

Forecasting Sparse Movement Speed of Urban Road Networks with Nonstationary Temporal Matrix Factorization

Xinyu Chen

Polytechnique Montreal, Montreal, QC H3T 1J4, Canada, chenxy346@gmail.com

Chengyuan Zhang

McGill University, Montreal, QC H3A 0C3, Canada, enzozy@gmail.com

Xi-Le Zhao

University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China, xlzhao122003@163.com

Nicolas Saunier

Polytechnique Montreal, Montreal, QC H3T 1J4, Canada, nicolas.saunier@polymtl.ca

Lijun Sun

McGill University, Montreal, QC H3A 0C3, Canada, lijun.sun@mcgill.ca

Movement speed data from urban road networks, computed from ridesharing vehicles or taxi trajectories, is often high-dimensional, sparse, and nonstationary (e.g., exhibiting seasonality). These characteristics pose challenges for developing scalable and efficient data-driven solutions for traffic flow estimation and forecasting using machine learning techniques. To address these challenges, we propose a Nonstationary Temporal Matrix Factorization (NoTMF) model that leverages matrix factorization to project high-dimensional and sparse movement speed data into low-dimensional latent spaces. This results in a concise formula with the multiplication between spatial and temporal factor matrices. To characterize the temporal correlations, NoTMF takes a latent equation on the seasonal differenced temporal factors using higher-order vector autoregression (VAR). This approach not only preserves the low-rank structure of sparse movement speed data but also maintains consistent temporal dynamics, including seasonality information. The learning process for NoTMF involves optimizing the spatial and temporal factor matrices along with a collection of VAR coefficient matrices. To solve this efficiently, we introduce an alternating minimization framework, which tackles a challenging procedure of estimating the temporal factor matrix using conjugate gradient method, as the subproblem involves both partially observed matrix factorization and seasonal differenced VAR. To evaluate the forecasting performance of NoTMF, we conduct extensive experiments on Uber movement speed datasets, which are estimated from ridesharing vehicle trajectories. These datasets contain a large proportion of missing values due to insufficient ridesharing vehicles on the urban road network. Despite the presence of missing data, NoTMF demonstrates superior forecasting accuracy and effectiveness compared to baseline models. Moreover, as the seasonality of movement speed data is of great concern, the experiment results highlight the significance of addressing the nonstationarity of movement speed data.

Key words: Urban transportation network, ridesharing vehicles, floating car data, movement speed data, traffic flow forecasting, machine learning, matrix factorization, vector autoregression

History: February 10, 2025

1. Introduction

In the field of urban transportation networks, the rise of crowdsourced data, collected through fixed sensing detectors (e.g., loop detectors, radar detectors, and cameras) and mobile sensors (e.g., floating cars equipped with the Global Positioning System (GPS) devices), has revolutionized the monitoring of urban traffic states such as movement speeds (Treiber and Kesting 2013, Zheng 2015). Mobile sensor data, including human mobility trajectories, has gained great attention over the past decades due to the widespread utilization of mobile devices. These data provide unprecedented insights into analyzing human mobility behaviors (e.g., movement from place A to place B with a sequence of positions forming a trajectory) and enable the computation of movement speeds across urban road networks. For instance, trajectory data from ridesharing vehicles not only reveal the traffic states of urban road networks but also serve as pivotal components in city-wide traffic state estimation tasks. However, urban movement speed data derived from ridesharing vehicles are inherently sparse, capturing only a small fraction of the total vehicular trajectories and leading to the issues of missing values. Thus, these imperfect data pose unprecedented methodological and practical challenges for accurate movement speed estimation and forecasting.

Using machine learning algorithms to estimate urban traffic flow from ridesharing trajectories is technically feasible but complex. The main challenge lies in leveraging partial observed movement speeds to accurately infer unobserved speeds. This challenge is central to the data-driven task of traffic flow imputation, which is critical for effective movement speed estimation (Chen and Sun 2022). An analysis of the Uber movement speed data¹ highlights significance differences in data completeness between peak and off-peak hours, with higher missing rates of movement speeds during off-peak hours. Additionally, the data reveals daily seasonal patterns in missing values. These findings align with our understanding of urban traffic dynamics and emphasize the need to address data biases to improve the accuracy of urban traffic forecasting.

¹ Available till October 2023 at <https://movement.uber.com/>.

