing cross-mode dependencies, and tends to scale more efficiently for large data. By integrating spatial and temporal regularization, LRTL methods inherently address the non-stationary and correlated nature of traffic data that classical kriging cannot easily handle.

(2) *Temporal extrapolation*. Temporal extrapolation involves estimating traffic data for completely unobserved time intervals without any contemporaneous observations. In LRTL frameworks, this is tackled by integrating interpolation models into tensor space. Key techniques include:

- Temporal factor/unfolding (mode) autoregression: Fitting an AR or VAR model on the learned temporal factor vectors or temporal mode, then projecting them forward to generate future latent factors. Methods such as TRMF ([Yu et al., 2016](#)) and LATC ([Chen et al., 2021b](#)) employ this strategy to recover missing time slices by essentially “forecasting” the tensor evolution.
- Seasonal/trend modeling via Hankel embedding: Transforming the extrapolation problem into a completion problem by constructing Hankel matrices/tensors, i.e., using delay embedding of the time series. For example, BHT-ARIMA ([Shi et al., 2020b](#)) forms a third-order block Hankel tensor from time-series data and applies low-rank Tucker decomposition, then fits an ARIMA model on the core tensor to forecast future values. Similarly, HTMF (Hankel temporal MF) ([Chen et al., 2024f](#)) creates a Hankel matrix from the latent temporal factor to a MF and enforces a low-rank structure to extrapolate future steps. These methods implicitly capture periodicities and long-term dependencies by encoding lagged information in the factorization, allowing the model to extrapolate complex temporal patterns, such as daily/weekly cycles, beyond the observed period.
- Kernelized GP (covariance regularized) low-rank modeling: The temporal periodicity can also be incorporated through covariance matrix built from for example periodic kernels, which enables extrapolation beyond local smoothness. In addition, AR process is also equivalent to OU covariance function. Representative models for spatiotemporal data extrapolation include deterministic model such as [Lei and Sun \(2024\)](#) and Bayesian probabilistic model in [Lei et al. \(2022b,a\); Lei \(2024\)](#).

Extrapolation capabilities are essential for several practical needs in ITS:

- Real-time traffic map generation: Filling in a complete traffic state map in real time from sparse sensor inputs, enabling traveler information systems and control centers to see conditions on road segments where no sensors are installed.

- Probe vehicle data expansion: Inferring traffic conditions in areas or times where probe vehicles did not traverse, by utilizing patterns from when/where they did. In effect, LRTL can extend partial trajectories or sparse measurements to create a more comprehensive picture of network conditions ([Yu et al., 2020](#)).
- Planning and infrastructure analysis: Identifying hotspots and network bottlenecks in regions with few sensors by extrapolating traffic volumes or speeds there. This helps transportation planners evaluate conditions in unmonitored areas, e.g., in feasibility studies for new sensor deployment or road infrastructure changes.

In practice, modern ITS platforms increasingly integrate tensor extrapolation models with visualization dashboards, e.g., generating real-time heatmaps of current traffic state conditions that include not only sensor readings but also estimates for unsensed locations. By combining data-driven generalization of LRTL models with transportation domain knowledge (networks, historical spatiotemporal patterns), allowing decision-makers to see beyond the direct coverage of sensors. In summary, LRTL-based extrapolation serves as a powerful tool for estimating the unseen parts of the traffic state in both space and time.

5.3. LRTL for spatiotemporal traffic data prediction

In this survey, prediction [P] (or forecasting) refers to estimating future traffic states based on past and current observations. This is a core component of proactive traffic management, congestion avoidance, and intelligent routing in ITS. Unlike imputation (which fills in current missing values in already-recorded data), or extrapolation (which fills in unobserved locations/times within the scope of current data), prediction specifically targets the temporal dimension beyond the present: forecasting what comes next. Effective forecasting demands models that can capture the complex, evolving temporal dynamics of traffic while also accounting for spatial dependencies ([Abadi et al., 2014](#); [Lana et al., 2018](#)).

Prediction tasks. Traffic prediction problems can vary in horizon (e.g., minutes-ahead vs. hours-ahead predictions) and spatial scope (predicting conditions on a single road vs. network-wide forecasting). Generally, effective predictors hinge on modeling a combination of patterns in the data:

- Short-term patterns: e.g., the propagation of congestion waves, queue build-up and dissipation at bottlenecks, or reaction to traffic signal timings on a minute-by-minute scale.
- Long-term cyclical patterns: e.g., daily rush-hour peaks, weekly weekday/weekend differences, and seasonal or annual variations in traffic states/demand.
- Sudden or irregular events: e.g., accidents, road closures, extreme weather, or special events causing abrupt deviations from normal patterns.

LRTL models can simultaneously uncover and leverage these patterns through their structured low-rank modeling of the data. By extracting multidimensional low-rank structures that correspond to general spatial patterns and temporal profiles, LRTL methods naturally represent recurring behaviors (short-term and long-term), while anomalies or events manifest as deviations in those low-rank approximations. This makes low-rank tensor-based models well-suited to serve as the backbone for traffic forecasting, often with additional constraints to explicitly handle temporal dynamics. Figure 11 illustrates a conceptual low-rank factorization for spatiotemporal traffic prediction.

Representative modeling approaches. A number of strategies have been explored within the LRTL framework to enable traffic prediction:

- Autoregressive regularized LRTL models: These methods either impose AR or VAR dynamic structure on the temporal mode of the estimated unfolding matrix, such as the tensor completion model LATC ([Chen et al., 2021b](#)) or on the temporal latent factor matrix from a low-rank factorization, such as the temporal AR regularized TRMF ([Yu et al., 2016](#)), to project it forward. By learning a transition dynamic for the temporal mode in LRLT, the model forecasts future tensor values while respecting spatiotemporal low-rank structure. For example, TRTF ([Baggag et al., 2019](#)) uses a CP decomposition and learns an AR process on each temporal factor; the learned spatial latent patterns are then advanced in time to produce multi-step forecasts.

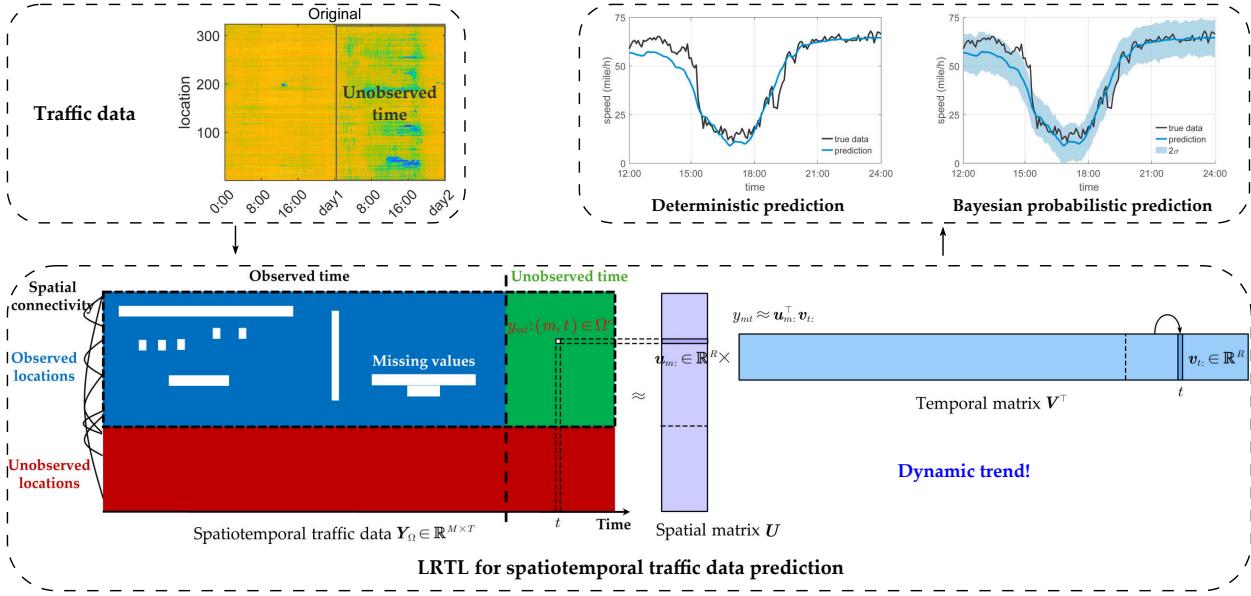


Figure 11: Illustration of LRTL for spatiotemporal traffic data prediction.

- Delay embedding (Hankelization) approaches: These convert the forecasting problem into a low-rank completion problem by constructing a Hankel matrix or tensor from the time series such that long-range dependencies and periodicities (daily/weekly cycles) are encoded explicitly. For example, the block Hankel tensor ARIMA (BHT-ARIMA) model (Shi et al., 2020b) builds a third-order Hankel tensor from traffic time series (stacking lagged copies of the data) and then applies Tucker decomposition followed by ARIMA on the core tensor. Another example, Hankel temporal MF (HTMF) (Chen et al., 2024f) forms a Hankel matrix from the latent temporal factor of a matrix factorization and performs low-rank approximation to infer future steps. Such embeddings are particularly effective when seasonality dominates and missingness is high (Yokota et al., 2018).
- Probabilistic state-space dynamic LRTL models: Probabilistic dynamic models embed LRTL within a state-space framework. The temporal evolution is modeled by a state transition equation, which could be linear or nonlinear. For example, one can imagine a Bayesian CP model coupled with a Kalman filter that evolves latent factors over time, such as the Bayesian probabilistic temporal collaborative filtering method in Xiong et al. (2010). BTTF (Chen and Sun, 2021) is another such example, where latent factors at each time step follow a VAR process with Gaussian noise, allowing the model to infer a distribution over future trajectories rather than a single forecast. Though less common in traffic literature than AR or Hankel methods, probabilistic state-space LRTL models offer a principled way to incorporate uncertainty and adapt to changing dynamics on the fly.
- Hybrid LRTL models: These approaches combine LRTL with external machine learning models (often deep neural networks) for prediction. One strategy is to use a tensor decomposition to reduce the data dimensionality and extract latent features, then feed those features into a sequence model such as RNN/LSTM or temporal convolutional network to produce forecasts. Conversely, a deep neural network could be used to generate one mode of a tensor (e.g., forecasting the future time-factor) while tensor completion handles the multi-way reconstruction. Hybrid models aim to combine the interpretability and efficiency of LRTL with the flexibility of modern deep learning. For example, Deng et al. (2021b) integrated RNN into CP factorization, and Yu et al. (2025) proposed a tensor ring factorization with learned temporal dynamics (akin to an ℓ_0 -constrained recurrent structure).

The literature has evolved from AR-regularized factorization for short-horizon link-level prediction, to explicit seasonal/Hankel embeddings that improve multi-horizon stability under high missingness, further to hybrid probabilistic/deep (neural network) models that provide uncertainty, ingest exogenous signals, and scale to network-wide forecasting and reliability analysis.

Many LRTL-based forecasting systems employ a rolling horizon strategy: they are re-fit or updated as new data arrive, using a sliding window of recent data to predict the next few time steps. This online

or incremental learning paradigm allows the models to adapt to changing traffic conditions, e.g., sudden demand shifts or incidents, while maintaining the low-rank structure for efficiency.

Empirical studies have shown that tensor-based forecasting can outperform traditional time-series models and even deep neural network models under certain conditions:

- When traffic data exhibit strong, stable spatiotemporal patterns that a low-rank model can capture succinctly;
- When useful side information, e.g., road network structure and weather, is available to inform the model via regularization or coupled modeling;
- In situations with limited training data or high missingness, where simpler low-rank models generalize better than data-hungry deep networks.

Indeed, even relatively simple combinations such as a CP decomposition with a VAR model on the temporal factors can rival or outperform complex deep learning models in low-data or cold-start scenarios. The reason is that LRTL models efficiently leverage the inherent data structure and are less prone to overfitting, whereas deep neural networks might struggle without abundant training data.

In summary, LRTL provides a flexible, interpretable, and scalable framework for traffic prediction. By integrating low-rank framework with temporal dynamics and side-information, tensor-based models can capture multi-scale patterns and adapt to new conditions, all while keeping the model complexity relatively manageable. Furthermore, recent extensions incorporate probabilistic predictions and hybrid architectures to handle uncertainty and non-linear patterns, pushing the accuracy closer to that of advanced deep learning models but with the added benefits of interpretability and uncertainty quantification (UQ). As a result, LRTL methods have become a valuable component of the traffic forecasting toolkit, particularly for short-to medium-term prediction in ITS applications.

5.4. LRTL for other tasks on spatiotemporal traffic data

Beyond imputation [I], extrapolation [E], and prediction [P], LRTL has also been widely applied to a range of other spatiotemporal traffic analysis tasks, including pattern discovery [PD], anomaly detection [AD], regression [R] and supervised learning problems, and multi-source data fusion, etc. In such tasks, the interpretability, flexibility, and efficiency of LRTL models are often advantageous, especially when combined with domain-specific side information or integrated into broader analytical workflows. We highlight a few of these extended applications below.

(1) *Pattern discovery [PD] and behavioral analysis.* One line of work uses LRTL (particularly tensor decomposition/factorization) to extract latent spatiotemporal patterns from traffic data for exploratory analysis, referred to as pattern discovery [PD]. By decomposing large traffic tensors, researchers can uncover meaningful low-dimensional structures that correspond to typical behaviors or patterns in the system. This supports tasks including periodicity analysis, travel behavior mining, and demand clustering. For example:

- Identifying periodic cycles: The latent temporal factors e.g., in a CP decomposition often reveal daily or weekly traffic cycles. For instance, one factor might correspond to the prototypical 24-hour weekday traffic pattern, another to a weekend pattern, etc. Examining these factors helps interpret the periodic components of traffic.
- Latent region discovery: By analyzing similarities in the latent spatial factors, one can group sensors or locations that have synchronized traffic dynamics. Locations with similar factor loadings likely experience traffic rises and falls in unison, indicating they belong to a common latent region or corridor of influence. This can inform clustering of road segments or OD pairs with similar usage patterns.
- Event and day segmentation: Low-rank decompositions can also assist in grouping days or time periods. For example, by looking at the representation of each day in the latent factor space, one might cluster days with similar traffic patterns, e.g., distinguishing holidays vs. weekdays, or normal days vs. special event days. Unusual days (e.g., during major events or extreme weather) will appear as outliers in the low-rank representation.

Researchers have demonstrated these uses in practice. For instance, periodicity analysis on transit ridership tensors can extract seasonal factors (e.g., distinguishing academic semester vs. occasional travelers). As an example, [Lei et al. \(2022a\)](#) used a Bayesian CP factorization on a third-order *location* \times *time of day* \times *day*

traffic speed tensor to uncover distinct daily patterns in urban traffic. Related studies for traffic state [PD] have been explored in [Yang et al. \(2019\)](#); [Chen et al. \(2023a\)](#); [Han and He \(2020\)](#); [Li et al. \(2021\)](#). Overall, factorization-based LRTL serves as a powerful exploratory tool: the learned latent factors and components act as compressed summaries of complex datasets, which can then be interpreted by analysts or visualized to yield insights. When combined with visualization techniques or clustering algorithms applied to the latent representations, LRTL-based analysis can unveil non-obvious structures in traffic data that would be difficult to discern from raw time series or spatial plots.

(2) *Anomaly detection [AD] and event identification.* Anomaly detection [AD] is crucial for real-time traffic monitoring and incident response ([Zhu et al., 2018](#)). Tensor models provide a natural way to capture “normal” traffic patterns in a low-rank structure, while allowing detection of anomalies as deviations ([Lin et al., 2024b](#)). Many LRTL anomaly detection approaches build on the concept of robust low-rank decomposition, analogous to robust PCA but extended to tensors and often with spatiotemporal regularization ([Zhou et al., 2015](#); [Ran et al., 2016a](#); [Mardani et al., 2012](#)). Typically, the observed traffic tensor \mathcal{Y} is modeled as the sum of low-rank component and a sparse component: $\mathcal{Y} = \mathcal{L} + \mathcal{S}$, where the low-rank tensor \mathcal{L} captures regular traffic dynamics, whereas the sparse tensor \mathcal{S} captures abrupt anomalies or events, e.g., sudden drops in speed due to an accident, unusual spikes in volume due to a sporting event. By constraining \mathcal{L} to be low-rank and \mathcal{S} to be sparse and often additionally enforcing temporal or spatial smoothness on \mathcal{L} and some continuity on \mathcal{S} , one can separate the abnormal signals from the background trend.

For example, GLOSS ([Sofuo glu and Aviyente, 2022](#)) imposes graph-based spatial smoothness and temporal consistency on the low-rank part while isolating anomalies in the sparse part. This model successfully detected city-scale traffic congestion anomalies by capturing normal traffic as low-rank baseline and identifying residual spikes as anomalies. The incorporation of temporal local smoothness on the sparse component helped ensure that the detected anomalies correspond to true events rather than noise. In another study, [Wu et al. \(2018\)](#) proposed fused CP decomposition for traffic volume, which imposed graph Laplacian and temporal TV smoothness on the low-rank component. The method could identify large-scale events, such as concerts or stadium events causing unusual surges in traffic flow, by spotting significant deviations captured in the sparse component. The combination of an ℓ_1 penalty to isolate sparse spikes with spatiotemporal low-rank smoothness helped reduce false alarms by ensuring anomalies were consistent with realistic event impacts, e.g., spanning multiple sensors and sustained over short time windows rather than single-sensor glitches. Similar works for traffic anomalies have also been studied in [Wang and Sun \(2021\)](#); [Mardani et al. \(2013\)](#); [Zeng et al. \(2024a\)](#); [Duan \(2025\)](#).

A strength of LRTL-based anomaly detection models is that they often handle missing data inherently, since the underlying low-rank component can be formulated to cope with incomplete inputs. In practice, this means a model can perform imputation and anomaly detection in one unified framework—filling in routine missing values as part of the low-rank reconstruction, while flagging outliers in the sparse part. The use of spatial/temporal smoothness constraints further improves detection fidelity by filtering out random noise and focusing on anomalies that exhibit certain continuity akin to real events. Such synergy makes LRTL attractive for traffic event detection: it provides a principled way to distinguish true anomalies from normal variation or sensor noise, even in noisy, incomplete data environments.

(3) *Regression [R] and classification/clustering tasks.* In supervised learning settings, LRTL can act as an effective feature extraction technique. A common scenario is regression [R], where the goal is to relate traffic data (often high-dimensional) to external variables or outputs ([Borchani et al., 2015](#); [Lei et al., 2024](#); [Wang et al., 2024a](#)). For example, predicting travel demand or transit ridership given traffic conditions and weather, or estimating emissions given traffic flows. Directly using raw spatiotemporal traffic tensor data with hundreds of sensors and time points for regression can be infeasible due to high dimensionality and multicollinearity. LRTL addresses this by providing a compact representation. A notable application is to use low-rank tensor regression ([Zhou et al., 2013](#); [Guhaniyogi et al., 2017](#); [Lock, 2018](#); [Li et al., 2018a](#); [Kossaifi et al., 2020](#)) for scalable spatiotemporally varying coefficient/effects modeling ([Gelfand et al., 2003](#); [Finley and Banerjee, 2020](#)), such as the BKTR in [Lei et al. \(2024\)](#) for spatiotemporal bike-sharing travel demand regression. Another example, [Wang et al. \(2024a\)](#) applied a CP decomposition for multi-output regression, using the latent factors as features to jointly predict transportation. The study showed that the low-rank feature captured shared temporal effects across multiple city districts, leading to more accurate and interpretable regression coefficients.

Other studies have used LRTL in classification or clustering contexts as well ([Zhao et al., 2023a](#); [Yang et al., 2019](#); [Zhao et al., 2017](#); [Wei et al., 2023](#)). By reducing data to a low-rank representation, one can apply

standard machine learning algorithms (SVMs, decision trees, etc.) on those features to classify traffic states (for example, labeling time periods as free-flow vs. congested, or detecting incident vs. normal conditions), or to cluster similar patterns. The key benefit is that the low-rank model automatically learns relevant combinations of sensors and time periods that explain most variance, which often align with meaningful patterns, e.g., overall congestion level, typical diurnal trend, etc. This can improve both the performance and interpretability of the downstream model.

In summary, LRTL serves as a bridge between raw high-dimensional traffic data and downstream supervised tasks. It yields low-dimensional embeddings that retain important information and can incorporate domain knowledge through regularization. These embeddings can then feed into various learning algorithms, providing better generalization and insight than using raw data directly.

(4) *Multi-modal data fusion and other extensions.* Transportation data rarely exists in isolation (Xing et al., 2022). Modern smart city applications often collect multi-modal datasets, such as traffic speeds, public transit ridership, rideshare demand, weather measurements, social media feeds, etc (Kashinath et al., 2021; Xing et al., 2022). LRTL methods have been extended to fuse multiple data sources by leveraging coupled factorizations and shared factors across tensors (Papalexakis et al., 2016).

For example, consider fusing traffic sensor data with public transit usage data (Kashinath et al., 2021; Zhao et al., 2020). Traffic speed can be represented as a tensor of $location \times time\ of\ day \times day$, and transit ridership as another tensor of $station \times time\ of\ day \times day$. A coupled tensor factorization could share the $time\ of\ day$ factors between the two, since both might exhibit common daily patterns, e.g., morning/evening peaks. By doing so, the model can capture how transit demand and traffic conditions co-evolve over time. Another example is combining traffic data with weather information: suppose traffic data is a tensor of $location \times hour \times date$ and weather is a matrix or small tensor of $weather\ stations \times date$, or $location \times date$ for rainfall. One can introduce an additional mode or perform a coupled factorization where the date factors are influenced by weather conditions, thereby explaining variations in traffic due to rain or snow. In an OD travel demand context, an OD flow tensor ($origin \times destination \times time$) could be coupled with a traffic speed tensor ($location \times time \times day$) via shared time factors, ensuring that the two data sources influence temporal patterns of each other, e.g., a big event increases both OD flows and road congestion simultaneously.

Common techniques for multi-source LRTL fusion include coupled matrix-tensor/tensor-tensor factorization, where a tensor and a matrix (or two tensors) are jointly factorized with one or more common factor matrices. This approach has been used to integrate diverse transportation data, achieving a richer understanding than analyzing each dataset independently (Xing et al., 2023a; Li et al., 2023b). For instance, if one city has abundant freeway traffic data but limited transit data, a coupled model with transit data from another city (or other related data) could leverage the shared temporal structures to improve the estimates for the sparse dataset—a form of transfer learning between domains (Bi et al., 2025). As urban data become more integrated, we expect coupled low-rank models to play an increasing role in cross-domain traffic analytics and smart city decision support (Zong et al., 2025). Overall, multi-modality tensor fusion extends LRTL to heterogeneous data. By finding a joint low-rank structure that spans different datasets, these methods unlock analyses that would be impossible on any single dataset alone—such as quantifying the impact of weather on traffic purely from data, or detecting anomalous events by concurrently examining traffic and social media tensors (Zißner et al., 2023).

5.5. Comparison and summary

We compare the characteristics of different transportation tasks in Table 7, discussing the challenges and requirements for each category, the types of spatiotemporal constraints typically used, and representative LRTL studies. Furthermore, Table 8 summarizes representative LRTL research in each task category, including the key regularization components (spatial, temporal, or other) and the missing data scenarios (RE, RB, RF, BM, K) evaluated, along with the highest missing rats reported in those studies. These tables highlight how each task often calls for a different mix of modeling techniques (e.g., graph constraints for spatial extrapolation, temporal dynamics for prediction, etc.), even as the underlying low-rank principle remains common.

In summary, LRTL offers a generalizable framework for a broad spectrum of transportation applications. For **imputation** [I], LRTL methods leverage multi-way correlations to recover missing data with high fidelity, providing a necessary foundation for real-time monitoring and analysis. For **extrapolation** [E], LRTL approaches integrate spatial network structure and temporal dynamics to infer traffic conditions in unobserved regions, effectively extending our sensing capabilities beyond where detectors are deployed. In **forecasting** [P], LRTL models combine low-rank tensor completion with time-series techniques

(AR/VAR, Hankel embedding, etc.) to capture complex dynamics, yielding interpretable yet accurate predictions. Beyond these core tasks, the ability to isolate structured components (low-rank vs. sparse) makes LRTL ideal for **anomaly detection** [AD], i.e., separating events from normal patterns, and **pattern discovery** [PD]—extracting latent behavioral modes over space and time. The latent space models serve as compact features for **regression** [R] and classification tasks, and the framework naturally supports data fusion across multiple sources by sharing low-rank structures.

As transportation systems become increasingly complex and data-rich, the importance of such flexible, interpretable, and scalable modeling techniques will only grow. LRTL, especially when enhanced with domain-driven constraints and coupled with modern probabilistic or deep learning elements, is poised to remain a key tool in the ITS analytics toolbox—enabling practitioners to make sense of multidimensional traffic data and to drive informed, data-centric decisions in traffic management and urban planning.

6. Spatiotemporal Constraints in LRTL for Traffic Data

Modern LRTL models for traffic data explicitly leverage the inherent spatial and temporal structure of transportation networks by incorporating spatiotemporal constraints. These constraints enforce consistency across connected locations and across contiguous time intervals, reflecting the physical road network connectivity and temporal dynamics of traffic. In deterministic optimization-based formulations, such domain knowledge is typically integrated via regularization terms added to the objective function; whereas in Bayesian probabilistic models, it enters through informative prior distributions on model parameters (latent factors). This section categorizes the common spatial and temporal constraints used in low-rank tensor learning, highlighting how each can be implemented as a deterministic regularizer or as a Bayesian prior. We then compare these approaches and summarize their roles in different traffic analysis tasks, referring to Table 10 for examples of models and the constraints they employ.

For clarity, throughout this section we assume the observed spatiotemporal traffic data are represented (after appropriate preprocessing) as a matrix \mathbf{Y}_Ω of dimensions $M \times T$, where M denotes the number of spatial locations (e.g., sensors) and T the number of time points, Ω is the set of observed indices. Let $\mathbf{U} \in \mathbb{R}^{M \times R}$ and $\mathbf{V} \in \mathbb{R}^{T \times R}$ represent the spatial and temporal latent factor matrices, respectively, with R indicating the latent dimensionality (rank). For higher-order tensors with additional modes, similar constraints can be applied to those modes as well; here we focus on the primary spatial and temporal modes. We illustrate each type of constraint in the context of a matrix/tensor factorization model, noting the parallel between an deterministic optimization perspective (regularizer $\mathcal{R}(\cdot)$) and a Bayesian probabilistic perspective (prior $\mathcal{P}(\cdot)$). Many of these constraints were introduced earlier in Sections 3.1.2, 3.2.2, and 4.1.2 when discussing specific models; here we present a consolidated view.

6.1. Spatial constraints

Spatial constraints encourage the model to respect relationships among road locations (sensors, road segments, regions) in the traffic network. By enforcing spatial smoothness or consistency (according to adjacency or similarity relationships), these constraints ensure that nearby or connected locations have similar latent representations or estimated values. The primary approaches to incorporate spatial structure in LRTL can be grouped into three categories, each with deterministic optimization-based and Bayesian probabilistic interpretations: (1) Graph-based adjacency constraints; (2) Variation-based smoothness constraints; and (3) Covariance/kernel-based constraints.

(1) *Graph-based adjacency constraints.* Assume one constructs a graph $\mathcal{G} = (S, E)$ over the spatial locations, with nodes (set S) representing sensors or road segments and edges (set E) representing neighborhood relations (e.g., adjacency on a road network or vicinity in a geographic grid). Let $\mathbf{Lap}_s \in \mathbb{R}^{M \times M}$ be the (unnormalized) graph Laplacian built from this graph, e.g., $\mathbf{Lap}_s = \mathbf{D} - \mathbf{Adj}_s$, where $\mathbf{Adj}_s = [\alpha_{ij}] \in \mathbb{R}^{M \times M}$ from edge weights and $\mathbf{D} \in \mathbb{R}^{M \times M}$ the degree matrix with $d_{ii} = \sum_j \alpha_{ij}$. A standard way to inject spatial smoothness into LRTL is to penalize variation of the latent spatial structures across adjacent nodes. In deterministic formulations, one typically adds a quadratic Laplacian regularization term to the loss (see Eq. (5) in Section 3.1.2), which encourages each pair of adjacent locations to have similar values. In a Bayesian setting, an equivalent approach is to place a GP prior on the spatial factors, with a covariance matrix derived from a graph kernel based on the graph structure. In essence, the Laplacian regularizer corresponds to a Gaussian prior with precision matrix proportional to \mathbf{Lap}_s , meaning the prior assumes that locations connected in the graph have highly correlated latent features.

Formally, the regularizer in deterministic LRTL and prior in Bayesian formulations for graph-based Laplacian constraint on \mathbf{U} can be written as:

Table 7: Comparison of transportation tasks.

Task	Challenge	Requirement	Constraints	LRTL
[I]	Random/Nonrandom missing; Extreme missingness	Global low-rankness	Local	Zhou et al. (2015) Wu et al. (2018) Jia et al. (2020) Said and Erradi (2021) Deng et al. (2021a) Nie et al. (2023) Zhao et al. (2023b) Xie et al. (2024) Zhang et al. (2024) Chen et al. (2024a) Li et al. (2024b) Zeng et al. (2024b) Gong et al. (2025) Yu et al. (2025) Nguyen et al. (2025) Chen et al. (2025c)
			Dynamic	Baggag et al. (2019) Chen and Sun (2021) Chen et al. (2021b) Paliwal et al. (2021)
			Hankel	Wang et al. (2023b)
			Covariance	Lei et al. (2022b) Lei et al. (2022a) Lei and Sun (2024)
[E]	Whole row/column missing	Side information	Graph	Nie et al. (2023)
			Dynamic	Chen et al. (2021b)
			Hankel	Yokota et al. (2018)
			Covariance	Lei et al. (2022b) Lei et al. (2022a) Lei and Sun (2024)
[P]	Evolving trend capturing	Temporal dynamics	Dynamic	Yu et al. (2016) Tan et al. (2016) Baggag et al. (2019) Chen and Sun (2021) Paliwal et al. (2021) Cheng et al. (2022) Wu et al. (2024c)
			Hankel	Chen et al. (2024f)
[PD]	Interpretable latent space	Interpretable factors	Local	Yang et al. (2019)
			Dynamic	Chen et al. (2023a)
			Covariance	Lei et al. (2022a)
[AD]	Sparse outlier detection	Robust/Local residual	Sparse residual + regularized low-rank	Wang et al. (2021) Sofuoğlu and Aviyente (2022)
[R]	Interpretable coefficients	Spatiotemporal feature explanation	Covariance	Lei et al. (2024)

"Local" constraints include difference/variation and graph Laplacian based regularization.

- Regularizer (deterministic optimization approach):

$$\mathcal{R}_{\text{Lap}}(\mathbf{U}; \mathbf{Lap}_s) = \frac{1}{2} \text{tr} \left(\mathbf{U}^\top \mathbf{Lap}_s \mathbf{U} \right) = \frac{1}{2} \sum_{(s_i, s_j) \in E} \alpha_{ij} \|\mathbf{u}_{i:} - \mathbf{u}_{j:}\|_2^2,$$

which penalizes the squared differences between latent factors of spatially adjacent nodes, thereby promoting smoothness over the spatial graph. Normalized Laplacians can be used analogously.

- Prior assumption (Bayesian probabilistic modeling):

$$\mathcal{P}_{\text{Lap}}(\mathbf{U}; \mathbf{Lap}_s) : \mathbf{u}_{::r} \sim \mathcal{N} \left(\mathbf{0}, (\mathbf{I}_M + \mathbf{Lap}_s)^{-1} \right), \quad r = 1, \dots, R,$$

Table 8: Comparison of LRTL research for different transportation tasks.

Task	Author (year)	Reg. S	Reg. T	Reg. O	Tech.	Data	MS
[I]	Tan et al. (2013)	-	-	-	Tucker	Traffic volume	90% RE,RF
	Zhou et al. (2015)	Local	Toeplitz	-	CP	OD	95% RE,RB
	Ran et al. (2016b)	-	-	-	NN	Traffic flow	80% RE,RF
	Wu et al. (2018)	Local	TV	ℓ_2, ℓ_1 norm	CP	Traffic flow	90% RE
	Chen et al. (2018)	-	-	ℓ_2 norm	Tucker	Traffic speed	80% RE,RF
	Xie et al. (2018a)	-	-	Reshape	CP	OD	90% RE
	Baggag et al. (2019)	Graph	Graph, AR	-	CP	Traffic speed	90% RE
	Chen et al. (2019a)	-	-	-	CP-B	Traffic speed	50% RE,RF
	Chen et al. (2019b)	-	-	-	CP-B	Traffic speed	50% RE,RF
	Jia et al. (2020)	Weighted	Weighted	Network	CP	Vehicle GPS	80% RE
	Yang et al. (2020a)	-	-	-	Tucker	Traffic flow	90% RE
	Gong and Zhang (2020)	-	-	Residual	Tucker	Speed, flow	80% RE,RF
	Chen et al. (2020)	-	-	-	TNN	Speed, flow	70% RE,RF
	Han and He (2020)	-	-	DP	CP-B	Smart card	50% RE,RF
	Yan et al. (2021)	-	Regression	Residual	CP	Traffic flow	RE
	Said and Erradi (2021)	Local	Entropy	-	CP	Traffic flow	80% RE,RF
	Deng et al. (2021a)	Graph spectral	Toeplitz	ℓ_2 norm	SVD	Traffic volume	90% RE,RF
	Sure et al. (2021)	-	-	-	TNN	Traffic volume	80% RE
	Chen et al. (2021a)	-	-	Tubal	NN	Traffic speed	70% RE,RF
	Chen and Sun (2021)	-	VAR	-	CP-B	Speed, flow	60% RE,RF
	Chen et al. (2021b)	-	VAR	-	TNN	Speed, flow	90% RE,RF,BM
	Nie et al. (2022)	-	-	-	Schatten- p	Speed, volume	90% RE,RF
	Li et al. (2022)	-	-	Residual	CP-B	Traffic speed	99% RE,RF
	Zhu et al. (2022)	-	-	-	CP-B	Traffic speed	50% RE,RF
	Lei et al. (2022b)	Covariance	Covariance	-	CP-B	Traffic speed	95% RE,K
	Lei et al. (2022a)	Covariance	Covariance	Residual	CP-B	Traffic speed	70% RE,RF
	Xing et al. (2023b)	-	-	Data fusion	CP	Traffic volume	50% RB
	Zhao et al. (2023b)	Manifold	-	-	TNN	Traffic speed	90% RE,RF
	Nie et al. (2023)	Graph	-	-	Schatten- p	Traffic speed	70% RE,K
	Li et al. (2023a)	-	-	Group	NN	Traffic speed	95% RB
	Lei and Sun (2024)	Covariance	Covariance	Residual	CP	Traffic speed	90% RE
	Xie et al. (2024)	Graph	Toeplitz	ℓ_2 norm	Tucker	Traffic speed	70% RE,RF
	Zhang et al. (2024)	-	Toeplitz	ℓ_2 norm	TT	Speed, flow	90% RE
	Ming et al. (2024)	-	-	ℓ_2 norm	TD	OD	RE
	Zhang and Wei (2024)	-	-	Residual	CP-B	Traffic speed	50% RE,RF
	Yu et al. (2025)	Graph	Graph	ℓ_0 norm	TR	Speed, volume	90% RE,RF
	Huang et al. (2025a)	-	-	Residual	CP-B	Traffic speed	50% RE,RF
	Liu et al. (2025a)	-	-	Sparse \mathcal{G}	TR-B	Speed, flow	60% RE,RF
[E]	Chen et al. (2021b)	-	VAR	-	TNN	Speed, flow	30% BM
	Lei et al. (2022b)	Covariance	Covariance	-	CP-B	Traffic speed	40% K
	Nie et al. (2023)	Graph	-	-	Schatten- p	Traffic speed	70% K
[P]	Tan et al. (2016)	-	Dynamic	-	TF	Traffic flow	50% RE,RF
	Baggag et al. (2019)	Graph	Graph, AR	-	CP	Traffic speed	90% RE
	Bhanu et al. (2020)	Graph	Graph	LSTM	CP	Traffic volume	40% RE
	Yang et al. (2020a)	-	-	-	Tucker	Traffic flow	90% RE
	Yan et al. (2021)	-	Regression	Residual	CP	Traffic flow	RE
	Chen and Sun (2021)	-	VAR	-	CP-B	Speed, flow	60% RE,RF
	Deng et al. (2021b)	-	-	RNN	CP-B	Traffic volume	70% RE
	Zhang et al. (2022)	-	TV	DMD	CP	Traffic speed	40% RE,RB
	Xu et al. (2024b)	-	-	-	Tucker	Traffic flow	-
	Lin et al. (2024a)	-	-	ℓ_2 norm	DPD	Traffic speed	70% RE
[PD]	Afshar et al. (2017)	-	-	Orthogonal	CP	Trip record	-
	Yang et al. (2019)	Local	-	Nonnegative	CP	Traffic speed	-
	Han and He (2020)	-	-	DP	CP-B	Smart card	50% RE,RF
	Li et al. (2021)	-	-	MGP	CP-B	Flow, speed	-
	Chen et al. (2023a)	-	VAR	-	Tucker	Trip record	-
[AD]	Wang and Sun (2021)	-	VAR	ℓ_2 norm	CP	Traffic speed	-
	Wang et al. (2021)	-	Hankel	ℓ_1 norm	NN	Traffic speed	90% RE
	Sofuoğlu and Aviyente (2022)	Graph	Graph	Residual	NN	Trip record	60% RF
[R]	Lei et al. (2024)	Covariance	Covariance	-	CP-B	Travel demand	30% RE,RF

Abbreviations: Reg. S / Reg. T / Reg. O: regularization for spatial mode, temporal mode, and other aspects; MS: missing scenario; “-B”: Bayesian model; TT: tensor train; TD: triple decomposition; TR: tensor ring; TF: tensor factorization; DP: Dirichlet process; DPD: dimension preserved decomposition; RE: random element missing; RB: random block missing; BM: blackout missing; K: kriging.

where $\mathbf{u}_{\cdot r} \in \mathbb{R}^M$ denotes the r th column of \mathbf{U} . This means that each column of the latent spatial factor matrix follows a Gaussian prior with a covariance matrix proportional to the inverse of $(\mathbf{I}_M + \mathbf{Lap}_s)$, which can also be regarded as a Gaussian Markov random field (graph GP) on each latent spatial factor with precision proportional to $(\mathbf{I}_M + \mathbf{Lap}_s)$. Intuitively, this implies that locations connected in the graph are expected a priori to have highly correlated latent features. This prior can be viewed as a GP on the graph from a graph regularized Laplacian kernel (up to scaling) (Lei et al., 2022b). Variants including using alternative graph-based kernels (such as graph diffusion/exponential \mathbf{K} , heat kernels) derived from the graph structure (Smola and Kondor, 2003).

In traffic data analysis, graph-based spatial smoothness is widely used to inject road network topology knowledge into LRTL models for imputation and extrapolation tasks. Many deterministic low-rank completion studies impose graph Laplacian penalties on the spatial factors or mode to ensure that the reconstructed traffic filed varies smoothly along the road network. In a Bayesian view, placing a GP prior with a graph-induced covariance on the spatial factors serves as a similar purpose. We list example LRTL models in deterministic and Bayesian settings for traffic data analysis in Table 9. In addition, we illustrate the correspondence between the deterministic and probabilistic forms of the Laplacian constraint under a unified factorization framework in Figure 12, highlighting how the regularization term aligns with a GP prior distribution over the latent factors.

Such spatial constraints are especially effective when the road network structure strongly influences the data, e.g., speeds or flows that propagate along connected roads. Imposing graph regularization has enabled LRTL frameworks to effectively perform spatial interpolation/extrapolation (kriging) within a low-rank model, as seen in Nie et al. (2023); Lei et al. (2022b). Table 10 lists several LRTL works that employ graph-based spatial constraints, in combination with other regularization. This constraint has proven beneficial in scenarios such as filling large sensor outages: the model can propagate information from functioning sensors to neighboring failed sensors, yielding plausible estimates that honor the underlying road connectivity.

(2) *Variation-based smoothness constraints.* Instead of using an explicit network graph, one can enforce local smoothness by penalizing differences between neighboring locations using discrete difference operators. A typical approach is to assume the locations can be ordered and then apply first-order or higher-order finite differences. Two common penalties are total variation (TV, using an ℓ_1 norm on differences) and quadratic variation (QV, using an ℓ_2 norm on differences), see Eq. (6) in Section 3.1.2. Such penalties encourage neighboring locations (in the chosen ordering) to have similar latent values by penalizing the magnitude of differences between adjacent entries in \mathbf{U} . Variation-based regularization does not require an explicit graph, but still encodes the intuition that geographically close locations or indices that are nearby in some predefined order should exhibit smooth traffic patterns. In practice, spatial QV/TV regularizer have been used alongside graph Laplacians to capture both local smoothness and network structure; e.g., Wu et al. (2018) combined a spatial TV penalty with a graph Laplacian for traffic flow completion.

From a probabilistic viewpoint, a QV smoothness penalty corresponds to a Gaussian prior on the spatial factors that enforces a certain covariance structure (for example, akin to a Matérn GP with a short length-scale, yielding smoothly varying functions). In contrast, a TV penalty can be seen as encouraging sparse spatial gradients—effectively assuming that most neighboring locations have almost equal values, with occasional larger jumps, which relates to a Laplace prior on differences, promoting piecewise-constant latent fields with sharp boundaries where needed. While TV penalties lack a simple GP interpretation due to the non-Gaussian prior, they impose a useful piecewise-constant bias on the spatial filed, which can capture situations such as uniform traffic conditions within regions separated by sharp boundaries, e.g., a congested vs. free-flow region.

Formally, let $\mathbf{L}_{\text{diff}} \in \mathbb{R}^{(M-1) \times M}$ be a first-order difference operator for the spatial dimension. For example, \mathbf{L}_{diff} could be a matrix representation of a discrete gradient along an ordering of locations, where each row of \mathbf{L}_{diff} might have entries $[1, -1]$ to compute the difference between consecutive locations. The QV constraint can then be written as:

- Regularizer (deterministic optimization-based factorization):

$$\mathcal{R}_{\text{var}(p=2)}(\mathbf{U}; \mathbf{L}_{\text{diff}}) = \frac{1}{2} \|\mathbf{L}_{\text{diff}}\mathbf{U}\|_F^2,$$

which penalizes the sum of squared differences between \mathbf{U} at consecutive locations.

- Bayesian prior assumption (probabilistic modeling):

$$\mathcal{P}_{\text{var}(p=2)}(\mathbf{U}; \mathbf{L}_{\text{diff}}) : \mathbf{u}_{::r} \sim \mathcal{N}\left(\mathbf{0}, \left(\mathbf{L}_{\text{diff}}^\top \mathbf{L}_{\text{diff}}\right)^{-1}\right), r = 1, \dots, R.$$

This implies each latent spatial factor $\mathbf{u}_{::r}$ is drawn from a zero-mean GP whose precision is $\mathbf{L}_{\text{diff}}^\top \mathbf{L}_{\text{diff}}$, which is effectively an explicit prior that the first spatial derivative is small.

The deterministic-Bayesian correspondence illustrates that variation-based regularization aligns with a GP prior assumption. In this view, the QV regularizer is the MAP solution under a Gaussian prior that enforces smooth spatial variation. Such constraints are also illustrated in Figure 12. In traffic applications, variation-based spatial constraints can be particularly useful when the exact network structure is not available. One can define \mathbf{L}_{diff} to reflect a specific spatial topology, e.g., on a linear corridor of sensors. The concept remains to penalize local spatial changes.

(3) *Covariance/kernel-based constraints.* A more general way to impose spatial structure is through a pre-defined covariance matrix or kernel function that captures similarities between locations (Lei and Sun, 2024). In an optimization framework (deterministic), one can add a covariance norm regularization term on each column of \mathbf{U} . Given a positive-definite covariance matrix $\mathbf{K}_s \in \mathbb{R}^{M \times M}$ encoding spatial relatedness (e.g., $\mathbf{K}_s(i, j)$ might be high if locations i and j are close or in the same region), a covariance-based regularizer can be written as: $\frac{1}{2} \|\mathbf{u}_{::r}\|_{\mathbf{K}_s}^2 = \frac{1}{2} \mathbf{u}_{::r}^\top \mathbf{K}_s^{-1} \mathbf{u}_{::r}$ (refer to Eq. (15) for covariance norm). The inverse \mathbf{K}_s^{-1} acts as a precision matrix to penalize deviations. This is effectively a weighted ℓ_2 penalty that measures the norm of each factor $\mathbf{u}_{::r}$ in the metric defined by \mathbf{K}_s^{-1} . By choosing \mathbf{K}_s appropriately, one can recover the earlier constraints as special instances: for example, $\mathbf{K}_s = (\mathbf{L}_{\text{diff}}^\top \mathbf{L}_{\text{diff}})^{-1}$ for a QV regularizer and $\mathbf{K}_s = (\mathbf{I}_M + \mathbf{Lap}_s)^{-1}$ for graph Laplacian regularization. Thus, this formulation subsumes graph and variation constraints as specific kernel choices. Its strength lies in flexibility: one can encode complex spatial correlations beyond local adjacency. For instance, \mathbf{K}_s could be constructed from a distance-decay kernel (e.g., Gaussian kernel on physical distance), or a combination of factors.

In Bayesian terms, using a covariance regularizer is equivalent to assuming each column of \mathbf{U} is drawn i.i.d. from a multivariate Gaussian with covariance \mathbf{K}_s : $\mathbf{u}_{::r} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_s)$, $r = 1, \dots, R$. This is precisely a GP prior on the spatial mode with covariance function \mathbf{K}_s . For example, a squared-exponential (SE) or Matérn kernel on spatial coordinates will enforce that latent factors vary smoothly over space with a certain length-scale; and a periodic kernel on locations arranged in a ring road would enforce a repeating pattern around the loop, etc.