This study aims to advance the forecasting of sparse movement speeds within urban transportation networks by relying on several key assumptions: (i) Partially observed movement speed data contains informative low-dimensional spatial and temporal patterns; (ii) Movement speed data exhibits strong temporal dynamics and trends, which can be effectively modeled using linear time-invariant systems, such as vector autoregression (VAR); (iii) Addressing the nonstationarity of movement speed data through seasonal differencing, either directly in the data space or indirectly in latent spaces, is crucial for mitigating biases inherent in urban movement speed data. It is worth noting that matrix factorization is invariant to the permutation of rows and columns, which complicates the efforts to mitigate data bias. In the proposed NoTMF model, using seasonal differenced VAR is meaningful for characterizing traffic flow dynamics, such as global and local time series trends in the latent space, mitigating data biases (e.g., columns of data with varying missing rates, as shown in Figure 3), and reinforcing matrix factorization for more accurate imputation and prediction.

To summarize, the contributions of this study are threefold:

- We propose the NoTMF model, which integrates higher-order VAR processes with differencing operations into the classical low-rank matrix factorization framework. This approach enhances the model's ability to capture low-dimensional temporal dynamics within high-dimensional and sparse spatiotemporal datasets.
- We design an alternating minimization algorithm for efficiently solving the optimization problem in the NoTMF model. The subproblem involving the low-dimensional temporal factor matrix—a generalized Sylvester equation that incorporates both partially observed matrix factorization and seasonal differenced VAR—is solved using the conjugate gradient method, which approximates the least squares solution of linear equations with minimal iterations.
- We validate the NoTMF model's ability to forecast sparse movement speeds using two real-world datasets: the Uber movement speed data from New York City (NYC) and Seattle, USA, both of which encompass large numbers of road segments and exhibit strong seasonality. The experiments demonstrate the NoTMF model's superior performance in handling sparse data compared to existing state-of-the-art forecasting methods. In the meantime, the results highlight the importance of nonstationarity modeling with seasonal differenced VAR.

As we aim to formulate the sparse movement speed forecasting problem using data-driven machine learning methods, Table 1 provides a summary of basic symbols and notations used in this study. Here, \mathbb{R} represents the set of real numbers, while \mathbb{Z}^+ denotes the set of positive integers.

Table 1 Summary of the basic notation.

Notation	Description
$x \in \mathbb{R}$	Scalar
$\mathbf{x} \in \mathbb{R}^n$	Vector of length n
$\mathbf{X} \in \mathbb{R}^{m \times n}$	Matrix of size $m \times n$
$\frac{\partial f}{\partial \mathbf{x}}, \frac{\partial f}{\partial \mathbf{X}}$	Partial derivatives of f with respect to vector \mathbf{x} and matrix \mathbf{X} , respectively
$\delta \in \mathbb{Z}^+$	Forecasting time horizon
$R \in \mathbb{Z}^+$	Rank of matrix factorization
$[i]$	Positive integer set $\{1, 2, \dots, i\}, i \in \mathbb{Z}^+$
$[0, i]$	Nonnegative integer set $\{0, 1, 2, \dots, i\}, i \in \mathbb{Z}^+$
$[i, j]$	Positive integer set $\{i, i+1, \dots, j\}, i, j \in \mathbb{Z}^+, i < j$
$\ \cdot\ _2$	ℓ_2 -norm of vector
$\ \cdot\ _F$	Frobenius norm of matrix
\otimes	Kronecker product, see Definition 2
$\text{vec}(\cdot)$	Vectorization operator

2. Literature Review

2.1. Opportunities and Challenges of Using Ridesharing Vehicle Data in Traffic State Estimation

In urban transportation networks, the utilization of fixed sensing detectors such as loop detectors, radar detectors, and cameras has been pivotal in monitoring traffic flow. These detectors are strategically placed to capture traffic data, collecting key parameters such as traffic density, flow, and speed. The effectiveness of these sensors in providing real-time traffic information has been well-documented in several studies (Jain et al. 2019, Guerrero-Ibáñez et al. 2018). However, the advent of mobile sensor technology, including smartphones and vehicular sensors, has shifted the source of traffic data, offering a more dynamic and reliable approach to data collection (Janecek et al. 2015).

Ridesharing vehicle data in the form of trajectories has been increasingly utilized to reveal traffic states across cities (Zheng et al. 2018), providing valuable insights into human mobility patterns. These patterns are not only help in understanding the movement from one position to another but also extend to computing movement speeds within urban road networks. Despite its potential, the reliance on mobile sensor data presents challenges, particularly concerning data sparsity and insufficient sampling. Ridesharing vehicles represent only a small fraction of the total traffic, resulting in datasets that are far from comprehensive. This sparsity is compounded by issues of data corruption and missing values, making it difficult to obtain accurate and reliable traffic states. As a result, there has been growing interest in developing methodologies capable of forecasting movement speeds despite these limitations. Substantial progress has been made in mitigating the effects of imperfect and sparse data, aiming to improve the accuracy and reliability of traffic state estimations derived from ridesharing vehicle data (Chen and Sun 2022).

2.2. Machine Learning Methods

Matrix and tensor factorization techniques are widely used to explore traffic flow data. For example, Yu et al. (2016) establish a temporal regularized matrix factorization (TRMF) framework for forecasting high-dimensional and sparse traffic time series in latent spaces, while Chen and Sun (2022) develop a Bayesian temporal factorization for forecasting matrix and tensor time series data in the presence of missing values. The classical time series forecasting problem has been widely studied with a variety of statistics and machine learning frameworks over the past few decades. In this study, we aim to leverage matrix factorization and VAR to address the traffic flow forecasting problem with high-dimensional and sparse movement data.

In practice, on small-scale traffic time series datasets, VAR is a classical and efficient solution for traffic flow forecasting. However, when dealing with high-dimensional data, the traditional VAR model severely suffers from the over-parameterization issue (Velu et al. 1986, Velu and Reinsel 1998, Basu and Michailidis 2015, Wang et al. 2021, Cheng et al. 2022), where the number of parameters vastly exceeds the number of observed data samples. In such cases, incorporating prior knowledge and assumptions about the parameter space, such as low-rank property and sparsity of the coefficient matrices, becomes essential for seeking an efficient learning process. For example, in multivariate reduced-rank regression (Velu et al. 1986, Velu and Reinsel 1998), the coefficient matrices are assumed to be

low-rank, with variants of this method including high-dimensional VAR that use matrix factorization (e.g., (Carriero et al. 2016, Koop et al. 2019)) and tensor factorization (e.g., (Wang et al. 2021)). Imposing sparsity on the coefficients is another potential solution, such as using ℓ_1 -norm regularization (Basu and Michailidis 2015, Han et al. 2015) or combining nuclear norm and sparsity-induced norms (Basu et al. 2019).

On incomplete time series data, a central challenge is to effectively learn temporal dynamics from partially observed data. A naive solution is to first impute the missing data and then make predictions on the estimated data points (e.g., (Che et al. 2018)). However, such methods may produce substantial estimation biases due to the separation of imputation and forecasting, not to mention that making accurate imputation itself is not a trivial task (Chen and Sun 2022). For high-dimensional datasets with missing values, the low-rank assumption has been demonstrated to be very effective for imputation (Yu et al. 2016, Chen and Sun 2022). In the literature, matrix and tensor factorization methods have been used to model high-dimensional and incomplete time series (Xiong et al. 2010, Yu et al. 2016, Takeuchi et al. 2017, Chen and Sun 2022). The underlying assumption in these models is that the incomplete high-dimensional dataset can be well characterized by a few latent factors—analogueous to the state space representation (Stock and Watson 2016) in dynamic factor models—which evolve over time following VAR processes. However, modeling the incomplete data in a matrix factorization framework is more computationally efficient due to the simplified isotropic covariance structure on errors (Yu et al. 2016).

In addition to the high-dimensionality and sparsity issues, nonstationarity is also a unique feature of real-world time series data. Recall that the properties of stationary time series do not depend on time. Most existing literature simply assumes that the data and the underlying latent temporal factors are stationary when applying VAR for modeling temporal dynamics (Yu et al. 2016, Takeuchi et al. 2017, Gultekin and Paisley 2018, Chen and Sun 2022). Only a few matrix/tensor factorization-based time series models address the nonstationarity issue by modeling the periodicity (Dunlavy et al. 2011, Yu et al. 2016, Kawabata et al. 2021). As mentioned by Kawabata et al. (2021), to overcome the limitation of matrix factorization for seasonal modeling, one classical data augmentation approach for temporal modeling with periodic patterns is converting matrix factorization into tensor factorization according to season information. In the temporal regularized matrix factorization

(Yu et al. 2016), seasonality is considered by using a well-designed lag set for the autoregressive model. Shifting seasonal matrix factorization proposed by Kawabata et al. (2021) can learn multiple seasonal patterns/regimes from multi-viewed data, i.e., matrix-variate time series. These studies demonstrate the effectiveness of matrix and tensor factorization techniques for time series data with diverse periodic patterns. Nevertheless, to the best of our knowledge, introducing proper differencing operations—a simple yet effective solution to address the nonstationarity issue—has been overlooked by existing studies.