To summarize mathematically, for a given set of spatial covariance matrices $\{\mathbf{K}_r^u\}_{r=1}^R$ (one per factor), one can write:

- Deterministic regularization:

$$\mathcal{R}_{\text{cov}}(\mathbf{U}; \{\mathbf{K}_r^u\}_{r=1}^R) = \sum_{r=1}^R \frac{1}{2} \|\mathbf{u}_{::r}\|_{\mathbf{K}_r^u}^2 = \sum_{r=1}^R \frac{1}{2} \mathbf{u}_{::r}^\top (\mathbf{K}_r^u)^{-1} \mathbf{u}_{::r}.$$

- Bayesian probabilistic prior:

$$\mathcal{P}_{\text{cov}}(\mathbf{U}; \{\mathbf{K}_r^u\}_{r=1}^R) : \mathbf{u}_{::r} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_r^u), \text{ for } r = 1, \dots, R.$$

In many cases $\{\mathbf{K}_r^u\}_{r=1}^R$ are all set to the same matrix \mathbf{K}_s , or share a parametric form with different hyperparameters per factor. See Figure 12 for a specific demonstration.

Recent kernel-based tensor models for traffic data have used this type of constraint to incorporate rich spatial information. For instance, kernelized factorization models in both Bayesian setting (Lei et al., 2022b,a, 2024) and deterministic forms (Lei and Sun, 2024) assume a covariance function over the road network or geographic space, rather than just nearest-neighbor smoothness, to better capture spatial effects or functional similarity of locations. Covariance-based spatial priors provide a general framework to bring domain knowledge—whether it is road network geometry/topology, geographic distance, land-use similarity, etc.—into low-rank models. Such constraints can improve model fidelity by encoding longer-range or non-local spatial correlations that Laplacian or local smoothness might miss. One main challenge is that a suitable covariance must be defined or learned. Furthermore, the added expressiveness often comes at a cost of additional parameters or complexity in choosing the kernel. In practice, one can estimate \mathbf{K}_s empirically from historical data (e.g., using sample covariance of traffic speeds) (Lei and Sun, 2024), or use a parametric kernel with hyperparameters learned via MCMC (Lei et al., 2022b).

6.2. Temporal constraints

Temporal constraints impose structure on the model to capture the continuity and dynamics of traffic data over time. Traffic measurements typically exhibit both short-term correlations (e.g., between consecutive time intervals) and longer-term patterns (such as daily or weekly periodicity, seasonal trends). LRTL methods can be guided to reflect these patterns. We outline several common temporal constraint strategies, again noting their roles as deterministic optimization regularizers vs. Bayesian probabilistic priors, including: (1) Temporal adjacency (graph-based) smoothness; (2) Difference-based smoothness (Toeplitz constraints); (3) Autoregressive (AR/VAR) dynamic constraints; (4) Hankel (delay embedding) constraints; and (5) Temporal covariance (kernel) constraints. We illustrate deterministic and Bayesian probabilistic forms of these temporal constraints in Figure 12.

(1) *Temporal adjacency (graph-based) smoothness.* In analogy to spatial graphs, one can construct a simple temporal graph where each time point is connected to its neighbors, e.g., a chain graph linking each time t to $t - 1$ and $t + 1$. Imposing a Laplacian penalty on this temporal graph encourages the latent factors at time t to be close to those at adjacent times, thereby smoothing out high-frequency noise. Concretely, one can define a tridiagonal temporal Laplacian $\mathbf{Lap}_t \in \mathbb{R}^{T \times T}$, with entries on the diagonal and first off-diagonals to represent a linear chain, and add the term $\frac{1}{2} \text{tr}(\mathbf{V}^\top \mathbf{Lap}_t \mathbf{V})$ to the objective. Equivalently, the Bayesian view is to place a GP prior on the time-factor matrix $\mathbf{V} \in \mathbb{R}^{T \times R}$ with a kernel that correlates nearby time points, such as a graph diffusion kernel on the time graph. This results in a prior belief that the underlying traffic state changes gradually from one interval to the next.

Let \mathbf{Lap}_t denote the temporal Laplacian. Formally, the temporal adjacency Laplacian regularization and its probabilistic counterpart setting on \mathbf{V} can be written as:

- Regularization (deterministic optimization-based LRTL):

$$\mathcal{R}_{\text{Lap}}(\mathbf{V}; \mathbf{Lap}_t) = \frac{1}{2} \text{tr}(\mathbf{V}^\top \mathbf{Lap}_t \mathbf{V}),$$

which penalizes rapid changes between successive time steps in the latent representation.

- Prior assumption (Bayesian probabilistic modeling):

$$\mathcal{P}_{\text{Lap}}(\mathbf{V}; \mathbf{Lap}_t) : \mathbf{v}_{::r} \sim \mathcal{N}\left(\mathbf{0}, (\mathbf{I}_T + \mathbf{Lap}_t)^{-1}\right), \quad r = 1, \dots, R.$$

This corresponds to a GP prior with a covariance structure that strongly correlates $\mathbf{v}_{t:}$ with $\mathbf{v}_{t \pm 1:}$ and to a lesser extent with $\mathbf{v}_{t \pm 2:}$, etc., depending on the graph connectivity. The identity \mathbf{I}_T ensures positive-definiteness of the covariance (essentially adding a small nugget or baseline variance). One can also replace the covariance function with other graph kernel functions—such as diffusion or heat kernels (Smola and Kondor, 2003).

For transportation applications, imposing temporal adjacency smoothness is useful to filter out high-frequency noise and align the model with the inherent patterns in traffic evolution, since traffic generally changes continuously rather than instantly. Such constraint has been used for short-term smoothing and ensure short-term continuity in tensor models (Yu et al., 2025). For instance, Bhanu et al. (2020) applied both spatial and temporal Laplacian regularizers in their CP factorization of traffic volumes, effectively smoothing the latent factors in time and space.

(2) *Difference-based smoothness (Toeplitz constraints).* A closely related constraint is to directly penalize first-order differences in the temporal dimension. Under a factorization scheme, this is often implemented via a Toeplitz matrix $\mathbf{T}_{\text{diff}} \in \mathbb{R}^{(T-1) \times T}$ that computes $\mathbf{v}_{t:} - \mathbf{v}_{t+1:}$ for each consecutive pair of time factor vectors. Adding a regularization term, e.g., $\frac{1}{2} \|\mathbf{T}_{\text{diff}} \mathbf{V}\|_p^p$ (with $p = 2$ giving a quadratic penalty, or $p = 1$ giving a total variation penalty) forces the factor matrix \mathbf{V} to vary slowly over time. Such regularization can be seen as the temporal analog of the spatial variation constraints described above. In a probabilistic model, a quadratic difference penalty corresponds to assuming each latent time series (i.e., each column of \mathbf{V}) follows a smooth GP, e.g., a Matérn GP that yields similar first-order smoothness; while a Toeplitz TV (ℓ_1) difference penalty aligns with a heavy-tailed process that allows occasional jumps.

Let \mathbf{T}_{diff} be the $(T - 1) \times T$ discrete difference operator matrix that computes first-order differences between adjacent time steps. Then the variation regularizer (deterministic) and prior (Bayesian probabilistic) on \mathbf{V} can be formally expresses as:

- Regularization (deterministic optimization formulation):

$$\mathcal{R}_{\text{var}(p=2)}(\mathbf{V}; \mathbf{T}_{\text{diff}}) = \frac{1}{2} \|\mathbf{T}_{\text{diff}} \mathbf{V}\|_F^2,$$

which penalizes changes between consecutive latent temporal states.

- Bayesian prior form (probabilistic modeling):

$$\mathcal{P}_{\text{var}(p=2)}(\mathbf{V}; \mathbf{T}_{\text{diff}}) : \mathbf{v}_{:,r} \sim \mathcal{N}\left(\mathbf{0}, \left(\mathbf{T}_{\text{diff}}^\top \mathbf{T}_{\text{diff}}\right)^{-1}\right), r = 1, \dots, R,$$

where $\mathbf{v}_{:,r} \in \mathbb{R}^T$ denotes the r th column of \mathbf{V} . This prior assumes that each column of latent temporal factor matrix follows a GP with a precision matrix proportional to $\mathbf{T}_{\text{diff}}^\top \mathbf{T}_{\text{diff}}$.

This correspondence illustrates how variation-based regularization in deterministic models aligns with Gaussian priors in Bayesian models.

In traffic data analysis, these smoothness priors help capture the fact that traffic usually changes gradually over short horizons. Empirically, studies have found that including such temporal regularization improves imputation performance especially during periods of rapid transitions such as congestion onset or dissipation (when unconstrained low-rank models might otherwise produce implausible oscillations). It also helps prevent the model from overfitting to noise or outlier spikes at single time points. For example, Zhou et al. (2015) introduced a finite-difference smoothing for traffic tensor completion, using a $p = 2$ (quadratic) Toeplitz constraint to discourage abrupt changes between successive days in a daily traffic tensor. Likewise, as mentioned, Wu et al. (2018) included a temporal TV penalty ($p = 1$ on first differences) to capture piecewise steady periods in daily traffic flow.

(3) *Autoregressive (AR/VAR) dynamic constraints.* To capture temporal dynamics and explicitly enable forecasting beyond the training period, many LRTL models incorporate autoregressive (AR/VAR) structure. An AR(d) or VAR(d) regularizer on the temporal factors/mode constrains the latent/low-rank representation at time t to be a linear combination of the previous d time steps. In practice, one can enforce that each row of \mathbf{V} (factor of each time point) approximately follows an AR model under a factorization scheme. This idea was popularized by the temporal regularized matrix factorization (TRMF) model (Yu et al., 2016), where adding AR constraints on the temporal factor matrix significantly improved multi-step traffic forecasting. The same idea has been extended to tensors and for traffic data.

For example, one can impose for each t : $\sum_{t \notin \mathcal{L}}^T \left\| \mathbf{v}_{t:} - \sum_{k=1}^d \mathbf{A}_k \mathbf{v}_{t-\ell_k:} \right\|_2^2$ as a regularization term, where $\mathcal{L} = \{\ell_1, \dots, \ell_k\}$ is a set of lag indices (e.g., 1 for AR(1) or 1, 24, 168 for daily/weekly seasonal AR terms on a hourly temporal resolution data) and $\{\mathbf{A}_k\}_{k=1}^d$ are coefficient matrices of size $R \times R$ to be learned. This encourages the latent time series to follow a VAR(d) process. Unlike the smoothness constraints which impose a generic “slowness”, AR constraints learn the temporal dependencies from data. They can capture lagged effects or seasonal cycles when setting lags ℓ_k following the data periodicity (e.g., 24 hours for daily seasonality) or when combined with seasonal difference terms.

Bayesian probabilistic approaches treat AR dynamic constraints as priors: instead of fixing an AR coefficient and penalizing deviations, one places a prior on the latent time factors that they are generated by an AR or VAR process. Chen and Sun (2021) follow this approach in their Bayesian temporal tensor factorization (BTTF) model, where a VAR prior on \mathbf{V} , coupled with conjugate priors on the AR coefficients themselves, allows the model to infer a distribution and quantify uncertainty in predictions.

To formalize AR constraints, suppose we have lag set $\mathcal{L} = \{\ell_1, \dots, \ell_d\}$ and corresponding coefficient matrices $\mathbf{A}_1, \dots, \mathbf{A}_d$ for a VAR(d) process. We only penalize for t not in the lag set or beyond the largest lag. Then in a low-rank factorization scheme:

- Regularizer (deterministic approach):

$$\mathcal{R}_{\text{AR}(d)}(\mathbf{V}; \{\mathbf{A}_k\}_{k=1}^d) = \frac{1}{2} \sum_{t \notin \mathcal{L}}^T \left\| \mathbf{v}_{t:} - \sum_{k=1}^d \mathbf{A}_k \mathbf{v}_{t-\ell_k:} \right\|_2^2,$$

which encourages each latent state $\mathbf{v}_{t:}$ to be well-approximated by a linear combination of its d process.

- Bayesian prior (probabilistic modeling):

$$\mathcal{P}_{\text{AR}(d)}(\mathbf{V}; \{\mathbf{A}_k\}_{k=1}^d) : \mathbf{v}_{t:} \sim \mathcal{N}\left(\left[\mathbf{A}_1, \dots, \mathbf{A}_d\right] \begin{bmatrix} \mathbf{v}_{t-\ell_1:} \\ \vdots \\ \mathbf{v}_{t-\ell_d:} \end{bmatrix}, \Sigma_g\right), \text{ for } t \notin \mathcal{L},$$

where $\Sigma_g \in \mathbb{R}^{R \times R}$ is the covariance matrix of the process noise. This formulation assumes that $\mathbf{v}_{t:} = \sum_{k=1}^d \mathbf{A}_k \mathbf{v}_{t-\ell_k:} + \mathbf{g}_t$, where $\mathbf{g}_t \in \mathbb{R}^R$ denotes a noise process for time t . The coefficient matrices $\{\mathbf{A}_k\}_{k=1}^d$ and the noise covariance Σ_g can be learned by placing a conjugate Matrix-Normal inverse-Wishart (MNIW) prior, as discussed in Eqs. (41)-(43).

Such equivalence again illustrates the close connection between penalized optimization and Bayesian inference: the AR regularizer corresponds to a MAP estimate under a AR prior. Bayesian modeling, however, has the added advantage of allowing posterior sampling or variational inference over latent factors, model coefficients, and uncertainty propagation—especially important for downstream prediction or anomaly detection tasks.

In transportation context, incorporating AR/VAR constraints has proven essential for forecasting [P] / temporal extrapolation [E] tasks. A static LRTL with smoothness regularization can impute within the time span of training/observed data, but cannot extrapolate or predict without a mechanism to project forward. AR constraints provide that mechanism by explicitly modeling the temporal evolution. Representative models include deterministic TNN (truncated nuclear norm)-minimization-based LATC (Chen et al., 2021b) for traffic speed and flow extrapolation [E] under BM scenarios, and Bayesian probabilistic BTTF (Chen and Sun, 2021) for traffic speed prediction [P].

(4) *Hankel (delay embedding) constraints.* To model longer-term dependencies and recurring seasonal patterns, several tensor-based methods incorporate temporal constraints using Hankelization, or more generally delay embedding techniques (Li et al., 1997; Yokota et al., 2018). This idea is rooted in classical time-series analysis such as singular spectrum analysis (SSA), where a time series is mapped to a trajectory matrix (Hankel matrix) whose rows (or columns) are time-shifted (lagged) versions of the original series (Golyandina, 2020). Imposing low-rank structure in such a Hankel matrix effectively forces the time series to lie in a low-dimensional linear dynamical subspace, capturing repeating patterns with a small number of modes.

In LRTL, a Hankel matrix can be constructed from the temporal factor matrix $\mathbf{V} \in \mathbb{R}^{T \times R}$ by arranging lagged segments of each column $\mathbf{v}_{:,r} \in \mathbb{R}^T$ (each latent time series) into a block Hankel form. For example, after choosing a window length τ , one can form $\mathcal{H}_\tau(\mathbf{V}) \in \mathbb{R}^{\tau \times (T-\tau+1)}$ where each column is a length- τ segment of $\mathbf{v}_{:,r}$ (i.e., $[\mathbf{V}(t,r), \mathbf{V}(t+1,r), \dots, \mathbf{V}(t+\tau-1,r)]^\top$ for $t = 1, \dots, T-\tau+1$). Enforcing rank ($\mathcal{H}_\tau(\mathbf{V})$) to be small (or adding a nuclear norm penalty on ($\mathcal{H}_\tau(\mathbf{V})$)) encourages the series to be approximated by a low-dimensional linear system. Essentially, this constrains $\mathbf{v}_{:,r}$ to have a limited number of distinct frequencies or recurring patterns. When applied to all R factors or directly to the reconstructed tensor, such constraint can effectively model periodic behavior in traffic data (daily rush-hour patterns, weekly cycles, etc.).

While Hankel constraints are usually implemented in deterministic frameworks (as a structured low-rank constraint on transformed data), they also admit a probabilistic interpretation: one can view enforcing low-rank on $\mathcal{H}_\tau(\mathbf{V})$ as assuming $\mathbf{v}_{:,r}$ arises from a low-order linear dynamical system. This is akin to saying there exists a state-space model of order d that generates $\mathbf{v}_{:,r}$, with d much smaller than T —effectively a state-space prior on the latent factors. It implicitly regularizes the model by limiting temporal complexity and focusing on dominant oscillatory modes.

Formally, one might not include an explicit “Hankel regularizer” in a loss function (since it constrains on a non-linear transformation of \mathbf{V}), but rather integrate it by constructing an augmented tensor or adding a factorization stage. Conceptually we can say:

$$\text{Regularizer: } \mathcal{R}_{\text{Hankel}}(\mathbf{V}; \tau) : \text{rank}(\mathcal{H}_\tau(\mathbf{V})) = R \quad \text{or} \quad \min \|\mathcal{H}_\tau(\mathbf{V})\|_*,$$

where $\mathcal{H}_\tau(\mathbf{V})$ means applying a Hankel embedding (with embedding dimension τ) to each column of \mathbf{V} and stacking, yielding a block Hankel matrix. Currently, a Bayesian probabilistic model of such constraint has not been developed yet, which can be a promising direction.

Hankel/delay embedding constraints bridge LRTL with subspace system identification and spectral analysis techniques. By including a Hankel embedding temporal constraint, LRTL models can effectively

capture seasonal and long-range correlations that static low-rank or short-memory AR models might miss. They have been especially useful for capturing periodic traffic phenomena (daily rush hours, weekly cycles) within a low-rank framework. In traffic data analysis, Hankel constraints have been particularly effective in seasonal pattern extraction tasks (Chen et al., 2024f; Wang et al., 2023b), where separating low-rank (regular periodic) dynamics from sparse (irregular or anomaly) components is critical. For example, the periodic weekday/weekend structure in traffic can be captured by a Hankel low-rank model, and deviations from it can signal anomalies. In forecasting models such as BHT-ARIMA (Shi et al., 2020b), the Hankel structure is used explicitly to enforce seasonal consistency.

(5) *Temporal covariance constraints.* Analogous to the spatial case, a broad way to encode temporal structure in LRTL is by assuming a covariance/kernel matrix across time mode. In a Bayesian probabilistic low-rank factorization, one can place a GP prior on each column of the temporal factor matrix \mathbf{V} with a kernel that reflects temporal dynamics, such as a Matérn kernel for smoothness with a characteristic correlation time, or a periodic kernel for daily/weekly cycles (Rasmussen and Williams, 2006; Lei et al., 2024). In a deterministic optimization setting, this is equivalent to applying a covariance norm regularization to the columns of \mathbf{V} (Lei and Sun, 2024).

Formally, let $\{\mathbf{K}_r^v \in \mathbb{R}^{T \times T}\}_{r=1}^R$ be temporal covariance matrices for the columns of $\mathbf{V} = [\mathbf{v}_{:1}, \dots, \mathbf{v}_{:R}]$, the temporal covariance constraint in deterministic and probabilistic models can be written as:

- Deterministic regularizer (covariance norm):

$$\mathcal{R}_{\text{cov}}(\mathbf{V}; \{\mathbf{K}_r^v\}_{r=1}^R) = \sum_{r=1}^R \frac{1}{2} \|\mathbf{v}_{:r}\|_{\mathbf{K}_r^v}^2 = \sum_{r=1}^R \frac{1}{2} \mathbf{v}_{:r}^\top \mathbf{K}_r^v \mathbf{v}_{:r},$$

where $\|\cdot\|_{\mathbf{K}_r^v}$ denotes the covariance norm (refer to Eq. (15)) which can be explained as the Mahalanobis distance with respect to the inverse covariance.

- Bayesian probabilistic prior (GP on time):

$$\mathcal{P}_{\text{cov}}(\mathbf{V}; \{\mathbf{K}_r^v\}_{r=1}^R) : \mathbf{v}_{:r} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_r^v), \quad r = 1, \dots, R.$$

One can set $\mathbf{K}_r^v = \mathbf{K}_t$ for all $r \in [1, R]$ for simplicity. In practice, choice of \mathbf{K}_t encodes temporal assumptions: a SE kernel yields a very smooth GP (akin to an infinite-order AR with decaying influence), while a periodic kernel directly enforces repeated cycles. In transportation applications, \mathbf{K}_t is often constructed to reflect diurnal and weekly periodic correlations in traffic speed data (Lei et al., 2022b); (Lei and Sun (2024)). As with spatial kernels, the main challenge is choosing or learning \mathbf{K}_t . In Bayesian models, kernel hyperparameters (e.g., length-scales, variances) can be learned from data (e.g., MCMC), enabling data-driven control of smoothness and periodicity. Efficient MCMC schemes for kernelized low-rank factorization are reported in (Lei et al. (2022b,a); Lei (2024)).

Covariance-based temporal constraints thus provide a flexible, principled way to capture complex temporal correlations within a unified Gaussian framework, generalizing classical finite-difference smoothness and AR/VAR dynamics. Conceptually, the optimization and Bayesian views are equivalent: the covariance-norm penalty corresponds to the MAP solution under the GP prior. Figure 12 illustrates this equivalence alongside other temporal constraints (graph-based, variation-based smoothness, AR dynamics, and Hankel embedding), showing how domain knowledge can be incorporated either as explicit penalties or as structured priors.

6.3. Comparison and summary

Spatial and temporal constraints in deterministic and probabilistic forms play complementary roles in LRTL models for traffic data, and choosing the right combination is often key to success in a given application. Table 9 summarizes deterministic and Bayesian prior formulations for common spatial and temporal regularization, comparing the abilities for common transportation tasks and lists several example LRTL research. Table 10 summarizes a range of LRTL models and the constraints they employ. Note that (here) in these tables, we use “Graph” to denote an undirected graph Laplacian-based constraint for spatial regularizer; for temporal, “Graph” refers to a chain graph (adjacency in time), which is conceptually similar to Toeplitz difference constraint.