2.3. Research Gaps

By definition, one can represent the movement speed data collected from N road segments and T consecutive time steps (e.g., with an hourly time resolution) as the matrix $\mathbf{Y} \in \mathbb{R}^{N \times T}$ whose movement speeds at time t are written as $\mathbf{y}_t \in \mathbb{R}^N$. The goal of movement speed forecasting is to estimate the future movement speeds $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times \delta}$ for the next δ time steps ahead. A classical and widely used approach for multivariate time series forecasting is the VAR process (Lütkepohl 2013):

$$\mathbf{y}_t = \sum_{k \in [d]} \mathbf{A}_k \mathbf{y}_{t-k} + \boldsymbol{\epsilon}_t, \forall t \in [d+1, T], \quad (1)$$

where $d \in \mathbb{Z}^+$ represents the order of VAR, $\mathbf{A}_k \in \mathbb{R}^{N \times N}$, $k \in [d]$ are the coefficient matrices, and $\boldsymbol{\epsilon}_t \in \mathbb{R}^N$ is the zero-mean Gaussian noise vector. With this formula, the coefficient matrices can capture coevolution movement patterns over different road segments. Typically, the standard VAR is well-suited to “short-fat” data with $N \ll T$. However, the high-dimensionality has posed methodological challenges to the implementation of VAR due to the quadratically growing parameter space (Verleysen and François 2005, Basu et al. 2019, Wang et al. 2021). On the “tall-skinny” data with $N \gg T$, it gives rise to the over-parameterization issue (Velu et al. 1986, Velu and Reinsel 1998, Basu and Michailidis 2015) with dN^2 parameters in (1), vastly exceeding NT observations. Regarding the sparsity of urban movement speed data, it prevents us from using VAR given that the model usually requires complete observations for parameter estimation.

To simultaneously address the high-dimensionality and sparsity issues, recent advances have integrated time series autoregression into the low-rank matrix/tensor factorization models (Xiong et al. 2010, Yu et al. 2016, Gultekin and Paisley 2018, Chen and Sun

2022), constructing a unified framework such as TMF. The TMF assumes that the movement speed data is dominated by a few “important” patterns (i.e., characterized by low-dimensional latent factors) and the temporal factor matrix captures the coevolution patterns of N movement speed time series. In the literature, substantial progress has been made in verifying the effectiveness of TMF on many real-world time series data in the presence of missing values (Yu et al. 2016, Gultekin and Paisley 2018, Chen and Sun 2022). Nevertheless, some methodological challenges remain in learning the temporal dependencies. It is worth noting that the default TMF does not impose any constraints on the stationarity of latent temporal factors. This becomes problematic when modeling movement speed data with strong trends and seasonality. In this case, the estimated coefficient matrices are no longer effective because we would expect the underlying dynamics of temporal factors to be time-dependent and thus nonstationary. If the movement speed data is fully observed, a simple solution is to first perform certain differencing operations such as first-order differencing and seasonal differencing, and then use the differenced data for parameter estimation. However, this simple solution becomes inaccessible when the data is sparse. To this end, we intend to formulate the movement speed forecasting problem with NoTMF and present an efficient solution algorithm with alternating minimization and conjugate gradient methods.

3. Nonstationary Temporal Matrix Factorization

3.1. Preliminaries: Matrix Factorization

The movement speed data in urban road networks typically demonstrates certain low-dimensional spatial and temporal patterns. For instance, as shown in Figure 1, the sparse movement speed data $\mathbf{Y} \in \mathbb{R}^{N \times T}$ can be approximated by the multiplication between a spatial factor matrix $\mathbf{W} \in \mathbb{R}^{R \times N}$ and a temporal factor matrix $\mathbf{X} \in \mathbb{R}^{R \times T}$, i.e., $\mathbf{Y} \approx \mathbf{W}^\top \mathbf{X}$ with a prescribed rank $R \in \mathbb{Z}^+ < \min\{N, T\}$. In practice, the observed data \mathbf{Y} is usually incomplete, which contains some missing/unobserved values in the data. Here we define $\mathcal{P}_\Omega(\cdot)$ as the orthogonal projection supported on the observed index set Ω to indicate the observed entries. For instance, on the data matrix \mathbf{Y} with observed index set Ω , we have the following definition for the orthogonal projection:

$$[\mathcal{P}_\Omega(\mathbf{Y})]_{n,t} = \begin{cases} y_{n,t}, & \text{if } (n,t) \in \Omega, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where $n \in [N]$ and $t \in [T]$. Notably, $[\cdot]_{n,t}$ denotes the (n, t) -th entry of a matrix. The entries of \mathbf{Y} filled with zeros imply the unobserved values, i.e., $(n, t) \notin \Omega$. Such a definition allows one to formulate the matrix factorization on the partially observed movement speed data.

As a consequence, the matrix factorization can be formulated as $\mathcal{P}_\Omega(\mathbf{Y}) \approx \mathcal{P}_\Omega(\mathbf{W}^\top \mathbf{X})$ or element-wise $y_{n,t} \approx \mathbf{w}_n^\top \mathbf{x}_t$, $\forall n \in [N], t \in [T]$ in which $\mathbf{w}_n, \mathbf{x}_t \in \mathbb{R}^R$ are the n -th and the t -th columns of the factor matrices \mathbf{W} and \mathbf{X} , respectively. To achieve such approximation, one key problem is specifying the latent factor matrices, so that the partially specified data matrix \mathbf{Y} matches $\mathbf{W}^\top \mathbf{X}$ as closely as possible (Koren et al. 2009). Formally, the optimization problem of matrix factorization is given by

$$\min_{\mathbf{W}, \mathbf{X}} \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2), \quad (3)$$

or on the observed entries of data matrix \mathbf{Y} , we have

$$\min_{\{\mathbf{w}_n\}_{n \in [N]}, \{\mathbf{x}_t\}_{t \in [T]}} \frac{1}{2} \sum_{(n,t) \in \Omega} (y_{n,t} - \mathbf{w}_n^\top \mathbf{x}_t)^2 + \frac{\rho}{2} \left(\sum_{n \in [N]} \|\mathbf{w}_n\|_2^2 + \sum_{t \in [T]} \|\mathbf{x}_t\|_2^2 \right), \quad (4)$$

with an objective function defined by the error of matrix factorization on the set Ω and the regularization terms (weighted by the hyperparameter ρ) corresponding to factor matrices \mathbf{W} and \mathbf{X} . Alternatively, the column vector $\mathbf{y}_t \in \mathbb{R}^N$, $\forall t \in [T]$ of the data \mathbf{Y} can be approximated by $\mathbf{W}^\top \mathbf{x}_t$. To characterize the temporal dynamics of urban movement speed, we next introduce the NoTMF model which integrates the seasonal differenced VAR into the matrix factorization framework. By doing so, NoTMF can build complicated temporal correlations when forecasting high-dimensional and sparse movement speeds of urban road networks.

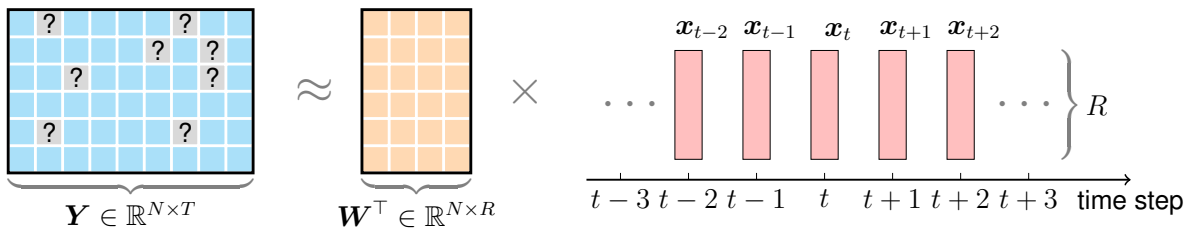


Figure 1 Illustration of TMF on sparse movement speed data where the symbols “?” represent the unobserved values in the data. TMF characterizes spatiotemporal patterns of the data $\mathbf{Y} \in \mathbb{R}^{N \times T}$ as a spatial factor matrix $\mathbf{W} \in \mathbb{R}^{R \times N}$ and a temporal factor matrix $\mathbf{X} \in \mathbb{R}^{R \times T}$ in which the temporal factor matrix is indeed a multivariate time series.

3.2. Optimization Problem of NoTMF

TMF that integrated temporal modeling techniques such as univariate autoregression and VAR has become extremely useful for multivariate time series forecasting in the presence of missing values (Yu et al. 2016, Gultekin and Paisley 2018, Chen and Sun 2022). The essential idea of the TMF models is that low-rank matrix factorization can discover low-dimensional temporal patterns from partially observed time series, while autoregression can reinforce the temporal correlations in the modeling process and capture time-evolving coefficients. To this end, we first characterize time series correlations and patterns from sparse time series data and then make efficient predictions for future data points. At the same time, to address the real-world sparse prediction problem, the design of TMF stems from some classical machine learning frameworks such as online learning and dictionary learning.

TMF is well-suited to handling the following emerging issues in real-world movement speed data: 1) High-dimensionality: Urban road networks such as transportation systems in NYC usually have thousands of road segments, i.e., large N ; 2) Sparsity: Road-level movement speed data are often sparse, which means that only a small portion of data are observed due to the limited penetration of float cars (e.g., taxi and ridesharing vehicle). On Uber movement speed datasets, we only have access to a small portion of movement speed values even in an hourly time resolution due to insufficient sampling and the limited penetration of ridesharing vehicles.