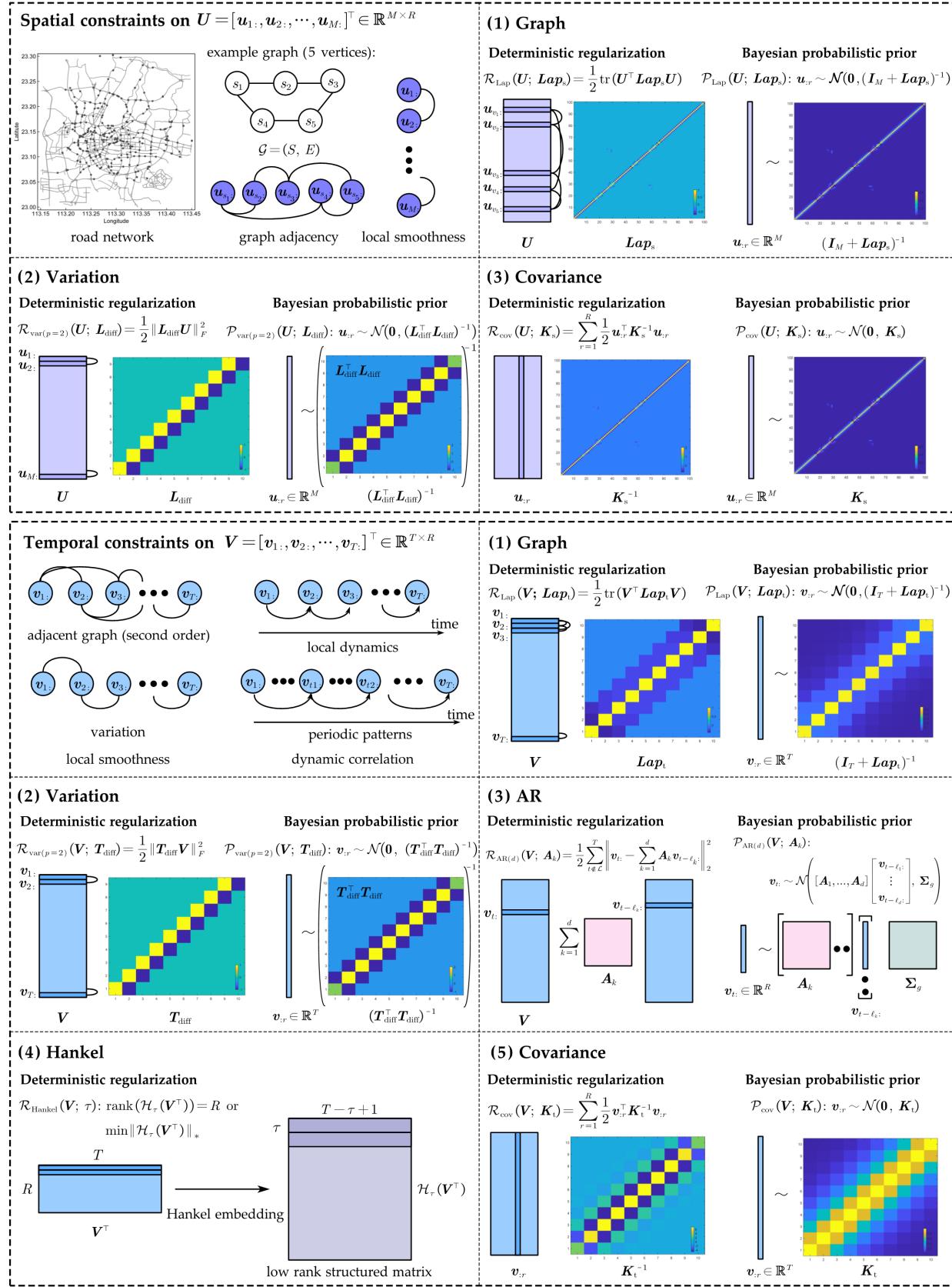


Figure 12: Conceptual illustration of spatial and temporal constraints based on a factorization scheme: implemented through regularization terms in deterministic optimization formulations and prior assumptions in Bayesian probabilistic approaches.

A clear pattern is that most of these regularization terms are geared toward improving data imputation [I] (completion) and denoising. Even without any explicit regularizer, a low-rank model can fill in missing traffic data by leveraging multi-dimensional correlations; however, adding spatial/temporal constraints significantly enhances the accuracy and consistency of imputation, especially under high missing rates or structured missing scenarios. Many studies report that including both spatial and temporal smoothness yields more realistic reconstructions. For example, [Gong et al. \(2025\)](#) found that removing either the graph-based spatial penalty or the temporal differencing penalty led to worse completion performance, underscoring that regularization in both dimensions was crucial. This joint approach helps the model propagate information in two directions: across locations (so that neighbors can borrow strength from each other when one has missing data) and across time (so that past and future observations inform a gap), thereby dramatically improving resilience to missing or noisy data.

For prediction/extrapolation tasks, incorporating temporal dynamics is essential. A static low-rank factorization, with or without smoothness regularization, can interpolate missing values in the time range of the training data, but it cannot predict future changes unless the model is endowed with a mechanism to project forward in time. Thus, for forecasting traffic conditions, deterministic models often include AR/VAR regularizers or related dynamic components such as RNNs or state-space models, and their Bayesian counterparts use priors that encapsulate those dynamics. For instance, the TRMF model ([Yu et al., 2016](#)) with AR constraints was explicitly designed for multi-step prediction, and follow-up tensor models combined low-rank structure with recurrent neural networks to capture nonlinear temporal dynamics, e.g., [Deng et al. \(2021b\)](#) integrates an RNN into a CP factorization. As reflected in Table 10, approaches targeting forecasting (marked as prediction [P] or extrapolation [E] in the table) invariably include some form of temporal dynamic constraint (AR, VAR, RNN, state-space, or “dynamic” factors) to enable time-forward inference. In contrast, methods aimed purely at imputation [I] or denoising often suffice with smoothness constraints and do not require an explicit temporal evolution model (some early works even achieved good imputation with no spatiotemporal regularizers beyond low-rank structure). That said, even for imputation, temporal constraints (smoothness or seasonal priors) help capture realistic traffic rhythms and prevent overfitting to noise.

From a regularization versus prior perspective, there is a strong conceptual correspondence between optimization-based and Bayesian approaches in how they encode spatiotemporal structure. Nearly every deterministic regularizer has an analogous probabilistic interpretation: e.g., graph Laplacian and difference penalties correspond to GP priors with appropriate covariance functions, and an AR term corresponds to an assumption that the latent factors follow an AR process *a priori*. This connection means that a regularized factorization is often obtaining the maximum a posteriori (MAP) solution of a suitably constructed Bayesian model. The chief distinction is that Bayesian treatments yield a full posterior distribution over the latent factors and predicted values, which provides uncertainty quantification (confidence intervals, etc.) that is not available in point-estimate methods. In practice, when data are plentiful and noise is moderate, optimization and Bayesian methods often produce similar mean estimates, but the Bayesian models offer additional insight by quantifying the confidence in those estimates. This can be advantageous in critical traffic applications such as anomaly detection or risk assessment, where knowing the uncertainty of an imputed travel time or predicted congestion level is as important as the value itself.

In summary, incorporating spatial and temporal constraints allows LRTL models to better respect the underlying road network relationships and time-series behaviors inherent in traffic data. Graph-based and variation-based regularizers imbue models with local consistency, while kernel-based priors offer a flexible way to encode complex correlations in a unified framework. Temporal dynamics constraints enable forecasting and capture sequential trends that static low-rank structure alone cannot handle. Table 10 highlights that recent works often combine multiple constraints (e.g., a graph Laplacian for space with an AR or RNN for time, sometimes alongside additional penalties such as sparsity or residual modeling) to tackle the multifaceted nature of traffic data. The combination of these techniques—whether through deterministic regularization or Bayesian priors—has proven effective across tasks such as imputation, prediction, anomaly detection, and pattern mining. Moving forward, we expect continued innovation in spatiotemporal constraints, e.g., learning data-driven graph structures, using physics-informed priors, to further improve the fidelity and interpretability of low-rank traffic models, ultimately enhancing their utility in intelligent transportation systems.

Table 9: Comparison of spatiotemporal constraints in deterministic (regularizer \mathcal{R}) and Bayesian probabilistic (prior \mathcal{P}) LRTL.

		Constraint	[I]	[E]	[P]	UQ	LRTL
S.	Graph	$\mathcal{R}_{\text{Lap}}(\mathbf{U}; \mathbf{Lap}_s) = \frac{1}{2} \text{tr}(\mathbf{U}^\top \mathbf{Lap}_s \mathbf{U})$	✓	✓	-	✗	Zhou et al. (2015) [I] Wu et al. (2018) [I] Yang et al. (2019) [PD] Jia et al. (2020) [I] Zhao et al. (2023b) [I] Nie et al. (2023) [I] [E] Xie et al. (2024) [I] Shu et al. (2024) [I] Li et al. (2024b) [I] Chen et al. (2024d) [I] Gong et al. (2025) [I] Yu et al. (2025) [I]
		$\mathcal{P}_{\text{Lap}}(\mathbf{U}; \mathbf{Lap}_s) : \mathbf{u}_{:r} \sim \mathcal{N}(\mathbf{0}, (\mathbf{I}_M + \mathbf{Lap}_s)^{-1})$	✓	✓	-	✓	Lei et al. (2022b) [I] [E] Lei et al. (2022a) [I] Lei et al. (2024) [R]
Variation		$\mathcal{R}_{\text{var}(p)}(\mathbf{U}; \mathbf{L}_{\text{diff}}) = \frac{1}{2} \ \mathbf{L}_{\text{diff}} \mathbf{U}\ _p^p$	✓	✗	-	✗	Wu et al. (2018) [I] Zeng et al. (2024b) [I] Yu et al. (2024) [I] Zeng et al. (2025) [I]
		$\mathcal{P}_{\text{var}(p=2)}(\mathbf{U}; \mathbf{L}_{\text{diff}}) : \mathbf{u}_{:r} \sim \mathcal{N}(\mathbf{0}, (\mathbf{L}_{\text{diff}}^\top \mathbf{L}_{\text{diff}})^{-1})$	✓	✗	-	✓	-
Covariance		$\mathcal{R}_{\text{cov}}(\mathbf{U}; \{\mathbf{K}_r^u\}_{r=1}^R) = \frac{1}{2} \sum_{r=1}^R \mathbf{u}_{:r}^\top (\mathbf{K}_r^u)^{-1} \mathbf{u}_{:r}$	✓	✓	-	✗	Lei and Sun (2024) [I]
		$\mathcal{P}_{\text{cov}}(\mathbf{U}; \{\mathbf{K}_r^u\}_{r=1}^R) : \mathbf{u}_{:r} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_r^u)$	✓	✓	-	✓	Lei et al. (2022b) [I] [E] Lei et al. (2022a) [I] Lei et al. (2024) [R]
T.	Graph	$\mathcal{R}_{\text{Lap}}(\mathbf{V}; \mathbf{Lap}_t) = \frac{1}{2} \text{tr}(\mathbf{V}^\top \mathbf{Lap}_t \mathbf{V})$	✓	✓	✗	✗	Jia et al. (2020) [I] Yan et al. (2021) [I] Yu et al. (2025) [I]
		$\mathcal{P}_{\text{Lap}}(\mathbf{V}; \mathbf{Lap}_t) : \mathbf{v}_{:r} \sim \mathcal{N}(\mathbf{0}, (\mathbf{I}_T + \mathbf{Lap}_t)^{-1})$	✓	✓	✗	✓	-
Variation		$\mathcal{R}_{\text{var}(p)}(\mathbf{V}; \mathbf{T}_{\text{diff}}) = \frac{1}{2} \ \mathbf{T}_{\text{diff}} \mathbf{V}\ _p^p$	✓	✗	✗	✗	Zhou et al. (2015) [I] Zhang et al. (2024) [I] Xie et al. (2024) [I] Shu et al. (2024) [I] Zeng et al. (2024b) [I] Yu et al. (2024) [I] Gong et al. (2025) [I] Zeng et al. (2025) [I] Chen et al. (2025c) [I]
		$\mathcal{P}_{\text{var}(p=2)}(\mathbf{V}; \mathbf{T}_{\text{diff}}) : \mathbf{v}_{:r} \sim \mathcal{N}(\mathbf{0}, (\mathbf{T}_{\text{diff}}^\top \mathbf{T}_{\text{diff}})^{-1})$	✓	✗	✗	✓	-
AR		$\mathcal{R}_{\text{AR}(d)}(\mathbf{V}; \{\mathbf{A}_k\}_{k=1}^d) = \frac{1}{2} \sum_{t \notin \mathcal{L}} \left\ \mathbf{v}_{t:} - \sum_{k=1}^d \mathbf{A}_k \mathbf{v}_{t-\ell_k:} \right\ _2^2$	✓	✓	✓	✗	Tan et al. (2016) [P] Baggag et al. (2019) [I] [P] Wang and Sun (2021) [AD] Chen et al. (2021b) [I] Chen et al. (2023a) [PD]
		$\mathcal{P}_{\text{AR}(d)}(\mathbf{V}; \{\mathbf{A}_k\}_{k=1}^d) : \mathbf{v}_{t:} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{A}_1, \dots, \mathbf{A}_d \end{bmatrix} \begin{bmatrix} \mathbf{v}_{t-\ell_1:} \\ \vdots \\ \mathbf{v}_{t-\ell_d:} \end{bmatrix}, \Sigma_g\right)$	✓	✓	✓	✓	Chen and Sun (2021) [I] [P]
Hankel		$\mathcal{R}_{\text{Hankel}}(\mathbf{V}; \tau) : \text{rank}(\mathcal{H}_\tau(\mathbf{V})) = R$	✓	✓	✓	✗	Wang et al. (2021) [AD] Wang et al. (2023b) [I] Chen et al. (2024f) [P]
Covariance		$\mathcal{R}_{\text{cov}}(\mathbf{V}; \{\mathbf{K}_r^v\}_{r=1}^R) = \frac{1}{2} \sum_{r=1}^R \mathbf{v}_{:r}^\top (\mathbf{K}_r^v)^{-1} \mathbf{v}_{:r}$	✓	✓	✓	✗	Lei and Sun (2024) [I]
		$\mathcal{P}_{\text{cov}}(\mathbf{V}; \{\mathbf{K}_r^v\}_{r=1}^R) : \mathbf{v}_{:r} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_r^v)$	✓	✓	✓	✓	Lei et al. (2022b) [I] [E] Lei et al. (2022a) [I] Lei et al. (2024) [R]

Abbreviations: S. / T.: spatial constraint, temporal constraint.

Table 10: Summary of spatiotemporal constraints in LRTL for traffic data.

Constraints	-	Graph	Variation	T. AR	Hankel	Covariance
S. -	Afshar et al. (2017) (Orthogonal,CP [PD]) Xie et al. (2018a) (Align,CP [I]) Xing et al. (2023b) (DF,CP [I]) Tan et al. (2013) (Tucker [I]) Chen et al. (2018) (Tucker [I]) Yang et al. (2020a) (Tucker [I] [P]) Gong and Zhang (2020) (Tucker [I]) Xu et al. (2024b) (Tucker [P]) Lin et al. (2024a) (Convolution,DPD [P]) Ming et al. (2024) (TD [I]) Ran et al. (2016b) (NN [I]) Chen et al. (2021a) (Tubal,NN [I]) Chen et al. (2022) (NN [I]) Li et al. (2023a) (NN [I]) Chen et al. (2020) (TNN [I]) Sure et al. (2021) (TNN [I]) Nie et al. (2022) (Schatten- p [I]) Chen et al. (2019a) (CP-B [I]) Chen et al. (2019b) (CP-B [I]) Han and He (2020) (CP-B [I] [PD]) Li et al. (2021) (CP-B [PD]) Deng et al. (2021b) (RNN,CP-B [P]) Li et al. (2022) (CP-B [I]) Zhu et al. (2022) (CP-B [I]) Zhang and Wei (2024) (CP-B [I]) Huang et al. (2025a) (CP-B [I])	Yan et al. (2021) (CP [I])	Zhang et al. (2022) (DMD,CP [P]) Zhang et al. (2024) (TT [I]) Chen et al. (2023a) (Tucker [PD]) Tan et al. (2016) (Dynamic MF [P]) Chen et al. (2021b) (TNN [I]) Chen and Sun (2021) (CP-B [I] [P])	Wang and Sun (2021) (CP [AD]) Chen et al. (2023a) (TT [I]) Chen et al. (2021b) (Tucker [PD]) Chen et al. (2021b) (TNN [I]) Chen and Sun (2021) (CP-B [I] [P])	Chen et al. (2024f) (MF [P]) Wang et al. (2021) (ℓ_1 ,NN [AD]) Wang et al. (2023b) (TNN [I])	
Graph		Yang et al. (2019) (Nonnegative,CP [PD]) Zhao et al. (2023b) (Schatten- p [I]) Nie et al. (2023) (Schatten- p [K])	Baggag et al. (2019) (AR,CP [I] [P]) Bhanu et al. (2020) (LSTM,CP [P]) Jia et al. (2020) (CP [I]) Yu et al. (2025) (TR [I])	Zhou et al. (2015) (CP [I]) Wu et al. (2018) (TV-S,CP [I]) Xie et al. (2024) (Tucker [I]) Shu et al. (2024) (TNN [I]) Gong et al. (2025) (Tucker [I])	Baggag et al. (2019) (G-T,CP [I] [P])	
Variation				Wu et al. (2018) (G-S,CP [I])		
Covariance						Lei et al. (2022b) (CP-B [I] [K]) Lei et al. (2024) (CP-B [R]) Lei et al. (2022a) (CP-B [I]) Lei and Sun (2024) (Residual,CP [I])

Abbreviations: S. / T.: spatial constraint, temporal constraint; [I]: imputation; [E]: extrapolation; [P]: prediction; “-B”: Bayesian model; DF: data fusion; NN: nuclear norm; DPD: dimension preserved decomposition; TD: triple decomposition; TT: tensor train; TR: tensor ring.

7. Evaluation and Benchmark Datasets

In this section, we outline the common metrics used to evaluate LRTL models and introduce the benchmark datasets widely used in spatiotemporal traffic data analysis. Given that LRTL methods can be implemented as deterministic optimization formulations or as probabilistic Bayesian models, the evaluation measures generally fall into two categories: **deterministic metrics**, which assess the accuracy of point mean estimates, and **probabilistic metrics**, which evaluate the quality of uncertainty estimation. Formally, consider a test or unobserved set of length n , with true values $\mathbf{y}_{\Omega^c} \in \mathbb{R}^n$, where Ω^c denotes the indices of test entries, and the model estimates $\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_n]^\top \in \mathbb{R}^n$. Below, we summarize the key performance metrics in each category and then describe the benchmark datasets (see Table 11) used to validate LRTL methods.

7.1. Evaluation analysis

7.1.1. Deterministic metrics

For models that produce point estimates, e.g., through deterministic low-rank modeling or the posterior mean in Bayesian models, performance is typically measured by error statistics comparing predictions \hat{y}_i to ground truth y_i . Common metrics include:

- **Mean absolute error (MAE)**: This is the average absolute deviation between predicted and true values, defined as

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|. \quad (53)$$

MAE is widely used to evaluate traffic data completion accuracy in LRTL studies, related works include Wu et al. (2018); Chen et al. (2018); Xie et al. (2018a); Yang et al. (2020a); Han and He (2020); Deng et al. (2021a); Nie et al. (2022); Li et al. (2022); Lei et al. (2022b,a); Nie et al. (2023); Lei and Sun (2024); Zhang and Wei (2024); Yu et al. (2025). For evaluating forecasting or prediction tasks specifically, studies such as Tan et al. (2016); Xu et al. (2024b); Lin et al. (2024a) also report MAE. In some cases, a normalized MAE (NMAE) is used, for example, Qu et al. (2009); Zhou et al. (2015); Sure et al. (2021); Ming et al. (2024); Zhang et al. (2024) applied NMAE by dividing MAE by the data mean or range for comparability.

- **Root mean squared error (RMSE)**: This metric penalizes larger errors more strongly and is given by

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}. \quad (54)$$

RMSE has been extensively used alongside MAE in traffic data evaluations, such as in Tan et al. (2013); Ran et al. (2016b); Chen et al. (2018); Xie et al. (2018a); Baggag et al. (2019); Chen et al. (2019a,b); Yang et al. (2020a); Gong and Zhang (2020); Chen et al. (2020); Han and He (2020); Sure et al. (2021); Chen et al. (2021a,b); Chen and Sun (2021); Nie et al. (2022); Li et al. (2022); Lei et al. (2022b,a); Xing et al. (2023b); Zhao et al. (2023b); Nie et al. (2023); Li et al. (2023a); Lei and Sun (2024); Xie et al. (2024); Zhang and Wei (2024); Yu et al. (2025); Liu et al. (2025a). For instance, Tan et al. (2013) and many subsequent works report RMSE as a primary metric. It has also been used in prediction task, including the studies in Zhang et al. (2022); Xu et al. (2024b); Lin et al. (2024a). A normalized RMSE (NRMSE) is also occasionally used to account for scale, see e.g., Qu et al. (2009); Baggag et al. (2019).

- **Mean absolute percentage error (MAPE)**: MAPE measures the average percentage error, defined as

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|. \quad (55)$$

MAPE has also been widely applied in evaluations for traffic data completion due to its interpretability in percentage terms. It has been reported in many LRTL studies, e.g., Baggag et al. (2019); Chen et al. (2019a,b); Yang et al. (2020a); Gong and Zhang (2020); Chen et al. (2020, 2021a,b); Chen and Sun (2021); Deng et al. (2021a); Li et al. (2022); Xing et al. (2023b); Zhao et al. (2023b); Li et al. (2023a); Xie et al. (2024); Liu et al. (2025a). In addition, (Tan et al., 2016; Zhang et al., 2022) use MAPE to compare prediction accuracy. A related metric is the mean relative error (MRE), which is similarly defined but not expressed as a percentage; Chen et al. (2018) employed MRE in their study. Some works use other variants of relative error, for instance, relative error measure has been applied in Said and Erradi (2021) for model evaluation.

7.1.2. Probabilistic metrics

Beyond point estimate accuracy, Bayesian low-rank tensor models and other probabilistic approaches provide uncertainty estimates, e.g., predictive distributions or intervals. To evaluate probabilistic predictions, researchers have adopted scoring rules that reward accurate and well-calibrated uncertainty estimates. One of the most widely used in recent literature is the continuous ranked probability score (CRPS) ([Gneiting and Raftery, 2007](#)). In addition, R^2 (coefficient of determination) has also been applied in probability models for model performance estimation. We discuss these two metrics below.

- **Continuous ranked probability score (CRPS):** CRPS generalizes MAE to probabilistic forecasts by measuring the difference between the predicted cumulative distribution function (CDF) and the CDF of the observation, which can be represented as a step function. Formally, let Y be a random variable, F be the CDF of Y , i.e., $F(t) = p(Y \leq t)$, and y be an observation, the CRPS is defined as:

$$\text{CRPS}(F, y) = \int_{-\infty}^{\infty} (F(t) - \mathbb{1}\{t = y\})^2 dt,$$

where $\mathbb{1}\{\cdot\}$ is an indicator function that equals 1 if the condition is true and 0 otherwise. This integral has a closed-form solution for certain distributions ([Gneiting and Raftery, 2007](#)). In particular, if Y is assumed Gaussian $\mathcal{N}(\mu, \sigma^2)$ —as is common in many LRTL probabilistic models—the CRPS can be computed in closed form as:

$$\text{CRPS}(\mathcal{N}(\mu, \sigma^2), y) = \sigma \left[\frac{1}{\sqrt{\pi}} - 2\psi\left(\frac{y - \mu}{\sigma}\right) - \frac{y - \mu}{\sigma} \left(2\Phi\left(\frac{y - \mu}{\sigma}\right) - 1\right) \right],$$

where $\psi(\cdot)$ and $\Phi(\cdot)$ are the probability density function (PDF) and CDF of a standard Gaussian, respectively. Given a set of n test points with predictive mean \hat{y}_i and standard deviation (std.) σ_i for the i th entry ($i = 1, \dots, n$), the average CRPS can be computed as:

$$\text{CRPS} = -\frac{1}{n} \sum_{i=1}^n \sigma_i \left[\frac{1}{\sqrt{\pi}} - 2\psi\left(\frac{y_i - \hat{y}_i}{\sigma_i}\right) - \frac{y_i - \hat{y}_i}{\sigma_i} \left(2\Phi\left(\frac{y_i - \hat{y}_i}{\sigma_i}\right) - 1\right) \right]. \quad (56)$$

Intuitively, CRPS evaluates the squared error in CDF space; it reduces to the MAE in the deterministic case and thus provides a strictly proper scoring rule for probabilistic forecasts. CRPS has gained traction in traffic data imputation and regression studies as a comprehensive metric for uncertainty-aware performance. For example, recent research use CRPS to assess the quality of imputed traffic distributions ([Lei et al., 2022a, 2024](#)).