Recall that an essential assumption of VAR is that the modeled time series is stationary. However, the periodicity/seasonality in real-world movement speed data \mathbf{Y} will be characterized by the latent temporal factor matrix \mathbf{X} , and such nonstationarity property contradicts the stationary assumption in VAR. Furthermore, the estimated coefficients in the VAR process would become sub-optimal and even produce biased estimations for the movement speed. To address this issue, we expect the temporal dynamics to be time-dependent and introduce seasonal differencing to stationarize traffic time series dynamics in latent spaces. For any sparse movement speed data $\mathbf{Y} \in \mathbb{R}^{N \times T}$ collected from N road segments and T (hourly) time steps, we formulate the NoTMF model as the following optimization problem:

$$\min_{\mathbf{W}, \mathbf{X}, \{\mathbf{A}_k\}_{k \in [d]}} \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2 + \frac{\gamma}{2} \sum_{t \in [d+m+1, T]} \left\| \dot{\mathbf{x}}_t^m - \sum_{k \in [d]} \mathbf{A}_k \dot{\mathbf{x}}_{t-k}^m \right\|_2^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2), \quad (5)$$

where the season is set as $m \in \mathbb{Z}^+$ and we let

$$\dot{\mathbf{x}}_t^m \triangleq \mathbf{x}_t - \mathbf{x}_{t-m}, \forall t \in [m+1, T], \quad (6)$$

be the result of the season- m differencing on the columns of the temporal factor matrix \mathbf{X} . On the seasonal different temporal factors, $\mathbf{A}_k \in \mathbb{R}^{R \times R}$, $k \in [d]$ represent the coefficient matrices of d th-order VAR for characterizing the temporal dynamics of \mathbf{X} . The errors of seasonal differenced VAR are formulated as a regularization term, leading to the design of temporal loss. Hyperparameters $\{\gamma, \rho\}$ are the weight parameters for the regularization terms.

In practice, the objective function in (5) has three components, including the loss function of matrix factorization, the temporal loss of seasonal differenced VAR, and the regularization terms. The matrix factorization can help recover missing values in the matrix \mathbf{Y} through $\mathbf{W}^\top \mathbf{X}$, but the data biases (e.g., the columns with different amounts of observed entries) possibly mislead the loss function behaviors. Thus, the temporal loss is of great significance for mitigating data biases of urban movement speeds. In theory, our NoTMF takes the form of linear time-invariant systems and has an observation equation and a latent space equation, namely,

$$\begin{cases} \mathcal{P}_{\Omega_t}(\mathbf{y}_t) = \mathcal{P}_{\Omega_t}(\mathbf{W}^\top \mathbf{x}_t) + \boldsymbol{\epsilon}_t, & \text{(Observation equation)} \\ \dot{\mathbf{x}}_t^m = \sum_{k \in [d]} \mathbf{A}_k \dot{\mathbf{x}}_{t-k}^m + \boldsymbol{\eta}_t, & \text{(Latent space equation)} \end{cases} \quad (7)$$

where Ω_t is the observed index set of the high-dimensional movement speeds $\mathbf{y}_t \in \mathbb{R}^N$ at time t . The vectors $\boldsymbol{\epsilon}_t \in \mathbb{R}^N$ and $\boldsymbol{\eta}_t \in \mathbb{R}^R$ are zero-mean Gaussian noises. As can be seen, the observation equation describes the relation between the partially observed variables \mathbf{y}_t , $\forall t \in [T]$ and spatial/temporal factors with matrix factorization, while the latent space equation reflects the dynamics of the unobserved factor variables \mathbf{x}_t , $\forall t \in [T]$ with VAR. A phenomenon related to (7) is the existence of “representative” temporal dynamics of urban traffic flow which comes from thousands of road segments. Despite the sparse learning process in NoTMF, the fundamental idea of NoTMF can be easily connected with some classical methods such as dynamic factor model in macroeconomic modeling (Forni et al. 2000, Stock and Watson 2016), state space model in system identification and control (Hangos et al. 2006), and recurrent neural networks in deep learning (LeCun et al. 2015, Prince 2023).

3.3. Matrix Representation: Temporal Operator Matrices

As mentioned above, the optimization problem of NoTMF has a matrix factorization component in the form of matrices and a VAR component in the form of vectors. To get a unified solution to the optimization problem, we rewrite the vector-form temporal loss in the form of matrices by utilizing the temporal operator matrices as described in Definition 1.

DEFINITION 1 (TEMPORAL OPERATOR MATRICES). According to the expression of d th-order VAR on the latent temporal factors $\{\mathbf{x}_t\}_{t \in [T]}$, the temporal operator matrices $\{\Psi_k\}_{k \in [0, d]}$ associated with the season $m \in \mathbb{Z}^+$ are defined as follows,

$$\begin{aligned} \Psi_k \triangleq & \begin{bmatrix} \mathbf{0}_{(T-d-m) \times (d-k)} & -\mathbf{I}_{T-d-m} & \mathbf{0}_{(T-d-m) \times (k+m)} \end{bmatrix} \\ & + \begin{bmatrix} \mathbf{0}_{(T-d-m) \times (d+m-k)} & \mathbf{I}_{T-d-m} & \mathbf{0}_{(T-d-m) \times k} \end{bmatrix} \\ & \in \mathbb{R}^{(T-d-m) \times T}, \forall k \in [0, d], \end{aligned} \quad (8)$$

where $\mathbf{0}_{m \times n}$ denotes the m -by- n matrix of zeros, and \mathbf{I}_m denotes the m -by- m identity matrix. In (8), the temporal operator matrix Ψ_k is an augmented matrix, i.e., the combination of two matrices. In particular, the first matrix has zeros as the entries in the first $d-k$ columns and the last $k+m$ columns, while the second matrix has zeros as the entries in the first $d+m-k$ columns and the last k columns.

REMARK 1. Since these temporal operator matrices $\{\Psi_k\}_{k \in [0, d]}$ are a sequence of sparse matrices and only have $2(T-d-m)$ nonnegative entries, utilizing such sparse structure allows one to reduce the memory consumption and develop efficient algorithms in the case of a very long time series, i.e., large T .

As a result, one can use the multiplication between the temporal factor matrix \mathbf{X} and the temporal operator matrices to reformulate the temporal loss. First, the season- m dif-

ferenced temporal factors can be written in the form of matrices:

$$\begin{aligned}
\mathbf{X}\Psi_0^\top &= \begin{bmatrix} & | & & | \\ \mathbf{x}_{d+m+1} - \mathbf{x}_{d+1} & \cdots & \mathbf{x}_T - \mathbf{x}_{T-m} & \\ & | & & | \end{bmatrix}, \\
\mathbf{X}\Psi_1^\top &= \begin{bmatrix} & | & & | \\ \mathbf{x}_{d+m} - \mathbf{x}_d & \cdots & \mathbf{x}_{T-1} - \mathbf{x}_{T-m-1} & \\ & | & & | \end{bmatrix}, \\
&\vdots \\
\mathbf{X}\Psi_d^\top &= \begin{bmatrix} & | & & | \\ \mathbf{x}_{m+1} - \mathbf{x}_1 & \cdots & \mathbf{x}_{T-d} - \mathbf{x}_{T-m-d} & \\ & | & & | \end{bmatrix},
\end{aligned} \tag{9}$$

of size $R \times (T - d - m)$. Then, one can eliminate the sum over all $t \in [d + m + 1, T]$ in the temporal loss and find an equivalent expression, namely,

$$\sum_{t \in [d+m+1, T]} \left\| \dot{\mathbf{x}}_t^m - \sum_{k \in [d]} \mathbf{A}_k \dot{\mathbf{x}}_{t-k}^m \right\|_2^2 = \left\| \mathbf{X}\Psi_0^\top - \sum_{k \in [d]} \mathbf{A}_k \mathbf{X}\Psi_k^\top \right\|_F^2. \tag{10}$$

By doing so, the matrix-form temporal loss can facilitate the matrix computations. Thus, the optimization problem of NoTMF in (5) now becomes

$$\min_{\mathbf{W}, \mathbf{X}, \{\mathbf{A}_k\}_{k \in [d]}} \frac{1}{2} \left\| \mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) \right\|_F^2 + \frac{\gamma}{2} \left\| \mathbf{X}\Psi_0^\top - \sum_{k \in [d]} \mathbf{A}_k \mathbf{X}\Psi_k^\top \right\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2), \tag{11}$$

where the latent factor matrices $\{\mathbf{W}, \mathbf{X}\}$ and the coefficient matrices $\{\mathbf{A}_k\}_{k \in [d]}$ are required to estimate. If one obtains both spatial factor matrix \mathbf{W} and temporal factor matrix \mathbf{X} , then the unobserved entries in the data \mathbf{Y} can be reconstructed accordingly. In particular, the VAR process can reinforce the temporal correlations among $\mathbf{x}_t, \forall t \in [T]$ and capture temporal dynamics, while the matrix factorization without temporal modeling is invariant to the permutation of column vectors $\{\mathbf{x}_t\}_{t \in [T]}$ in the temporal factor matrix \mathbf{X} . We next introduce an alternating minimization framework to solve the optimization problem of NoTMF in (11), in which we denote the objective function by f .