- **Coefficient of determination (R^2):** Although R^2 is traditionally a deterministic metric for regression, it has also been used in probabilistic LRTL evaluations to summarize the accuracy of predictions, by using the mean of the predictive distribution. R^2 (the fraction of variance explained) is defined as

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

where \bar{y} is the mean of the observed data. A higher R^2 (closer to 1) indicates better fit. Some Bayesian models for traffic data analysis report R^2 values for the predictions to complement other metrics, e.g., the works by [Deng et al. \(2021a\)](#); [Lei et al. \(2024\)](#).

In addition to CRPS and R^2 , other probabilistic evaluation scores have also been used. For example, the interval score (INT) ([Gneiting and Raftery, 2007](#)) and prediction interval coverage (CVG) ([Heaton et al., 2019](#)) are employed to assess the quality of predictive uncertainty. The interval score penalizes predictive intervals that are too narrow or too wide, and whether the true value falls outside the interval; while coverage measures the percentage of times the true value falls within a predicted interval of a given confidence level, which ideally matches the nominal confidence. These metrics are common in the statistical forecasting literature ([Heaton et al., 2019](#)) and have been adopted in some traffic studies (e.g., [Lei et al. \(2022a\)](#)) to evaluate how well calibrated the predictive intervals are for imputation tasks.

7.2. Benchmark datasets

Empirical studies in low-rank tensor learning for traffic data rely on several well-known benchmark datasets. Table 11 provides a comprehensive list of datasets that have been used in prior LRTL works, categorized by data type (traffic speed, traffic flow/volume, OD matrices, etc.), along with their tensor dimensionality, size, and the primary task addressed ([I] imputation, [P] prediction, [E] extrapolation, anomaly detection, etc.). These benchmarks range from small-scale case studies to large-scale metropolitan datasets. Below, we highlight a few of the most widely used traffic datasets in tensor-based research, along with their characteristics:

- **PeMS (California Performance Measurement System)²** (Chen et al., 2001): This is one of the most extensively used sources of traffic data. The PeMS database gathers highway loop detector measurements across California, including traffic volume, speed, and occupancy. Data are recorded every 5 minutes, i.e., 288 time points per day). Many studies formulate PeMS data as a 3-way tensor (e.g., *sensor × time of day × day*) for tasks including traffic imputation and prediction. Depending on the region and time span, the tensor can be quite large. For example, one configuration uses 11,160 sensor stations over 56 days with 288 readings per day, which yields on the order of 10^8 data points. LRTL models have demonstrated the ability to handle such large-scale data through low-rank structure exploitation.
- **Guangzhou urban traffic speed³**: This dataset consists of traffic speed readings from an urban road network in Guangzhou, China. It contains data from 214 road segments over 61 days (August 1 to September 30, 2016) at a 10-minute resolution (144 time slots per day). The data can be naturally represented as a third-order tensor of size $214 \times 144 \times 61$ with structure *road segment × time of day × day*. Guangzhou traffic data have been used in numerous studies on tensor factorization or completion for spatiotemporal traffic imputation, e.g., Chen et al. (2018, 2019a,b); Nie et al. (2022), often serving as a benchmark for comparing model performance under different low-rank assumptions or spatiotemporal constraints.
- **Seattle freeway traffic speed⁴**: This dataset contains traffic speed measurements from 323 loop detectors on highways in the Seattle area (Year 2015), recorded every 5 minutes (288 points per day). A common usage is to focus on a subset, such as one month of data. For example, Lei and Sun (2024) extract the traffic speed tensor for January 2015, which can be structured as $323 \times 288 \times 30$ of *location × time of day × day*—approximately 2.88 million data points. Seattle freeway data have been employed to evaluate LRTL methods for imputation and short-term forecasting, e.g., Chen et al. (2020, 2021b); Lei et al. (2022a); Lei and Sun (2024), among others. It is another example demonstrating that tensor models can scale to large urban networks with hundreds of sensors and high temporal resolution.

From Table 11, one can observe that low-rank tensor techniques are applicable to a diverse array of traffic datasets, including highway sensor networks, urban road network traffic speed, public transit passenger flows, ridesharing trip records, and bike-sharing systems, and to data volumes that are quite large. Many recent LRTL studies handle tensors with millions of entries or higher without issue. For instance, the London traffic speed dataset involves 35,912 road sensors over one month, and the NYC taxi trip dataset involves tens of thousands of OD pairs over multiple years, yet tensor factorization/completion models have been successfully applied in these cases. The ability of low-rank models to decompose high-dimensional tensors enables scalability to such large datasets, provided that an inherent low-rank structure exists.

In summary, rigorous evaluation of LRTL models involves a combination of error metrics for accuracy (MAE, RMSE, MAPE, etc.) and, when applicable, probabilistic scores (CRPS, interval coverage, etc.) for uncertainty. The availability of standardized benchmark datasets (Table 11) covering different traffic scenarios has facilitated consistent comparison across studies. These datasets, together with well-defined evaluation metrics, form the foundation for validating new LRTL methods and gauging progress in the field of ITS.

²<https://pems.dot.ca.gov/>

³<https://doi.org/10.5281/zenodo.1205229>

⁴<https://github.com/zhiyongc/Seattle-Loop-Data>

Table 11: Comparison of benchmark datasets used in LRTL for spatiotemporal traffic data.

Type	Source	Author (year)	Tensor structure	Size	Task
Traffic speed	Guangzhou	Chen et al. (2018)	$location \times day \times time of day$	$214 \times 61 \times 144$	[I]
		Chen et al. (2019a)	$location \times day \times time of day$	$214 \times 61 \times 144$	[I]
		Chen et al. (2019b)	$location \times day \times time of day$	$214 \times 61 \times 144$	[I]
		Gong and Zhang (2020)	$location \times day \times time of day$	$209 \times 61 \times 144$	[I]
		Chen et al. (2020)	$location \times day \times time of day$	$214 \times 61 \times 144$	[I]
		Chen et al. (2021a)	$road segment \times time of day \times day$	$214 \times 144 \times 61$	[I]
		Chen et al. (2021b)	$sensor \times time of day \times day$	$214 \times 144 \times 61$	[I]
		Nie et al. (2022)	$time of day \times location \times day$	$144 \times 214 \times 61$	[I]
		Li et al. (2023a)	$location \times day \times time of day$	$214 \times 61 \times 144$	[I]
		Yu et al. (2025)	$location \times time of day \times day$	$214 \times 144 \times 61$	[I]
		Lin et al. (2024a)	$location \times time of day \times day$	$214 \times 144 \times 61$	[P]
	Madrid highway ^a	Gong and Zhang (2020)	$location \times day \times time of day$	$331 \times 120 \times 96$	[I]
Seattle freeway	Seattle freeway	Chen et al. (2020)	$location \times day \times time of day$	$323 \times 28 \times 288$	[I]
		Chen et al. (2021b)	$location \times time of day \times day$	$323 \times 288 \times 28$	[I]
		Lei et al. (2022a)	$location \times time of day \times day$	$323 \times 288 \times 30$	[I]
		Lei and Sun (2024)	$location \times time of day \times day$	$323 \times 288 \times 30$	[I]
		Xie et al. (2024)	$location \times time of day \times day$	$15 \times 288 \times 28$	[I]
		Chen et al. (2024c)	$location \times day \times time of day$	$323 \times 28 \times 288$	[I]
		Lin et al. (2024a)	$location \times time of day \times day$	$323 \times 288 \times 28$	[P]
PeMS	PeMS	Chen et al. (2021a)	$sensor \times time of day \times day$	$11160 \times 288 \times 56$	[I]
		Lei et al. (2022a)	$location \times time of day \times day$	$319 \times 144 \times 59$	[I]
		Nie et al. (2023)	$time of day \times location \times day$	$288 \times 11160 \times 28$	[E]
		Lei and Sun (2024)	$location \times time of day \times day$	$319 \times 144 \times 28$	[I]
London ^b	Chen et al. (2021a)		$sensor \times time of day \times day$	$35912 \times 24 \times 30$	[I]
	Shenzhen	Yu et al. (2025)	$location \times time of day \times day$	$2545 \times 288 \times 7$	[I]
Traffic flow/volume	PeMS	Tan et al. (2013)	$day \times hour per day \times time per hour$	$16 \times 24 \times 12$	[I]
		Ran et al. (2016b)	$week \times day of week \times time of day$	$11 \times 7 \times 288$	[I]
		Gong and Zhang (2020)	$week \times day of week \times time of day \times sensor$	$11 \times 7 \times 288 \times 11$	[I]
		Deng et al. (2021a)	$location \times day \times time of day$	$9 \times 49 \times 288$	[I]
		Chen et al. (2024c)	$time of day \times day \times sensor$	$288 \times 59 \times 307$	[I]
		Yu et al. (2025)	$time of day \times day \times sensor$	$288 \times 62 \times 170$	[I]
		Lin et al. (2024a)	$location \times day \times time of day$	$228 \times 44 \times 288$	[I]
			$location \times time of day \times day$	$170 \times 288 \times 62$	[I]
			$location \times time of day \times day$	$11160 \times 288 \times 84$	[P]
	Portland highway ^c	Chen et al. (2021b)	$location \times time of day \times day$	$1156 \times 96 \times 31$	[I]
		Nie et al. (2022)	$time of day \times location \times day$	$96 \times 1156 \times 31$	[I]
		Nie et al. (2023)	$time of day \times location \times day$	$96 \times 1057 \times 90$	[I]
		Chen et al. (2024c)	$location \times day \times time of day$	$1156 \times 31 \times 96$	[I]
Hangzhou metro ^d	Hangzhou metro ^d	Chen et al. (2020)	$location \times day \times time of day$	$80 \times 25 \times 108$	[I]
		Zhang et al. (2024)	$time of day \times station \times day$	$108 \times 80 \times 25$	[I]
		Chen et al. (2024c)	$location \times day \times time of day$	$80 \times 25 \times 108$	[I]
Guangzhou metro	Guangzhou metro	Han and He (2020)	$station \times day \times time of day$	$148 \times 14 \times 96$	[I] [PD]
	Abilene ^e	Zhou et al. (2015)	$origin \times destination \times time per week$	$11 \times 11 \times 1008$	[I]
OD pairs	GÉANT ^f	Xie et al. (2018a)	$OD pair \times time of day \times day$	$144 \times 288 \times 168$	[I]
		Zhang et al. (2024)	$time of day \times OD pair \times day$	$288 \times 121 \times 21$	[I]
		Zhou et al. (2015)	$origin \times destination \times time per week$	$23 \times 23 \times 672$	[I]
Xuhui, Shanghai	Xuhui, Shanghai	Xie et al. (2018a)	$OD pair \times time of day \times day$	$529 \times 96 \times 112$	[I]
		Zhang et al. (2024)	$time of day \times OD pair \times day$	$96 \times 529 \times 36$	[I]
Occupancy	Birmingham ^g	Nie et al. (2022)	$time of day \times OD pairs \times day$	$64 \times 289 \times 14$	[I]
	NYC, taxi ^h	Chen et al. (2020)	$location \times day \times time of day$	$30 \times 77 \times 18$	[I]
Trip record (Demand)	NYC, taxi ^h	Nie et al. (2022)	$time of day \times location \times day$	$18 \times 30 \times 77$	[I]
		Chen and Sun (2021)	$zone \times zone \times time slot$	$30 \times 30 \times 1464$	[I] [P]
		Chen et al. (2023a)	$zone \times zone \times time slot$	$69 \times 69 \times 3653$	[PD]
		Sofuo glu and Aviyente (2022)	$hour per day \times day per week \times week \times zone$	$24 \times 7 \times 52 \times 81$	[AD]
Montreal, BIXI ⁱ	Montreal, BIXI ⁱ	Lei et al. (2024)	$station \times day \times covariate$	$587 \times 196 \times 19$	[R]
Trajectory	Chengdu	Yu et al. (2020)	$time intervals \times links$	588×114	[I]

^a <https://datos.madrid.es/portal/site/egob/>.^b <https://movement.uber.com>.^c <https://portal.its.pdx.edu/home>.^d <https://tianchi.aliyun.com/competition/entrance/231708/information>.^e <https://www.cs.bu.edu/fac/crovella/abilene-distro.tar>.^f <https://totem.info.ucl.ac.be/dataset.html>.^g Car parking, <https://archive.ics.uci.edu/ml/datasets/Parking+Birmingham>.^h <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>.ⁱ Bike-sharing service, <https://bixi.com>.

8. Challenges and Future Directions

8.1. Challenges

Despite the growing success of LRTL in modeling spatiotemporal traffic data, several important challenges remain:

Rank determination. Determining the appropriate tensor rank is critical yet non-trivial. An underestimated rank leads to underfitting and poor reconstruction, while an overestimated rank increases computational cost and risks overfitting. Automatic rank determination methods, e.g., Bayesian approaches using multiplicative gamma priors (MGP) are promising but not yet widely adopted in practice, and effective heuristics or priors for rank selection remain an open issue.

Computational scalability. Although LRTL methods are generally more scalable than Gaussian-process-based models for probabilistic modeling, real-world transportation datasets—such as high-resolution OD matrices, multi-year multi-modal demand tensors, or full-city traffic state tensors at fine temporal resolution—can still be prohibitively large and easily exceed the memory and time budgets of standard algorithms. For deterministic LRTL, recent work has explored distributed and out-of-core optimization (e.g., parallel ALS on clusters or MapReduce-style implementations), streaming and online tensor updates that incrementally refine low-rank factors as new data arrive, and randomized tensor sketching techniques that operate on compressed tensor representations (Ma et al., 2021). However, these approaches typically provide only point estimates and do not directly support probabilistic LRTL or scalable uncertainty quantification. In the probabilistic setting, one instead needs scalable Bayesian inference schemes—such as variational Bayes, stochastic variational inference, or distributed MCMC—that exploit low-rank, sparse, or Kronecker structure. Designing such methods so that they remain computationally tractable while faithfully preserving rich spatiotemporal correlations and yielding reliable UQ at city scale remains a central open challenge for scalable LRTL in transportation.

Cold-start and high missing rates. When entire spatial regions or long time intervals are missing—so-called “cold-start” scenarios—the tensor model becomes severely under-constrained. While incorporating spatial graph priors and temporal dynamics can help infer unobserved entries, fully unobserved locations or time periods remain difficult to recover without external side information or additional constraints. Developing methods to better handle extremely sparse scenarios or to transfer knowledge from other cities/datasets is still a pressing challenge.

Noise robustness and heteroscedasticity. Traffic data often exhibit heteroscedastic, bursty, and non-Gaussian noise, e.g., higher variance during peak hours or anomalies under sensor faults. Many existing LRTL models assume i.i.d. Gaussian noise, which limits their robustness to outliers and varying noise levels. Extensions using heavy-tailed distributions, e.g., Student-t, robust loss functions, e.g., Huber or ℓ_1 norms, or data-confidence weighting strategies, e.g., the CP-WOPT method (Acar et al., 2011), have shown promise, but achieving high fidelity under all noise conditions remains difficult. Handling heteroscedastic noise and outliers in a principled way is an ongoing area of research.

Non-Gaussian data characteristics. A significant portion of traffic data, such as traffic counts, congestion categories, and occupancy levels, are discrete, heavy-tailed, or exhibit zero inflation, which violates the Gaussian assumptions used in most LRTL models. Classical low-rank frameworks commonly rely on squared-error losses or assume Gaussian noise, limiting their applicability to non-Gaussian or non-continuous data distributions. This leads to biased reconstructions, underestimated uncertainty, and poor performance on count-based or categorical variables. Moreover, traditional Gaussian-based models struggle to handle overdispersion and skewed distributions often present in urban traffic data during peak hours or anomalous events. Accurately modeling such non-Gaussian characteristics within a low-rank or probabilistic tensor framework remains a technical and computational challenge, particularly under high missingness or sparse observations.

Parameter tuning and model selection. Low-rank tensor learning models often involve multiple hyperparameters, such as regularization weights, kernel bandwidths, or noise variances, that critically influence model performance. Tuning these parameters is typically done via heuristic search, cross-validation, or trial-and-error, which can be computationally prohibitive for large-scale traffic tensors or under high missingness. Moreover, regularization terms, such as graph-based Laplacians, smoothness penalties, or temporal dynamics, often require careful calibration to balance flexibility with overfitting prevention. The challenge is

further compounded when integrating multiple side-information sources, e.g., spatial graphs and autoregressive priors, as their associated penalties interact in complex ways. There is currently a lack of principled, efficient, and task-specific methods for tuning these regularization parameters, especially in the presence of noisy, sparse, or heterogeneous traffic data.

Dynamic networks and events. Road networks and traffic patterns are not static: incidents, roadwork, and policy changes can dynamically alter traffic flows. However, most tensor models assume a fixed spatial topology and stationary processes. Modeling time-varying road network structure, e.g., dynamic graph Laplacians that change with road closures or new links, integrating online event detection to flag unusual patterns or regime shifts, or combining rule-based traffic management interventions with low-rank estimation, remains largely unexplored. Capturing these dynamic changes in the tensor model without retraining from scratch is an important challenge for real-world deployment.

Model representation limits. Even with expressive spatiotemporal regularizers, purely low-rank models struggle to capture highly localized or high-frequency patterns in traffic data unless the chosen rank is very high. Complex nonlinear correlations or multi-scale phenomena, such as short-term congestion “shockwaves” superimposed on daily or weekly cycles, may not be fully captured by the linear factorization structure. Complementary frameworks that capture residual structure, e.g., adding sparse Gaussian process components or localized neural modules on top of the global low-rank model, can be essential to overcome this limitation. Balancing the simplicity and interpretability of low-rank representations with the flexibility to model fine-grained anomalies or nonlinear effects is an ongoing challenge.

8.2. Future directions

Building on the above challenges, several promising research directions are emerging:

Deep learning integration. Rather than viewing deep neural networks as competitors to LRTL, they can be used in tandem to enhance tensor models. Graph neural networks (GNNs), temporal convolutional networks (TCNs), transformers, and LSTMs can extract complex nonlinear spatiotemporal features that serve as side information, initialization, or inputs for LRTL models. Conversely, enforcing a low-rank structure within deep models, e.g., constraining weight matrices of a GNN to lie in a low-rank subspace, can act as a form of regularization to improve generalization. Such hybrid approaches are beginning to appear. For example, multitask neural tensor factorization (Wu et al., 2022c) combines neural network components with tensor factorization. This synergy between deep learning and low-rank modeling is likely to continue growing.

Pretraining and foundation models. Inspired by recent successes in natural language and computer vision, large-scale spatiotemporal foundation models for traffic data can be promising. The idea is to pre-train a universal model on massive, heterogeneous traffic datasets (potentially via self-supervision) to capture transferable traffic dynamics, and then adapt to specific tasks or cities. Early efforts such as OpenCity (Li et al., 2024e) target zero-/few-shot forecasting by learning representations that generalize across regions and conditions. Integrating LRTL into these foundation models, for example, using tensor structure in the model architecture or loss function, could provide interpretability and parameter efficiency, while the broad pretraining provides robustness to distribution shifts and rare events. In addition, large language models (LLMs) can align textual incident reports, policies, and operator notes with numeric tensor states, translate probabilistic forecasts into human-readable guidance, and propose candidate control policies subject to LRTL constraints. Retrieval-augmented and tool-use paradigms (e.g., calling an LRTL forecaster from an LLM) provide practical pathways that preserve the strengths of both approaches.

Complementary modeling frameworks. To improve representation power without excessively increasing the tensor rank, residual or hybrid modeling frameworks are a promising avenue. For instance, BCKL (Lei et al., 2022a) combines a global low-rank tensor factorization with local sparse Gaussian process components that model the remaining structure. This two-tier model allows high-frequency or highly nonlinear signals to be captured as “residuals” on top of a low-rank approximation, preserving scalability and interpretability. In general, combining tensor models with complementary methods, e.g., kernel machines or small neural networks focusing on the residual patterns, can address the representation limits noted above without sacrificing the core advantages of LRTL.

Incorporation of exogenous and multi-modal data. Traffic is influenced by many external factors beyond the inherent spatiotemporal patterns. For example, weather conditions, holidays/events, road closures, or public policies. Systematically incorporating such exogenous variables as additional tensor modes, features, or regularization terms is an ongoing research effort. Multi-modal data fusion is another frontier: modern ITS collect not only point traffic sensor readings but also rich data including camera images, GPS trajectories, incident reports, and more. Incorporating heterogeneous data sources (visual, textual, or other sensor modalities) into a unified low-rank tensor framework (potentially via coupled tensors or tensor regression models) could greatly enrich the model context and robustness. However, effective multi-modal fusion will require handling differing data formats, resolutions, and noise characteristics. Developing LRTL models that can seamlessly integrate exogenous and multi-modal information is key to improving both the accuracy and the scope of traffic analytics, e.g., jointly modeling traffic flow with weather or event data for improved prediction and anomaly detection.

Physics-informed tensor models. Traffic dynamics are governed by well-understood physical principles, such as flow conservation laws and car-following behavior. Embedding such domain knowledge into LRTL is a promising direction to improve model realism and reliability. Recent work has started to encode physical constraints directly into tensor models. For instance, by designing spatial factor constraints that reflect one-way road directions or conserved flow between adjacent road segments, or by introducing priors inspired by partial differential equation (PDE) models of traffic flow. By making tensor factors obey known traffic flow relationships (possibly through physics-informed regularizers or architectures), one can ensure that the learned representations respect fundamental system behavior. This integration of theory and data could yield models that not only fit historical data well but also remain stable under out-of-distribution conditions or sudden changes such as network disruptions. Exploring neural-network-based physics emulation, such as neural ODE/PDE models, in combination with low-rank structures is a fruitful avenue to bridge analytical traffic flow models and data-driven learning.

Neural operator approaches and continuous modeling. Moving beyond discrete tensor representations, another emerging trend is to apply neural operators to traffic modeling. Neural operators, such as the Fourier neural operator (Li et al., 2020b), are designed to learn mappings between functions in infinite-dimensional spaces, and they have shown success in capturing the behavior of systems governed by PDEs. In the context of traffic, neural operator frameworks can model the evolution of traffic states as continuous spatiotemporal fields, potentially capturing fine-grained dynamics that a fixed tensor grid might miss. These approaches could complement LRTL by generalizing across different spatial or temporal resolutions and by naturally incorporating continuous physical dynamics. For example, a neural operator could learn the continuous propagation of congestion waves on a freeway, as described by fluid-dynamics-based traffic models, and a low-rank tensor factorization could then be used to approximate or constrain this complex behavior for scalability. Although still in early stages, this direction opens a new avenue to blend data-driven learning with traditional traffic flow theory, potentially leading to more powerful and generalizable models.

Adaptive and online learning. Real-time traffic management demands models that update continuously as new data arrive, rather than relying on periodic batch retraining. Developing LRTL methods for streaming data, online learning, or incremental model updates is therefore crucial for deployment in ITS. Some initial efforts exist: For example, streaming tensor decomposition algorithms that update factor matrices on the fly, online Bayesian inference that refines tensor estimates with each new time window, and mechanisms for dynamically adjusting the tensor rank as data patterns evolve. These strategies enable the model to handle concept drift, e.g., sudden shifts in traffic patterns due to accidents or long-term changes due to infrastructure adjustments, and to maintain performance without frequent full retraining. Future research will likely focus on improving the stability and speed of such online LRTL algorithms, ensuring they can scale to high-frequency sensor feeds and city-scale networks while preserving accuracy. The ability for a low-rank tensor model to “learn on its feet” opens up new possibilities for real-time anomaly detection, adaptive traffic signal control, and other closed-loop ITS applications where the model must continuously reflect the current state of the system.

Toward automated and adaptive tuning. As tensor models grow in complexity, with multiple regularization components, probabilistic priors, and side information, manual hyperparameter tuning becomes impractical and unsustainable. Future research should explore automated parameter selection frameworks for

LRTL, such as Bayesian evidence maximization, hierarchical modeling, e.g., hyperpriors over regularization coefficients, or bilevel optimization where parameters are learned jointly with the factorization. Meta-learning approaches, where models learn to adapt hyperparameters from related traffic tasks or datasets, may also help improve generalization and reduce tuning costs. Adaptive regularization strategies that adjust penalty weights based on data characteristics, e.g., missingness pattern, variance levels, or temporal instability, could further improve model robustness. These automated tuning methods are essential for building scalable, plug-and-play LRTL systems suitable for real-time or cross-city deployment in intelligent transportation applications.

9. Conclusion

In this paper, we present a comprehensive survey of Low-Rank Tensor Learning (LRTL) for spatiotemporal traffic data (Figure 1). As traffic data becomes increasingly large-scale, high-dimensional, noisy, and incomplete, LRTL provides a principled and scalable framework to uncover latent patterns and recover missing information by exploiting inherent multi-way correlations in transportation systems.

We review both deterministic LRTL approaches (Section 3)—including tensor decompositions and rank-surrogate-based completion—and Bayesian probabilistic LRTL methods (Section 4). Across these frameworks, we examined how spatial (e.g., road-network topology) and temporal side information (e.g., diurnal/weekly periodicity, autoregressive dynamics) can be embedded into LRTL to enhance performance. We then discuss LRTL solutions for core transportation tasks (Section 5). We furthermore summarize common spatiotemporal constraints in LRTL, with parallel deterministic regularizer and Bayesian prior formulations, tied to transportation scenarios (Section 6). We also synthesized benchmark datasets that are widely used in LRTL studies for transportation, and evaluation practices from both deterministic and probabilistic perspectives (Section 7). We highlight practical challenges and future research directions (Section 8). An overall taxonomy of this survey see Figure 2.

To summarize the main contributions and insights:

- 1) We analyze LRTL for spatiotemporal traffic data from deterministic (Section 3) and Bayesian probabilistic (Section 4) viewpoints, with a focus on spatiotemporal modeling; the taxonomy is provided in Figure 2. Deterministic LRTL techniques (Figure 4) cover classical tensor decompositions (CP, Tucker, etc., Table 3) and convex/nonconvex rank-surrogate models (e.g., tensor nuclear norms, Table 5). Bayesian tensor factorization methods infer posterior distributions over tensor structure and quantified uncertainty (representative models are compared in Figure 7). We clarified how spatiotemporal correlations are incorporated as regularizer in deterministic formulations and as prior in Bayesian settings, summarizing related LRTL research for spatiotemporal traffic data. A guiding principle of choosing LRTL methods is to balance model effectiveness (deterministic accuracy, calibrated uncertainty) and computational efficiency (theoretical and wall-clock cost).
- 2) We reviewed LRTL across core transportation data analysis tasks (Section 5, Figure 8): missing data [I]mputation (Figure 9), spatiotemporal [E]xtrapolation (Figure 10), temporal [P]rediction (Figure 11), and other tasks such as pattern discovery [PD], anomaly detection [AD], and regression [R]. For each, we summarize challenges, requirements, and example models with appropriate constraints (Table 7), and compare LRTL studies across tasks (Table 8). Implementation trade-offs and practical choices are discussed in this section.
- 3) We consolidate spatial and temporal constraint families used in LRTL for traffic data analysis (Section 6, Table 10)—graph-based similarity, variation-based smoothness, and covariance/kernel-based modeling—and show their deterministic regularizer formulations vs. Bayesian probabilistic prior correspondences, along with transportation task linkages (Table 9, Figure 12). These constraints encode domain knowledge (network connectivity, continuity, periodicity, dynamics) and are central to LRTL model performance.
- 4) We organize benchmark traffic datasets commonly used in LRTL studies (Table 11), and summarize evaluation metrics for deterministic and probabilistic models (Section 7). Section 8 outlines open challenges (e.g., scalability, adaptive/online learning, combining exogenous signals, automated rank and hyperparameter selection, reproducibility) and promising directions in LRTL for traffic data (e.g., hybrid deep-low-rank models, residual/nonlinear components, learned graphs/priors, physics-informed constraints).

We hope this survey offers researchers and practitioners a structured overview of the methodological landscape and practical guidance for applying LRTL to traffic data and furthermore ITS contexts. As traffic data continue to grow in volume, complexity, and modality, we believe that LRTL—especially when augmented with domain knowledge and modern machine learning techniques—will remain a robust and evolving foundation for spatiotemporal traffic data analytics.

References

- Abadi, A., Rajabioun, T., Ioannou, P.A., 2014. Traffic flow prediction for road transportation networks with limited traffic data. *IEEE Transactions on Intelligent Transportation Systems* 16, 653–662.
- Acar, E., Dunlavy, D.M., Kolda, T.G., Mørup, M., 2011. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems* 106, 41–56.
- Afshar, A., Ho, J.C., Dilkina, B., Perros, I., Khalil, E.B., Xiong, L., Sunderam, V., 2017. Cp-ortho: An orthogonal tensor factorization framework for spatio-temporal data, in: Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 1–4.
- Afshar, A., Perros, I., Papalexakis, E.E., Searles, E., Ho, J., Sun, J., 2018. Copa: Constrained parafac2 for sparse & large datasets, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 793–802.
- Ahmad Jan, M., Adil, M., Brik, B., Harous, S., Abbas, S., 2025. Making sense of big data in intelligent transportation systems: Current trends, challenges and future directions. *ACM Computing Surveys* 57, 1–43.
- Ahmadi-Asl, S., Abukhovich, S., Asante-Mensah, M.G., Cichocki, A., Phan, A.H., Tanaka, T., Oseledets, I., 2021. Randomized algorithms for computation of tucker decomposition and higher order svd (hosvd). *IEEE Access* 9, 28684–28706.
- Allen, G.I., Grosenick, L., Taylor, J., 2014. A generalized least-square matrix decomposition. *Journal of the American Statistical Association* 109, 145–159.
- Azarhoosh, Z., Ghazaan, M.I., 2025. A review of recent advances in surrogate models for uncertainty quantification of high-dimensional engineering applications. *Computer Methods in Applied Mechanics and Engineering* 433, 117508.
- Babacan, S.D., Luessi, M., Molina, R., Katsaggelos, A.K., 2012. Sparse bayesian methods for low-rank matrix estimation. *IEEE Transactions on Signal Processing* 60, 3964–3977.
- Bae, B., Kim, H., Lim, H., Liu, Y., Han, L.D., Freeze, P.B., 2018. Missing data imputation for traffic flow speed using spatio-temporal cokriging. *Transportation Research Part C: Emerging Technologies* 88, 124–139.
- Baggag, A., Abbar, S., Sharma, A., Zanouda, T., Al-Homaid, A., Mohan, A., Srivastava, J., 2019. Learning spatiotemporal latent factors of traffic via regularized tensor factorization: Imputing missing values and forecasting. *IEEE Transactions on Knowledge and Data Engineering* 33, 2573–2587.
- Bahadori, M.T., Yu, Q.R., Liu, Y., 2014. Fast multivariate spatio-temporal analysis via low rank tensor learning. *Advances in Neural Information Processing Systems*, 3491–3499.
- Banerjee, S., Carlin, B.P., Gelfand, A.E., 2014. Hierarchical modeling and analysis for spatial data. CRC press.
- Banerjee, S., Gelfand, A.E., Finley, A.O., Sang, H., 2008. Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70, 825–848.
- Battaglino, C., Ballard, G., Kolda, T.G., 2018. A practical randomized cp tensor decomposition. *SIAM Journal on Matrix Analysis and Applications* 39, 876–901.
- Bhanu, M., Mendes-Moreira, J., Chandra, J., 2020. Embedding traffic network characteristics using tensor for improved traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* 22, 3359–3371.

- Bi, X., Li, W., Sun, Y., Qi, S., Wang, Z., 2025. Leveraging multiple source cities in selective transfer learning for traffic prediction with limited data. *IEEE Transactions on Intelligent Transportation Systems* 26, 15646 – 15659.
- Borchani, H., Varando, G., Bielza, C., Larranaga, P., 2015. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5, 216–233.
- Cai, J.F., Li, J., Xia, D., 2022. Provable tensor-train format tensor completion by riemannian optimization. *Journal of Machine Learning Research* 23, 1–77.
- Candes, E., Recht, B., 2012. Exact matrix completion via convex optimization. *Communications of the ACM* 55, 111–119.
- Carroll, J.D., Chang, J.J., 1970. Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young” decomposition. *Psychometrika* 35, 283–319.
- Cemgil, A.T., 2009. Bayesian inference for nonnegative matrix factorisation models. *Computational intelligence and neuroscience* 2009, 785152.
- Chen, C., Petty, K., Skabardonis, A., Varaiya, P., Jia, Z., 2001. Freeway performance measurement system: mining loop detector data. *Transportation research record* 1748, 96–102.
- Chen, H., Lin, M., Liu, J., Yang, H., Zhang, C., Xu, Z., 2024a. Nt-dptc: a non-negative temporal dimension preserved tensor completion model for missing traffic data imputation. *Information Sciences* 653, 119797.
- Chen, H., Lin, M., Zhao, L., Xu, Z., Luo, X., 2025a. Fourth-order dimension preserved tensor completion with temporal constraint for missing traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems* 26, 6734–6748.
- Chen, P., Li, F., Wei, D., Lu, C., 2024b. Low-rank and deep plug-and-play priors for missing traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems* 26, 2690–2706.
- Chen, P., Li, F., Wei, D., Lu, C., 2024c. Spatiotemporal traffic data completion with truncated minimax-concave penalty. *Transportation Research Part C: Emerging Technologies* 164, 104657.
- Chen, X., Chen, Y., Saunier, N., Sun, L., 2021a. Scalable low-rank tensor learning for spatiotemporal traffic data imputation. *Transportation research part C: emerging technologies* 129, 103226.
- Chen, X., Cheng, Z., Cai, H., Saunier, N., Sun, L., 2024d. Laplacian convolutional representation for traffic time series imputation. *IEEE Transactions on Knowledge and Data Engineering* 36, 6490–6502.
- Chen, X., He, Z., Chen, Y., Lu, Y., Wang, J., 2019a. Missing traffic data imputation and pattern discovery with a bayesian augmented tensor factorization model. *Transportation Research Part C: Emerging Technologies* 104, 66–77.
- Chen, X., He, Z., Sun, L., 2019b. A bayesian tensor decomposition approach for spatiotemporal traffic data imputation. *Transportation research part C: emerging technologies* 98, 73–84.
- Chen, X., He, Z., Wang, J., 2018. Spatial-temporal traffic speed patterns discovery and incomplete data recovery via svd-combined tensor decomposition. *Transportation research part C: emerging technologies* 86, 59–77.
- Chen, X., Lei, M., Saunier, N., Sun, L., 2021b. Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems* 23, 12301–12310.
- Chen, X., Liang, S., Zhang, Z., Zhao, F., 2022. A novel spatiotemporal data low-rank imputation approach for traffic sensor network. *IEEE Internet of Things Journal* 9, 20122–20135.
- Chen, X., Sun, L., 2021. Bayesian temporal factorization for multidimensional time series prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 4659–4673.
- Chen, X., Wang, K., Ye, Q., 2025b. Nonconvex low-tubal-rank tensor completion with temporal regularization for spatiotemporal traffic data recovery. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1–14.

- Chen, X., Wang, K., Zhao, F., Deng, F., Ye, Q., 2024e. Composite nonconvex low-rank tensor completion with joint structural regression for traffic sensor networks data recovery. *IEEE Transactions on Computational Social Systems* 11, 6882–6896.
- Chen, X., Xu, N., Wan, K., 2025c. Dual-domain low-rank tensor completion for traffic data recovery. *Applied Mathematical Modelling* , 116404.
- 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.
- Chen, X., Zhang, C., Chen, X., Saunier, N., Sun, L., 2023a. Discovering dynamic patterns from spatiotemporal data with time-varying low-rank autoregression. *IEEE Transactions on Knowledge and Data Engineering* 36, 504–517.
- Chen, X., Zhang, C., Zhao, X.L., Saunier, N., Sun, L., 2025d. Forecasting sparse movement speed of urban road networks with nonstationary temporal matrix factorization. *Transportation Science* 59, 451–687.
- Chen, X., Zhao, X.L., Cheng, C., 2024f. Forecasting urban traffic states with sparse data using hankel temporal matrix factorization. *INFORMS Journal on Computing* 37, 785–1141.
- Chen, Y., Cheng, L., Wu, Y.C., 2023b. Bayesian low-rank matrix completion with dual-graph embedding: Prior analysis and tuning-free inference. *Signal Processing* 204, 108826.
- Cheng, L., Wu, Y.C., Poor, H.V., 2016. Probabilistic tensor canonical polyadic decomposition with orthogonal factors. *IEEE Transactions on Signal Processing* 65, 663–676.
- Cheng, X., Kong, W., Luo, X., Qin, W., Zhang, F., Wang, J., 2024. Tensor completion via joint reweighted tensor q-nuclear norm for visual data recovery. *Signal Processing* 219, 109407.
- Cheng, Z., Trépanier, M., Sun, L., 2022. Real-time forecasting of metro origin-destination matrices with high-order weighted dynamic mode decomposition. *Transportation science* 56, 904–918.
- Choi, J.H., Vishwanathan, S., 2014. Dfacto: Distributed factorization of tensors. *Advances in Neural Information Processing Systems* 27.
- Cichocki, A., Mandic, D., De Lathauwer, L., Zhou, G., Zhao, Q., Caiafa, C., Phan, H.A., 2015. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE signal processing magazine* 32, 145–163.
- De Lathauwer, L., De Moor, B., Vandewalle, J., 2000. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications* 21, 1253–1278.
- Deng, D., Shahabi, C., Demiryurek, U., Zhu, L., Yu, R., Liu, Y., 2016. Latent space model for road networks to predict time-varying traffic, in: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1525–1534.
- Deng, L., Liu, X.Y., Zheng, H., Feng, X., Chen, Y., 2021a. Graph spectral regularized tensor completion for traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems* 23, 10996–11010.
- Deng, T., Wan, M., Shi, K., Zhu, L., Wang, X., Jiang, X., 2021b. Short term prediction of wireless traffic based on tensor decomposition and recurrent neural network. *SN Applied Sciences* 3, 779.
- Duan, Z., 2025. Outlier traffic flow detection and pattern analysis under unplanned disruptions: A low-rank robust decomposition model. *IEEE Access* 13, 91552–91563.
- Fan, J., Sitek, K., Chandrasekaran, B., Sarkar, A., 2022. Bayesian tensor factorized vector autoregressive models for inferring granger causality patterns from high-dimensional multi-subject panel neuroimaging data. *arXiv preprint arXiv:2206.10757*.
- Favier, G., de Almeida, A.L., 2014. Overview of constrained parafac models. *EURASIP Journal on Advances in Signal Processing* 2014, 1–25.
- Feng, L., Liu, Y., Liu, Z., Zhu, C., 2024. Online nonconvex robust tensor principal component analysis. *IEEE Transactions on Neural Networks and Learning Systems* 36, 14384–14398.

- Finley, A.O., Banerjee, S., 2020. Bayesian spatially varying coefficient models in the spbayes r package. *Environmental Modelling & Software* 125, 104608.
- Gandy, S., Recht, B., Yamada, I., 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse problems* 27, 025010.
- Gao, Q., Zhang, P., Xia, W., Xie, D., Gao, X., Tao, D., 2020. Enhanced tensor rpca and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 2133–2140.
- Gelfand, A.E., Kim, H.J., Sirmans, C., Banerjee, S., 2003. Spatial modeling with spatially varying coefficient processes. *Journal of the American Statistical Association* 98, 387–396.
- Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B., 2013. Bayesian data analysis. CRC press.
- Gneiting, T., Raftery, A.E., 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association* 102, 359–378.
- Golyandina, N., 2020. Particularities and commonalities of singular spectrum analysis as a method of time series analysis and signal processing. *Wiley Interdisciplinary Reviews: Computational Statistics* 12, e1487.
- Golyandina, N., Korobeynikov, A., Zhigljavsky, A., 2018. Singular spectrum analysis with R. Springer.
- Gong, C., Zhang, Y., 2020. Urban traffic data imputation with detrending and tensor decomposition. *IEEE Access* 8, 11124–11137.
- Gong, W., Huang, Z., Lu, J., Yang, L., 2025. Tuckerapp: A novel spatiotemporal tucker decomposition approach for traffic imputation. *Signal Processing* 240, 110383.
- Gong, W., Huang, Z., Yang, L., 2023. Accurate regularized tucker decomposition for image restoration. *Applied Mathematical Modelling* 123, 75–86.
- 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.
- Goulart, J.H.d.M., Boizard, M., Boyer, R., Favier, G., Comon, P., 2015. Tensor cp decomposition with structured factor matrices: Algorithms and performance. *IEEE Journal of Selected Topics in Signal Processing* 10, 757–769.
- Grasedyck, L., 2010. Hierarchical singular value decomposition of tensors. *SIAM journal on matrix analysis and applications* 31, 2029–2054.
- Gu, S., Xie, Q., Meng, D., Zuo, W., Feng, X., Zhang, L., 2017. Weighted nuclear norm minimization and its applications to low level vision. *International journal of computer vision* 121, 183–208.
- Gu, S., Zhang, L., Zuo, W., Feng, X., 2014. Weighted nuclear norm minimization with application to image denoising, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2862–2869.
- Guhaniyogi, R., Qamar, S., Dunson, D.B., 2017. Bayesian tensor regression. *Journal of Machine Learning Research* 18, 1–31.
- Gürsun, G., Crovella, M., 2012. On traffic matrix completion in the internet, in: *Proceedings of the 2012 internet measurement conference*, pp. 399–412.
- Han, Y., He, Z., 2020. Simultaneous incomplete traffic data imputation and similarity pattern discovery with bayesian nonparametric tensor decomposition. *Journal of advanced transportation* 2020, 8810753.
- Han, Y., Yang, D., Zhang, C.H., Chen, R., 2024. Cp factor model for dynamic tensors. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 86, 1383–1413.
- Harshman, R.A., et al., 1970. Foundations of the parafac procedure: Models and conditions for an “explatory” multi-modal factor analysis. *UCLA working papers in phonetics* 16, 84.

- Håstad, J., 1990. Tensor rank is np-complete. *Journal of algorithms* 11, 644–654.
- He, C., Xu, Y., Wu, Z., Zheng, S., Wei, Z., 2024. Multi-dimensional visual data restoration: Uncovering the global discrepancy in transformed high-order tensor singular values. *IEEE Transactions on Image Processing* 33, 6409–6424.
- He, Y., Jia, Y., Hu, L., An, C., Lu, Z., Xia, J., 2022. A parameter-free nonconvex low-rank tensor completion model for spatiotemporal traffic data recovery. *arXiv preprint arXiv:2209.13786*.
- Heaton, M.J., Datta, A., Finley, A.O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R.B., Hammerling, D., Katzfuss, M., et al., 2019. A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological and Environmental Statistics* 24, 398–425.
- Hu, L., Chen, W., Liu, Y., Qu, X., Ye, Z., 2024. Vehicle state recovery: A tailored laplace function-based tensor completion approach. *IEEE Transactions on Intelligent Vehicles* 10, 4038–4051.
- Hu, L., Jia, Y., Chen, W., Wen, L., Ye, Z., 2023. A flexible and robust tensor completion approach for traffic data recovery with low-rankness. *IEEE Transactions on Intelligent Transportation Systems* 25, 2558–2572.
- Hu, Y., Zhang, D., Ye, J., Li, X., He, X., 2012. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 2117–2130.
- Huang, L., Zhu, Y., Shao, H., Tang, L., Zhu, Y., Yu, G., 2025a. Bayesian robust tensor decomposition based on mcmc algorithm for traffic data completion. *IET Signal Processing* 2025, 4762771.
- Huang, R., Gong, W., Lu, J., Huang, Z., Yang, L., 2024a. Bacp: Bayesian augmented cp factorization for traffic data imputation, in: *International Conference on Intelligent Computing*, Springer. pp. 108–120.
- Huang, Z., Qiu, Y., Chen, X., Sun, W., Zhou, G., 2024b. Bayesian robust tensor ring decomposition for incomplete multiway data. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 54, 4005–4018.
- Huang, Z., Zhou, G., Qiu, Y., Chen, X., Zhao, Q., 2025b. Kernel bayesian tensor ring decomposition for multiway data recovery. *Neural Networks* , 107500.
- Jacob, M., Mani, M.P., Ye, J.C., 2020. Structured low-rank algorithms: Theory, magnetic resonance applications, and links to machine learning. *IEEE Signal Processing Magazine* 37, 54–68.
- Jia, R., Li, Z., Xia, Y., Zhu, J., Ma, N., Chai, H., Liu, Z., 2020. Urban road traffic condition forecasting based on sparse ride-hailing service data. *IET Intelligent Transport Systems* 14, 668–674.
- Jiang, T.X., Huang, T.Z., Zhao, X.L., Deng, L.J., 2020a. Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm. *Journal of Computational and Applied Mathematics* 372, 112680.
- Jiang, T.X., Ng, M.K., Zhao, X.L., Huang, T.Z., 2020b. Framelet representation of tensor nuclear norm for third-order tensor completion. *IEEE Transactions on Image Processing* 29, 7233–7244.
- Jin, J., Liu, P., Huang, H., Dong, Y., 2024. Analyzing urban traffic crash patterns through spatio-temporal data: A city-level study using a sparse non-negative matrix factorization model with spatial constraints approach. *Applied Geography* 172, 103402.
- Jin, Z., Luo, Y., Zhao, X., Meng, D., 2025. Compressive imaging reconstruction via tensor decomposed multi-resolution grid encoding. *arXiv preprint arXiv:2507.07707*.
- Kang, U., Papalexakis, E., Harpale, A., Faloutsos, C., 2012. Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries, in: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 316–324.
- Kargas, N., Qian, C., Sidiropoulos, N.D., Xiao, C., Glass, L.M., Sun, J., 2021. Stelar: Spatio-temporal tensor factorization with latent epidemiological regularization, in: *Proceedings of the AAAI conference on artificial intelligence*, pp. 4830–4837.

- Kashinath, S.A., Mostafa, S.A., Mustapha, A., Mahdin, H., Lim, D., Mahmoud, M.A., Mohammed, M.A., Al-Rimy, B.A.S., Fudzee, M.F.M., Yang, T.J., 2021. Review of data fusion methods for real-time and multi-sensor traffic flow analysis. *IEEE Access* 9, 51258–51276.
- Kidando, E., Balyagati, P., Ngereza, A., Kutela, B., Kalambay, P., Kitali, A.E., 2024. Integrated multi regime and gaussian processes model for calibrating traffic fundamental diagram, in: 2024 IEEE 27th International Conference on Intelligent Transportation Systems, IEEE. pp. 4282–4287.
- Kiers, H.A., 2000. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics: A Journal of the Chemometrics Society* 14, 105–122.
- Kim, J., Park, K.H., Jang, J.G., Kang, U., 2024. Fast and accurate domain adaptation for irregular tensor decomposition, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1383–1394.
- Kolda, T.G., Bader, B.W., 2009. Tensor decompositions and applications. *SIAM Review* 51, 455–500.
- Kolda, T.G., Hong, D., 2020. Stochastic gradients for large-scale tensor decomposition. *SIAM Journal on Mathematics of Data Science* 2, 1066–1095.
- Kolda, T.G., Sun, J., 2008. Scalable tensor decompositions for multi-aspect data mining, in: 2008 Eighth IEEE international conference on data mining, IEEE. pp. 363–372.
- Kong, H., Xie, X., Lin, Z., 2018. t-schatten- p norm for low-rank tensor recovery. *IEEE Journal of Selected Topics in Signal Processing* 12, 1405–1419.
- Kong, J., Fan, X., Jin, X., Lin, S., Zuo, M., 2023. A variational bayesian inference-based en-decoder framework for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems* 25, 2966–2975.
- Kossaifi, J., Lipton, Z.C., Kolbeinsson, A., Khanna, A., Furlanello, T., Anandkumar, A., 2020. Tensor regression networks. *Journal of Machine Learning Research* 21, 1–21.
- Kruskal, J.B., 1989. Rank, decomposition, and uniqueness for 3-way and n-way arrays, in: Multiway data analysis, pp. 7–18.
- Lana, I., Del Ser, J., Velez, M., Vlahogianni, E.I., 2018. Road traffic forecasting: Recent advances and new challenges. *IEEE Intelligent Transportation Systems Magazine* 10, 93–109.
- Lanthier, J., Lei, M., Sun, L., Labbe, A., 2023. Bktr: An efficient spatiotemporally varying coefficient regression package in r and python .
- Larsen, B.W., Kolda, T.G., Zhang, A.R., Williams, A.H., 2024. Tensor decomposition meets rkhs: Efficient algorithms for smooth and misaligned data. *arXiv preprint arXiv:2408.05677* .
- Lei, M., 2024. Bayesian Kernelized Low-Rank Modeling for Multidimensional Spatiotemporal Data and Stochastic Optimization. McGill University (Canada).
- Lei, M., Labbe, A., Sun, L., 2022a. Bayesian complementary kernelized learning for multidimensional spatiotemporal data. *arXiv preprint arXiv:2208.09978* .
- Lei, M., Labbe, A., Sun, L., 2024. Scalable spatiotemporally varying coefficient modeling with bayesian kernelized tensor regression. *Bayesian Analysis* 1, 1–29.
- Lei, M., Labbe, A., Wu, Y., Sun, L., 2022b. Bayesian kernelized matrix factorization for spatiotemporal traffic data imputation and kriging. *IEEE Transactions on Intelligent Transportation Systems* 23, 18962–18974.
- Lei, M., Sun, L., 2023. Bayesian kernelized tensor factorization as surrogate for Bayesian optimization. *arXiv preprint arXiv:2302.14510* .
- Lei, M., Sun, L., 2024. Generalized least squares kernelized tensor factorization. *arXiv preprint arXiv:2412.07041* .
- Li, B.Z., Zhao, X.L., Chen, X., Ding, M., Liu, R.W., 2024a. Convolutional low-rank tensor representation for structural missing traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems* 25, 18847–18860.

- Li, B.Z., Zhao, X.L., Zhang, X., Ji, T.Y., Chen, X., Ng, M.K., 2023a. A learnable group-tube transform induced tensor nuclear norm and its application for tensor completion. *SIAM Journal on Imaging Sciences* 16, 1370–1397.
- Li, C., Chen, Y., Li, D.H., 2024b. Internet traffic tensor completion with tensor nuclear norm. *Computational Optimization and Applications* 87, 1033–1057.
- Li, D., Chu, D., Guan, X., He, W., Shen, H., 2024c. Adaptive regularized low-rank tensor decomposition for hyperspectral image denoising and destriping. *IEEE Transactions on Geoscience and Remote Sensing* 62, 1–17.
- Li, J., Xu, L., Li, R., Wu, P., Huang, Z., 2022. Deep spatial-temporal bi-directional residual optimisation based on tensor decomposition for traffic data imputation on urban road network. *Applied Intelligence* 52, 11363–11381.
- Li, L., Li, W., Qu, Y., Zhao, C., Tao, R., Du, Q., 2020a. Prior-based tensor approximation for anomaly detection in hyperspectral imagery. *IEEE Transactions on Neural Networks and Learning Systems* 33, 1037–1050.
- Li, L., Lin, X., Ran, B., Du, B., 2024d. Tensor decomposition of transportation temporal and spatial big data: A brief review. *Fundamental Research* .
- Li, L., Su, X., Zhang, Y., Lin, Y., Li, Z., 2015. Trend modeling for traffic time series analysis: An integrated study. *IEEE Transactions on Intelligent Transportation Systems* 16, 3430–3439.
- Li, Q., Luo, C., Yang, X., Wang, Y., Tan, H., 2025a. A tensor factorization based data imputation method for electric vehicle monitor platform. Available at SSRN 5360096 .
- Li, Q., Tan, H., Jiang, Z., Wu, Y., Ye, L., 2021. Nonrecurrent traffic congestion detection with a coupled scalable bayesian robust tensor factorization model. *Neurocomputing* 430, 138–149.
- Li, Q., Yang, X., Wang, Y., Wu, Y., He, D., 2023b. Spatial-temporal traffic modeling with a fusion graph reconstructed by tensor decomposition. *IEEE Transactions on Intelligent Transportation Systems* 25, 1749–1760.
- Li, S., Cheng, L., Zhang, T., Zhao, H., Li, J., 2023c. Graph-guided bayesian matrix completion for ocean sound speed field reconstruction. *The Journal of the Acoustical Society of America* 153, 689–710.
- Li, X., Xu, D., Zhou, H., Li, L., 2018a. Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences* 10, 520–545.
- Li, X., Ye, Y., Xu, X., 2017. Low-rank tensor completion with total variation for visual data inpainting, in: *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Li, Y., Liu, K.R., Razavilar, J., 1997. A parameter estimation scheme for damped sinusoidal signals based on low-rank hankel approximation. *IEEE Transactions on Signal Processing* 45, 481–486.
- Li, Y., Yu, R., Shahabi, C., Liu, Y., 2018b. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: *International Conference on Learning Representations*.
- Li, Y., Zhang, X., Luo, Y., Meng, D., 2025b. Deep rank-one tensor functional factorization for multi-dimensional data recovery, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 18539–18547.
- Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., Anandkumar, A., 2020b. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895* .
- Li, Z., Xia, L., Shi, L., Xu, Y., Yin, D., Huang, C., 2024e. Opencity: Open spatio-temporal foundation models for traffic prediction. *arXiv preprint arXiv:2408.10269* .
- Liang, W., Lv, F., Chen, L., Tao, H., Shi, M., Zhu, X., Cao, J., 2025. Citywide multi-step crime prediction via context-aware bayesian tensor decomposition. *IEEE Transactions on Information Forensics and Security* 20, 8030–8042.

- Liao, X., Wu, H., Luo, X., 2025. A novel tensor causal convolution network model for highly-accurate representation to spatio-temporal data. *IEEE Transactions on Automation Science and Engineering* 22, 19525–19537.
- Lim, L.H., Comon, P., 2009. Nonnegative approximations of nonnegative tensors. *Journal of Chemometrics: A Journal of the Chemometrics Society* 23, 432–441.
- Lin, M., Liu, J., Chen, H., Xu, X., Luo, X., Xu, Z., 2024a. A 3D convolution-incorporated dimension preserved decomposition model for traffic data prediction. *IEEE Transactions on Intelligent Transportation Systems* 26, 673–690.
- Lin, W., Li, C., Xu, L., Xie, K., 2024b. Enhanced network traffic anomaly detection: Integration of tensor eigenvector centrality with low-rank recovery models. *IEEE Transactions on Services Computing* 17, 3597–3612.
- Liu, H., Li, Y., Tsang, M., Liu, Y., 2019a. Costco: A neural tensor completion model for sparse tensors, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 324–334.
- Liu, J., Musalski, P., Wonka, P., Ye, J., 2012. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 208–220.
- Liu, M., Lyu, H., Ge, H., Cheng, R., 2025a. A bayesian tensor ring decomposition model with automatic rank determination for spatiotemporal traffic data imputation. *Applied Mathematical Modelling* 137, 115654.
- Liu, P., Long, H., Zheng, Z., Huang, N., Xiao, L., 2025b. Gradient subspace-regularized hyperspectral image and stripe-coupled non-convex tensor low-rank priors for destriping and denoising. *IEEE Transactions on Geoscience and Remote Sensing* 63.
- Liu, P., Long, H., Zheng, Z., Xiao, L., 2025c. Hyperspectral image denoising and destriping via gradient tensor subspace low-rank learning and along-across stripe directional constraints. *IEEE Transactions on Geoscience and Remote Sensing* 63.
- Liu, S., Zhao, X.L., Zhang, H., 2025d. Block tensor ring decomposition: Theory and application. *IEEE Transactions on Signal Processing* 73, 3029–3043.
- Liu, X., Hou, J., Peng, J., Wang, H., Meng, D., Wang, J., 2023. Tensor compressive sensing fused low-rankness and local-smoothness, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 8879–8887.
- Liu, Y., Long, Z., Huang, H., Zhu, C., 2019b. Low cp rank and tucker rank tensor completion for estimating missing components in image data. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 944–954.
- Liu, Z., Han, Z., Tang, Y., Zhao, X.L., Wang, Y., 2024. Low-tubal-rank tensor recovery via factorized gradient descent. *IEEE Transactions on Signal Processing* 72, 5470–5483.
- Lock, E.F., 2018. Tensor-on-tensor regression. *Journal of Computational and Graphical Statistics* 27, 638–647.
- Lone, M.A., Akhtar, N., 2024. Spatio-temporal data analytics for intelligent transportation systems: An overview, in: 2024 First International Conference on Pioneering Developments in Computer Science & Digital Technologies, IEEE. pp. 228–233.
- Long, Z., Liu, Y., Chen, L., Zhu, C., 2019. Low rank tensor completion for multiway visual data. *Signal processing* 155, 301–316.
- Long, Z., Zhu, C., Liu, J., Liu, Y., 2021. Bayesian low rank tensor ring for image recovery. *IEEE Transactions on Image Processing* 30, 3568–3580.
- Lu, C., Feng, J., Chen, Y., Liu, W., Lin, Z., Yan, S., 2019. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 925–938.
- Lu, Y., Yousaf, M., Meng, X., Chen, E., 2025. Mnt-tnn: Spatiotemporal traffic data imputation via compact multimode nonlinear transform-based tensor nuclear norm. *arXiv preprint arXiv:2503.22955*.

- Luo, X., Wu, H., Li, Z., 2022a. Neulf: A novel approach to nonlinear canonical polyadic decomposition on high-dimensional incomplete tensors. *IEEE Transactions on Knowledge and Data Engineering* 35, 6148–6166.
- Luo, X., Wu, H., Yuan, H., Zhou, M., 2019. Temporal pattern-aware qos prediction via biased non-negative latent factorization of tensors. *IEEE Transactions on Cybernetics* 50, 1798–1809.
- Luo, Y., Qiu, Y., Yang, P., Rao, H., Huang, Z., Zhou, G., 2025a. Latent low-rank tensor wheel decomposition for visual data completion. *Neurocomputing* , 130889.
- Luo, Y., Zhao, X., Li, Z., Ng, M.K., Meng, D., 2023. Low-rank tensor function representation for multi-dimensional data recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 3351–3369.
- Luo, Y., Zhao, X., Meng, D., 2024. Revisiting nonlocal self-similarity from continuous representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47, 450–468.
- Luo, Y., Zhao, X., Meng, D., 2025b. Continuous representation methods, theories, and applications: An overview and perspectives. arXiv preprint arXiv:2505.15222 .
- Luo, Y., Zhao, X.L., Meng, D., Jiang, T.X., 2022b. Hlrf: Hierarchical low-rank tensor factorization for inverse problems in multi-dimensional imaging, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19303–19312.
- Luttinen, J., 2013. Fast variational bayesian linear state-space model, in: Joint european conference on machine learning and knowledge discovery in databases, Springer. pp. 305–320.
- Luttinen, J., Ilin, A., 2009. Variational gaussian-process factor analysis for modeling spatio-temporal data. *Advances in Neural Information Processing Systems* 22, 1177–1185.
- Lyu, C., Lu, Q.L., Wu, X., Antoniou, C., 2024. Tucker factorization-based tensor completion for robust traffic data imputation. *Transportation research part C: emerging technologies* 160, 104502.
- Ma, J., Zhang, Q., Ho, J.C., Xiong, L., 2021. Spatio-temporal tensor sketching via adaptive sampling, in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I, Springer. pp. 490–506.
- Ma, Z., Teschendorff, A.E., Leijon, A., Qiao, Y., Zhang, H., Guo, J., 2014. Variational bayesian matrix factorization for bounded support data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 876–889.
- Mardani, M., Mateos, G., Giannakis, G.B., 2012. Dynamic anomalography: Tracking network anomalies via sparsity and low rank. *IEEE Journal of Selected Topics in Signal Processing* 7, 50–66.
- Mardani, M., Mateos, G., Giannakis, G.B., 2013. Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies. *IEEE Transactions on Information Theory* 59, 5186–5205.
- Ming, Z., Qin, Z., Zhang, L., Xu, Y., Qi, L., 2024. Network traffic recovery from link-load measurements using tensor triple decomposition strategy for third-order traffic tensors. *Journal of Computational and Applied Mathematics* 447, 115901.
- Minster, R., Li, Z., Ballard, G., 2024. Parallel randomized tucker decomposition algorithms. *SIAM Journal on Scientific Computing* 46, A1186–A1213.
- Minster, R., Saibaba, A.K., Kilmer, M.E., 2020. Randomized algorithms for low-rank tensor decompositions in the tucker format. *SIAM journal on mathematics of data science* 2, 189–215.
- Mukai, M., Hontani, H., Yokota, T., 2025. Plug-and-play low-rank tensor completion and reconstruction for improving applicability of tensor decompositions. *Frontiers in Applied Mathematics and Statistics* 11, 1594873.
- Ng, M.K.P., Yuan, Q., Yan, L., Sun, J., 2017. An adaptive weighted tensor completion method for the recovery of remote sensing images with missing data. *IEEE Transactions on Geoscience and Remote Sensing* 55, 3367–3381.

- Nguyen, D.T., Slavakis, K., Kofidis, E., Pados, D., 2025. Kernel regression of multi-way data via tensor trains with hadamard overparametrization: The dynamic graph flow case. arXiv preprint arXiv:2509.22197 .
- Nie, F., Huang, H., Ding, C., 2012. Low-rank matrix recovery via efficient schatten p-norm minimization, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 655–661.
- Nie, S., An, S., Peng, Q., Su, Y., 2025. Collar: Combating low-rank temporal latent representation for high-dimensional multivariate time series prediction using dynamic koopman regularization .
- 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.
- Nie, T., Qin, G., Wang, Y., Sun, J., 2023. Correlating sparse sensing for large-scale traffic speed estimation: A laplacian-enhanced low-rank tensor kriging approach. *Transportation research part C: emerging technologies* 152, 104190.
- Oliver, M., Webster, R., 2014. A tutorial guide to geostatistics: Computing and modelling variograms and kriging. *Catena* 113, 56–69.
- Oliver, M.A., Webster, R., et al., 2015. Basic steps in geostatistics: the variogram and kriging. volume 106. Springer.
- Paliwal, C., Bhatt, U., Biyani, P., Rajawat, K., 2021. Traffic estimation and prediction via online variational bayesian subspace filtering. *IEEE Transactions on Intelligent Transportation Systems* 23, 4674–4684.
- Pan, C., Ling, C., He, H., Qi, L., Xu, Y., 2024. A low-rank and sparse enhanced tucker decomposition approach for tensor completion. *Applied Mathematics and Computation* 465, 128432.
- Panagakis, Y., Kossaifi, J., Chrysos, G.G., Oldfield, J., Nicolaou, M.A., Anandkumar, A., Zafeiriou, S., 2021. Tensor methods in computer vision and deep learning. *Proceedings of the IEEE* 109, 863–890.
- Papalexakis, E.E., Faloutsos, C., Sidiropoulos, N.D., 2016. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Transactions on Intelligent Systems and Technology* 8, 1–44.
- Peng, C., Xu, C., Kudryavtsev, F., Ai, Q., Gao, Y., Jiao, Y., 2025a. Spatiotemporal factorized graph neural networks for joint large-scale traffic prediction and online pattern recognition. *IEEE Transactions on Intelligent Transportation Systems* 26, 14896–14909.
- Peng, J., Luo, Y., Cao, X., Xu, S., Meng, D., 2025b. Beyond low-rankness: Guaranteed matrix recovery via modified nuclear norm. arXiv preprint arXiv:2507.18327 .
- Peng, J., Wang, H., Cao, X., Xu, S., . Fast guaranteed tensor recovery with adaptive tensor nuclear norm .
- Phan, A.H., Tichavský, P., Cichocki, A., 2013. Fast alternating ls algorithms for high order cande-comp/parafac tensor factorizations. *IEEE Transactions on Signal Processing* 61, 4834–4846.
- Phipps, E.T., Johnson, N.T., Kolda, T.G., 2023. Streaming generalized canonical polyadic tensor decompositions, in: Proceedings of the Platform for Advanced Scientific Computing Conference, pp. 1–10.
- Qi, H., Zhao, X., Yao, Y., Yang, H., Chai, S., Chen, X., 2023. Bgcp-based traffic data imputation and accident detection applications for the national trunk highway. *Accident Analysis & Prevention* 186, 107051.
- Qi, L., Chen, Y., Bakshi, M., Zhang, X., 2021. Triple decomposition and tensor recovery of third order tensors. *SIAM Journal on Matrix Analysis and Applications* 42, 299–329.
- Qiu, D., Bai, M., Ng, M.K., Zhang, X., 2021. Robust low-rank tensor completion via transformed tensor nuclear norm with total variation regularization. *Neurocomputing* 435, 197–215.
- Qiu, Y., Zhou, G., Wang, A., Zhao, Q., Xie, S., 2024. Balanced unfolding induced tensor nuclear norms for high-order tensor completion. *IEEE Transactions on Neural Networks and Learning Systems* 36, 4724–4737.
- Qu, L., Li, L., Zhang, Y., Hu, J., 2009. Ppca-based missing data imputation for traffic flow volume: A systematical approach. *IEEE Transactions on intelligent transportation systems* 10, 512–522.

Qu, L., Zhang, Y., Hu, J., Jia, L., Li, L., 2008. A bPCA based missing value imputing method for traffic flow volume data, in: 2008 IEEE Intelligent Vehicles Symposium, IEEE. pp. 985–990.

Rajih, M., Comon, P., Harshman, R.A., 2008. Enhanced line search: A novel method to accelerate parafac. SIAM journal on matrix analysis and applications 30, 1128–1147.

Ran, B., Song, L., Zhang, J., Cheng, Y., Tan, H., 2016a. Using tensor completion method to achieving better coverage of traffic state estimation from sparse floating car data. PloS one 11, e0157420.

Ran, B., Tan, H., Wu, Y., Jin, P.J., 2016b. Tensor based missing traffic data completion with spatial-temporal correlation. Physica A: Statistical Mechanics and its Applications 446, 54–63.

Rao, N., Yu, H.F., Ravikumar, P., Dhillon, I.S., 2015. Collaborative filtering with graph information: Consistency and scalable methods. Advances in Neural Information Processing Systems , 2107–2115.

Rasmussen, C.E., Williams, C.K., 2006. Gaussian processes for machine learning. MIT press Cambridge, MA.

Recht, B., Fazel, M., Parrilo, P.A., 2010. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM review 52, 471–501.

Ren, P., Chen, X., Sun, L., Sun, H., 2021. Incremental bayesian matrix/tensor learning for structural monitoring data imputation and response forecasting. Mechanical Systems and Signal Processing 158, 107734.

Ren, Q., Banerjee, S., 2013. Hierarchical factor models for large spatially misaligned data: A low-rank predictive process approach. Biometrics 69, 19–30.

Saberi, M., Hamedmoghadam, H., Ashfaq, M., Hosseini, S.A., Gu, Z., Shafiei, S., Nair, D.J., Dixit, V., Gardner, L., Waller, S.T., et al., 2020. A simple contagion process describes spreading of traffic jams in urban networks. Nature communications 11, 1616.

Said, A.B., Erradi, A., 2021. Spatiotemporal tensor completion for improved urban traffic imputation. IEEE Transactions on Intelligent Transportation Systems 23, 6836–6849.

Salakhutdinov, R., Mnih, A., 2008. Bayesian probabilistic matrix factorization using markov chain monte carlo, in: Proceedings of the 25th international conference on Machine learning, pp. 880–887.

Sarkar, T., Rakhlin, A., Dahleh, M.A., 2021. Finite time lti system identification. Journal of Machine Learning Research 22, 1–61.

Schmidt, A.M., Lopes, H.F., 2019. Dynamic models, in: Handbook of environmental and ecological statistics. Chapman and Hall/CRC, pp. 57–80.

Schmidt, M.N., Winther, O., Hansen, L.K., 2009. Bayesian non-negative matrix factorization, in: International Conference on Independent Component Analysis and Signal Separation, Springer. pp. 540–547.

Semerci, O., Hao, N., Kilmer, M.E., Miller, E.L., 2014. Tensor-based formulation and nuclear norm regularization for multienergy computed tomography. IEEE Transactions on Image Processing 23, 1678–1693.

Shang, F., Cheng, J., Liu, Y., Luo, Z.Q., Lin, Z., 2017. Bilinear factor matrix norm minimization for robust pca: Algorithms and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 40, 2066–2080.

Sharma, S., Nayak, R., Bhaskar, A., 2025. Harmonizing recurring patterns and non-recurring trends in traffic datasets for enhanced estimation of missing information. Transportation Research Part C: Emerging Technologies 174, 105083.

Shekhar, S., Jiang, Z., Ali, R.Y., Eftelioglu, E., Tang, X., Gunturi, V.M., Zhou, X., 2015. Spatiotemporal data mining: A computational perspective. ISPRS International Journal of Geo-Information 4, 2306–2338.

Shi, J., Leau, Y.B., Li, K., Park, Y.J., Yan, Z., 2020a. Optimization and decomposition methods in network traffic prediction model: A review and discussion. IEEE Access 8, 202858–202871.

- Shi, Q., Yin, J., Cai, J., Cichocki, A., Yokota, T., Chen, L., Yuan, M., Zeng, J., 2020b. Block hankel tensor arima for multiple short time series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 5758–5766.
- Shi, Y., Shen, W., 2023. Bayesian methods in tensor analysis. arXiv preprint arXiv:2302.05978 .
- Shu, H., Li, J., Lei, T., Sun, L., 2025. Robust tensor completion via gradient tensor nuclear l1-l2 norm for traffic data recovery. arXiv preprint arXiv:2506.22732 .
- Shu, H., Wang, H., Peng, J., Meng, D., 2024. Low-rank tensor completion with 3-d spatiotemporal transform for traffic data imputation. IEEE Transactions on Intelligent Transportation Systems 25, 18673–18687.
- Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P., 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE signal processing magazine 30, 83–98.
- Smith, S., Ravindran, N., Sidiropoulos, N.D., Karypis, G., 2015. Splatt: Efficient and parallel sparse tensor-matrix multiplication, in: 2015 IEEE International Parallel and Distributed Processing Symposium, IEEE. pp. 61–70.
- Smola, A.J., Kondor, R., 2003. Kernels and regularization on graphs, in: Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings, Springer. pp. 144–158.
- Sofuo glu, S.E., Aviyente, S., 2022. Gloss: Tensor-based anomaly detection in spatiotemporal urban traffic data. Signal Processing 192, 108370.
- Song, Q., Ge, H., Caverlee, J., Hu, X., 2019. Tensor completion algorithms in big data analytics. ACM Transactions on Knowledge Discovery from Data 13, 1–48.
- Sørensen, M., Lathauwer, L.D., Comon, P., Icart, S., Deneire, L., 2012. Canonical polyadic decomposition with a columnwise orthonormal factor matrix. SIAM Journal on Matrix Analysis and Applications 33, 1190–1213.
- Sun, J., Tao, D., Papadimitriou, S., Yu, P.S., Faloutsos, C., 2008. Incremental tensor analysis: Theory and applications. ACM Transactions on Knowledge Discovery from Data 2, 1–37.
- Sure, P., Srinivasan, C.P., Babu, C.N., 2021. Spatio-temporal constraint-based low rank matrix completion approaches for road traffic networks. IEEE Transactions on Intelligent Transportation Systems 23, 13452–13462.
- Takemoto, S., Naganuma, K., Ono, S., 2025. Spatio-spectral structure tensor total variation for hyperspectral image denoising and destriping. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 18, 19157–19175.
- 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.
- Tan, H., Wu, Y., Shen, B., Jin, P.J., Ran, B., 2016. Short-term traffic prediction based on dynamic tensor completion. IEEE Transactions on Intelligent Transportation Systems 17, 2123–2133.
- Tokcan, N., Sofi, S.S., Prévost, C., Kharbech, S., Magnier, B., Nguyen, T.P., Zniyed, Y., De Lathauwer, L., et al., 2025. Tensor decompositions for signal processing: Theory, advances, and applications. Signal Processing , 110191.
- Tong, X., Cheng, L., Wu, Y.C., 2023. Bayesian tensor tucker completion with a flexible core. IEEE Transactions on Signal Processing 71, 4077–4091.
- Tu, L., Wu, J., Wang, C., Meng, D., Jin, Z., 2025. Fourier-based decoupling network for joint low-light image enhancement and deblurring. IEEE Transactions on Image Processing .
- Tucker, L.R., 1963. Implications of factor analysis of three-way matrices for measurement of change. Problems in measuring change 15, 3.

- Tucker, L.R., 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* 31, 279–311.
- Tzikas, D.G., Likas, A.C., Galatsanos, N.P., 2008. The variational approximation for bayesian inference. *IEEE Signal Processing Magazine* 25, 131–146.
- Veganzones, M.A., Cohen, J.E., Farias, R.C., Chanussot, J., Comon, P., 2015. Nonnegative tensor cp decomposition of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing* 54, 2577–2588.
- Wang, A., Qiu, Y., Bai, M., Jin, Z., Zhou, G., Zhao, Q., 2024a. Generalized tensor decomposition for understanding multi-output regression under combinatorial shifts. *Advances in Neural Information Processing Systems* 37, 47559–47635.
- Wang, H., Peng, J., Qin, W., Wang, J., Meng, D., 2023a. Guaranteed tensor recovery fused low-rankness and smoothness. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 10990–11007.
- Wang, J., Zou, L., Zhao, J., Wang, X., 2024b. Dynamic capacity drop propagation in incident-affected networks: Traffic state modeling with sis-ctm. *Physica A: Statistical Mechanics and its Applications* 637, 129536.
- Wang, X., Miranda-Moreno, L., Sun, L., 2021. Hankel-structured tensor robust pca for multivariate traffic time series anomaly detection. *arXiv preprint arXiv:2110.04352*.
- Wang, X., Sun, L., 2021. Diagnosing spatiotemporal traffic anomalies with low-rank tensor autoregression. *IEEE Transactions on Intelligent Transportation Systems* 22, 7904–7913.
- Wang, X., Wu, Y., Zhuang, D., Sun, L., 2023b. Low-rank hankel tensor completion for traffic speed estimation. *IEEE Transactions on Intelligent Transportation Systems* 24, 4862–4871.
- Wang, Y., Zhang, Y., Piao, X., Liu, H., Zhang, K., 2018. Traffic data reconstruction via adaptive spatial-temporal correlations. *IEEE Transactions on Intelligent Transportation Systems* 20, 1531–1543.
- Wang, Y.X., Sharpnack, J., Smola, A.J., Tibshirani, R.J., 2016. Trend filtering on graphs. *Journal of Machine Learning Research* 17, 1–41.
- Wang, Z., Liu, Z., Cheng, Q., Gu, Z., 2024c. Integrated self-consistent macro-micro traffic flow modeling and calibration framework based on trajectory data. *Transportation research part C: emerging technologies* 158, 104439.
- Wei, Z., He, D., Jin, Z., Liu, B., Shan, S., Chen, Y., Miao, J., 2023. Density-based affinity propagation tensor clustering for intelligent fault diagnosis of train bogie bearing. *IEEE Transactions on Intelligent Transportation Systems* 24, 6053–6064.
- Wimalawarne, K., Mamitsuka, H., 2021. Reshaped tensor nuclear norms for higher order tensor completion. *Machine Learning* 110, 507–531.
- Wu, F., Cheng, Z., Chen, H., Qiu, Z., Sun, L., 2024a. Traffic state estimation from vehicle trajectories with anisotropic gaussian processes. *Transportation Research Part C: Emerging Technologies* 163, 104646.
- Wu, F., Liu, S., Wang, H., Tao, B., Luo, J., Peng, Z., 2026. Neural spatial–temporal tensor representation for infrared small target detection. *Pattern Recognition* 169, 111929.
- Wu, G., Li, H., Tang, Y., Huang, W., Peng, J., 2024b. A novel shrinkage operator for tensor completion with low-tubal-rank approximation. *Digital Signal Processing* 152, 104597.
- Wu, H., Mi, J., 2025. A cauchy loss-incorporated nonnegative latent factorization of tensors model for spatiotemporal traffic data recovery. *Neurocomputing* 626, 129575.
- Wu, J., Zhao, Y., Zhang, H., Hu, G., Zeng, H., Li, S., 2022a. Spatio-temporal traffic data tensor restoration method based on direction weighting and p-shrinkage norm. *Mathematical Problems in Engineering* 2022, 3304677.
- Wu, P., Pei, M., Wang, T., Liu, Y., Liu, Z., Zhong, L., 2024c. A low-rank bayesian temporal matrix factorization for the transfer time prediction between metro and bus systems. *IEEE Transactions on Intelligent Transportation Systems* 25, 7206–7222.

- Wu, P.L., Ding, M., Zheng, Y.B., 2023. Spatiotemporal traffic data imputation by synergizing low tensor ring rank and nonlocal subspace regularization. *IET Intelligent Transport Systems* 17, 1908–1923.
- Wu, T., Fan, J., 2024. Smooth tensor product for tensor completion. *IEEE Transactions on Image Processing* 33, 6483–6496.
- Wu, T., Gao, B., Fan, J., Xue, J., Woo, W.L., 2022b. Low-rank tensor completion based on self-adaptive learnable transforms. *IEEE Transactions on Neural Networks and Learning Systems* 35, 8826–8838.
- Wu, X., Xu, M., Fang, J., Wu, X., 2022c. A multi-attention tensor completion network for spatiotemporal traffic data imputation. *IEEE Internet of Things Journal* 9, 20203–20213.
- Wu, Y., Tan, H., Li, Y., Zhang, J., Chen, X., 2018. A fused cp factorization method for incomplete tensors. *IEEE Transactions on Neural Networks and Learning Systems* 30, 751–764.
- Wu, Y., Zhuang, D., Lei, M., Labbe, A., Sun, L., 2021. Spatial aggregation and temporal convolution networks for real-time kriging. *arXiv preprint arXiv:2109.12144*.
- Wu, Z., Liu, X., Zhang, X., 2025. Multi dynamic temporal representation graph convolutional network for traffic flow prediction. *Scientific Reports* 15, 16734.
- Xia, C., Yin, X., Yu, J., Liang, X., Chen, L., 2025. Missing traffic data imputation based on tensor completion and graph network fusion. *Transportation Research Record* , 03611981251330889.
- Xie, H., Gong, Y., Dong, X., 2024. Spatial-temporal regularized tensor decomposition method for traffic speed data imputation. *International Journal of Data Science and Analytics* 17, 203–223.
- Xie, K., Peng, C., Wang, X., Xie, G., Wen, J., Cao, J., Zhang, D., Qin, Z., 2018a. Accurate recovery of internet traffic data under variable rate measurements. *IEEE/ACM Transactions on Networking* 26, 1137–1150.
- Xie, K., Wang, L., Wang, X., Xie, G., Wen, J., Zhang, G., Cao, J., Zhang, D., 2018b. Accurate recovery of internet traffic data: A sequential tensor completion approach. *IEEE/ACM Transactions on Networking* 26, 793–806.
- Xie, Q., Fu, J., Xu, Z., Meng, D., 2025. Rotation equivariant arbitrary-scale image super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47, 11382–11399.
- Xie, Y., Gu, S., Liu, Y., Zuo, W., Zhang, W., Zhang, L., 2016. Weighted schatten p -norm minimization for image denoising and background subtraction. *IEEE Transactions on Image Processing* 25, 4842–4857.
- Xing, J., Liu, R., Anish, K., Liu, Z., 2023a. A customized data fusion tensor approach for interval-wise missing network volume imputation. *IEEE Transactions on Intelligent Transportation Systems* 24, 12107–12122.
- Xing, J., Liu, R., Anish, K., Liu, Z., 2023b. A data fusion candecomp-parafac method for interval-wise missing network volume imputation. *IEEE Transactions on Intelligent Transportation Systems* 24, 12107–12122.
- Xing, J., Wu, W., Cheng, Q., Liu, R., 2022. Traffic state estimation of urban road networks by multi-source data fusion: Review and new insights. *Physica A: Statistical Mechanics and its Applications* 595, 127079.
- Xiong, L., Chen, X., Huang, T.K., Schneider, J., Carbonell, J.G., 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization, in: *Proceedings of the 2010 SIAM international conference on data mining, SIAM*. pp. 211–222.
- Xu, S., Peng, J., Ji, T.Y., Cao, X., Sun, K., Fei, R., Meng, D., 2024a. Stacked tucker decomposition with multi-nonlinear products for remote sensing imagery inpainting. *IEEE Transactions on Geoscience and Remote Sensing* 62,
- Xu, S., Zhao, Z., Cao, X., Peng, J., Zhao, X.L., Meng, D., Zhang, Y., Timofte, R., Van Gool, L., 2025a. Parameterized low-rank regularizer for high-dimensional visual data. *International Journal of Computer Vision* , 1–24.

- Xu, X., Lin, M., Luo, X., Xu, Z., 2023a. Hrst-lr: a hessian regularization spatio-temporal low rank algorithm for traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems* 24, 11001–11017.
- Xu, X., Lin, M., Xu, Z., Luo, X., 2025b. Attention-mechanism-based neural latent-factorization-of-tensors model. *ACM Transactions on Knowledge Discovery from Data* 19, 1–27.
- Xu, Y., Yin, W., 2013. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on imaging sciences* 6, 1758–1789.
- Xu, Z., Lv, Z., Chu, B., Li, J., 2023b. Fast autoregressive tensor decomposition for online real-time traffic flow prediction. *Knowledge-Based Systems* 282, 111125.
- Xu, Z., Lv, Z., Chu, B., Li, J., 2024b. A fast spatial-temporal information compression algorithm for online real-time forecasting of traffic flow with complex nonlinear patterns. *Chaos, Solitons & Fractals* 182, 114852.
- Xue, S., Qiu, W., Liu, F., Jin, X., 2018. Low-rank tensor completion by truncated nuclear norm regularization, in: 2018 24th International Conference on Pattern Recognition, IEEE. pp. 2600–2605.
- Yan, J., Li, H., Bai, Y., Lin, Y., 2021. Spatial—temporal traffic flow data restoration and prediction method based on the tensor decomposition. *Applied Sciences* 11, 9220.
- Yang, F., Hu, Q., Su, X., 2025a. Hyperspectral image denoising based on hyper-laplacian total variation in spectral gradient domain. *IEEE Transactions on Geoscience and Remote Sensing* 63.
- Yang, F., Liu, G., Huang, L., Chin, C.S., 2020a. Tensor decomposition for spatial—temporal traffic flow prediction with sparse data. *Sensors* 20, 6046.
- Yang, H., Lin, M., Chen, H., Luo, X., Xu, Z., 2024. Latent factor analysis model with temporal regularized constraint for road traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems* 26, 724–741.
- Yang, J.H., Chen, C., Dai, H.N., Ding, M., Wu, Z.B., Zheng, Z., 2022. Robust corrupted data recovery and clustering via generalized transformed tensor low-rank representation. *IEEE Transactions on Neural Networks and Learning Systems* 35, 8839–8853.
- Yang, J.H., Zhao, X.L., Ji, T.Y., Ma, T.H., Huang, T.Z., 2020b. Low-rank tensor train for tensor robust principal component analysis. *Applied Mathematics and Computation* 367, 124783.
- Yang, S., Wu, J., Xu, Y., Yang, T., 2019. Revealing heterogeneous spatiotemporal traffic flow patterns of urban road network via tensor decomposition-based clustering approach. *Physica A: Statistical Mechanics and its Applications* 526, 120688.
- Yang, Z., Yang, L.T., Li, C., Shan, L., Zhao, H., Nie, X., 2025b. Collaborative bayesian tensor factorization-based reliable traffic speed data prediction in t-cps. *IEEE Transactions on Intelligent Transportation Systems* 26, 14393–14406.
- Yang, Z., Yang, L.T., Wang, H., Zhao, H., Liu, D., 2025c. Bayesian nonnegative tensor completion with automatic rank determination. *IEEE Transactions on Image Processing* 34, 2036–2051.
- Yin, K., Afshar, A., Ho, J.C., Cheung, W.K., Zhang, C., Sun, J., 2020. Logpar: Logistic parafac2 factorization for temporal binary data with missing values, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1625–1635.
- Yokota, T., 2024. Very basics of tensors with graphical notations: Unfolding, calculations, and decompositions. *arXiv preprint arXiv:2411.16094*.
- Yokota, T., Erem, B., Guler, S., Warfield, S.K., Hontani, H., 2018. Missing slice recovery for tensors using a low-rank model in embedded space, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8251–8259.
- Yokota, T., Lee, N., Cichocki, A., 2016a. Robust multilinear tensor rank estimation using higher order singular value decomposition and information criteria. *IEEE Transactions on Signal Processing* 65, 1196–1206.

- Yokota, T., Zhao, Q., Cichocki, A., 2016b. Smooth parafac decomposition for tensor completion. *IEEE Transactions on Signal Processing* 64, 5423–5436.
- Yu, G., Wan, S., Ling, C., Qi, L., Xu, Y., 2024. Tensor train factorization with spatio-temporal smoothness for streaming low-rank tensor completion. *Frontiers of Mathematics* 19, 933–959.
- Yu, H.F., Rao, N., Dhillon, I.S., 2016. Temporal regularized matrix factorization for high-dimensional time series prediction. *Advances in Neural Information Processing Systems* 29.
- Yu, J., Stettler, M.E., Angeloudis, P., Hu, S., Chen, X.M., 2020. Urban network-wide traffic speed estimation with massive ride-sourcing gps traces. *Transportation Research Part C: Emerging Technologies* 112, 136–152.
- Yu, J., Zou, T., Zhou, G., 2023. Low tensor-ring rank completion: parallel matrix factorization with smoothness on latent space. *Neural Computing and Applications* 35, 7003–7016.
- Yu, L., Guan, C., Wang, H., He, Y., Cao, W., Leung, C.S., 2025. Robust tensor ring decomposition for urban traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems* 26, 8707–8719.
- Yu, R., Li, G., Liu, Y., 2018. Tensor regression meets gaussian processes, in: International conference on artificial intelligence and statistics, PMLR. pp. 482–490.
- Yu, Y., Li, H., 2024. A block-randomized stochastic method with importance sampling for cp tensor decomposition. *Advances in Computational Mathematics* 50, 22.
- Yuan, Y., Zhang, Z., Yang, X.T., Zhe, S., 2021. Macroscopic traffic flow modeling with physics regularized gaussian process: A new insight into machine learning applications in transportation. *Transportation Research Part B: Methodological* 146, 88–110.
- Zeng, J., Yang, L.T., Wang, C., Ruan, Y., Zhu, C., 2024a. Structured-anomaly pursuit of network traffic via corruption-robust low-rank tensor decomposition. *IEEE Transactions on Network Science and Engineering* 11, 2510–2523.
- Zeng, Z., Feng, J., Huang, Z., Liu, B., Zhou, B., 2025. A flexible approach based on hybrid global low-rankness and smoothness regularization with nonlocal structure for traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems* 26, 8864–8879.
- Zeng, Z., Liu, B., Feng, J., Yang, X., 2024b. Low-rank tensor and hybrid smoothness regularization-based approach for traffic data imputation with multimodal missing. *IEEE Transactions on Intelligent Transportation Systems* 25, 13014–13026.
- Zhang, A., 2019. Cross: Efficient low-rank tensor completion. *The Annals of Statistics* 47, 936–964.
- Zhang, D., Hu, Y., Ye, J., Li, X., He, X., 2012. Matrix completion by truncated nuclear norm regularization, in: 2012 IEEE Conference on computer vision and pattern recognition, IEEE. pp. 2192–2199.
- Zhang, J., Wei, J., 2024. Multi-residual tensor completion for spatiotemporal data imputation. *Internet of Things* 25, 101114.
- Zhang, W., 2024. Bayesian dynamic factor models for high-dimensional matrix-valued time series. arXiv preprint arXiv:2409.08354 .
- Zhang, W., Cribben, I., Guindani, M., et al., 2021. Bayesian time-varying tensor vector autoregressive models for dynamic effective connectivity. arXiv preprint arXiv:2106.14083 .
- Zhang, X., Zhang, Y., Wei, X., Hu, Y., Yin, B., 2022. Traffic forecasting with missing data via low rank dynamic mode decomposition of tensor. *IET Intelligent Transport Systems* 16, 1164–1176.
- Zhang, Z., Aeron, S., 2016. Exact tensor completion using t-svd. *IEEE Transactions on Signal Processing* 65, 1511–1526.
- Zhang, Z., Ling, C., He, H., Qi, L., 2024. A tensor train approach for internet traffic data completion. *Annals of Operations Research* 339, 1461–1479.

Zhao, B., Gao, X., Liu, J., Zhao, J., Xu, C., 2020. Spatiotemporal data fusion in graph convolutional networks for traffic prediction. *IEEE Access* 8, 76632–76641.

Zhao, H., Yang, L.T., Yang, Z., Liu, D., Nie, X., Ren, B., 2024a. Sparse bayesian tensor completion for data recovery in intelligent iot systems. *IEEE Internet of Things Journal* 11, 25682–25693.

Zhao, Q., Lin, Y., Wang, F., Meng, D., 2024b. Adaptive weighting function for weighted nuclear norm based matrix/tensor completion. *International Journal of Machine Learning and Cybernetics* 15, 697–718.

Zhao, Q., Zhang, L., Cichocki, A., 2015. Bayesian cp factorization of incomplete tensors with automatic rank determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 1751–1763.

Zhao, X.L., Yang, J.H., Ma, T.H., Jiang, T.X., Ng, M.K., Huang, T.Z., 2021. Tensor completion via complementary global, local, and nonlocal priors. *IEEE Transactions on Image Processing* 31, 984–999.

Zhao, Y., Lu, W., Rui, Y., Ran, B., 2023a. Classification of the traffic status subcategory with etc gantry data: an improved support tensor machine approach. *Journal of Advanced Transportation* 2023, 2765937.

Zhao, Y., Tuo, M., Zhang, H., Zhang, H., Wu, J., Gao, F., 2024c. Nonnegative low-rank tensor completion method for spatiotemporal traffic data. *Multimedia Tools and Applications* 83, 61761–61776.

Zhao, Y., Yang, L.T., Zhang, R., 2017. A tensor-based multiple clustering approach with its applications in automation systems. *IEEE Transactions on Industrial Informatics* 14, 283–291.

Zhao, Z., Tang, L., Fang, M., Yang, X., Li, C., Li, Q., 2023b. Toward urban traffic scenarios and more: a spatio-temporal analysis empowered low-rank tensor completion method for data imputation. *International Journal of Geographical Information Science* 37, 1936–1969.

Zheng, Y.B., Huang, T.Z., Ji, T.Y., Zhao, X.L., Jiang, T.X., Ma, T.H., 2019. Low-rank tensor completion via smooth matrix factorization. *Applied Mathematical Modelling* 70, 677–695.

Zheng, Y.B., Zhao, X.L., Li, H.C., Li, C., Huang, T.Z., Zhao, Q., 2025. Tensor network decomposition for data recovery: Recent advancements and future prospects. *Neural Networks*, 107808.

Zhou, H., Li, L., Zhu, H., 2013. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association* 108, 540–552.

Zhou, H., Zhang, D., Xie, K., Chen, Y., 2015. Spatio-temporal tensor completion for imputing missing internet traffic data, in: 2015 IEEE 34th International Performance Computing and Communications Conference, IEEE. pp. 1–7.

Zhou, J., Zhang, Z., 2025. Dynamic and static lagrange multiplier μ based bayesian cp factorization with orthogonal factors: Theory and applications. *Signal Processing* 236, 110045.

Zhou, P., Lu, C., Lin, Z., Zhang, C., 2017. Tensor factorization for low-rank tensor completion. *IEEE Transactions on Image Processing* 27, 1152–1163.

Zhou, S., Vinh, N.X., Bailey, J., Jia, Y., Davidson, I., 2016. Accelerating online cp decompositions for higher order tensors, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1375–1384.

Zhou, T., Shan, H., Banerjee, A., Sapiro, G., 2012. Kernelized probabilistic matrix factorization: Exploiting graphs and side information, in: Proceedings of the 2012 SIAM international Conference on Data mining, SIAM. pp. 403–414.

Zhou, W., Zheng, H., Feng, X., Lin, D., 2019. A multi-source based coupled tensors completion algorithm for incomplete traffic data imputation, in: 2019 11th International Conference on Wireless Communications and Signal Processing, IEEE. pp. 1–6.

Zhu, J., Ge, Z., Song, Z., Gao, F., 2018. Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data. *Annual Reviews in Control* 46, 107–133.

Zhu, Y., Wang, W., Yu, G., Wang, J., Tang, L., 2022. A bayesian robust cp decomposition approach for missing traffic data imputation. *Multimedia Tools and Applications* 81, 33171–33184.

Zifßner, P., Rettore, P.H., Santos, B.P., Loevenich, J.F., Lopes, R.R.F., 2023. Datafits: A heterogeneous data fusion framework for traffic and incident prediction. *IEEE Transactions on Intelligent Transportation Systems* 24, 11466–11478.

Zong, X., Yan, H., Qi, Y., 2025. Recent advances in multi-source data fusion for traffic flow prediction: A review. *Archives of Computational Methods in Engineering* , 1–23.