3.4. Spatial Factor Matrix Estimation

The spatial factor matrix \mathbf{W} represents low-dimensional patterns of the sparse movement speed data \mathbf{Y} . In the alternating minimization framework, estimating the spatial factor

matrix \mathbf{W} requires one to fix the temporal factor matrix \mathbf{X} and the coefficient matrices $\{\mathbf{A}_k\}_{k \in [d]}$ as known variables. First of all, we write down the partial derivative of f with respect to \mathbf{W} as follows,

$$\frac{\partial f}{\partial \mathbf{W}} = -\mathbf{X} \mathcal{P}_\Omega^\top (\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) + \rho \mathbf{W}. \quad (12)$$

Let $\frac{\partial f}{\partial \mathbf{W}} = \mathbf{0}$, then we have the following generalized Sylvester equation:

$$\mathbf{X} \mathcal{P}_\Omega^\top (\mathbf{W}^\top \mathbf{X}) + \rho \mathbf{W} = \mathbf{X} \mathcal{P}_\Omega^\top (\mathbf{Y}). \quad (13)$$

In fact, (13) is a generalized system of linear equations, and it is not difficult to obtain the closed-form solution. A natural solution could be pursued through the least squares of each column vector of \mathbf{W} (Koren et al. 2009):

$$\mathbf{w}_n = \left(\sum_{t:(n,t) \in \Omega} \mathbf{x}_t \mathbf{x}_t^\top + \rho \mathbf{I}_R \right)^{-1} \left(\sum_{t:(n,t) \in \Omega} \mathbf{x}_t y_{n,t} \right), \forall n \in [N], \quad (14)$$

where the notation $\sum_{t:(n,t) \in \Omega}$ implies the sum over all t in the observed index set Ω with the fixed index n .

3.5. Temporal Factor Matrix Estimation: Factorization Plus Autoregression

The temporal factor matrix \mathbf{X} represents low-dimensional dynamic patterns of the sparse movement speed data \mathbf{Y} . We assume a seasonal differencing on the nonstationary temporal factor matrix and assert that such a VAR process can reinforce the temporal modeling capability of matrix factorization on sparse movement speed data. However, the challenge would arise because VAR on the seasonal differenced temporal factor matrix complicates the optimization problem. With respect to the temporal factor matrix \mathbf{X} , the partial derivative of f can be written as follows,

$$\frac{\partial f}{\partial \mathbf{X}} = -\mathbf{W} \mathcal{P}_\Omega (\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) + \gamma \sum_{k \in [0,d]} \mathbf{A}_k^\top \left(\sum_{h \in [0,d]} \mathbf{A}_h \mathbf{X} \Psi_h^\top \right) \Psi_k + \rho \mathbf{X}, \quad (15)$$

where $\mathbf{A}_0 \triangleq -\mathbf{I}_R$ is introduced as a negative identity matrix.

Let $\frac{\partial f}{\partial \mathbf{X}} = \mathbf{0}$, then we have a generalized Sylvester equation with multiple terms:

$$\mathbf{W} \mathcal{P}_\Omega (\mathbf{W}^\top \mathbf{X}) + \gamma \sum_{k \in [0,d]} \mathbf{A}_k^\top \left(\sum_{h \in [0,d]} \mathbf{A}_h \mathbf{X} \Psi_h^\top \right) \Psi_k + \rho \mathbf{X} = \mathbf{W} \mathcal{P}_\Omega (\mathbf{Y}), \quad (16)$$

which involves $d^2 + 2$ terms in the left-hand side. Both orthogonal projection in matrix factorization and VAR on the temporal factor matrix complicate this matrix equation. If we convert the formula into a generalized system of linear equations, then finding the closed-form solution to (16) would take $\mathcal{O}(R^3T^3)$ time, infeasible as the dimension RT is large, see details as shown in Lemma 2.

LEMMA 1. *Suppose $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{X} \in \mathbb{R}^{n \times p}$, and $\mathbf{B} \in \mathbb{R}^{p \times q}$ be three matrices commensurate from multiplication in that order, then it always hold that*

$$\text{vec}(\mathbf{AXB}) = (\mathbf{B}^\top \otimes \mathbf{A}) \text{vec}(\mathbf{X}), \quad (17)$$

where $\text{vec}(\cdot)$ denotes the vectorization operator, and \otimes denotes the Kronecker product (see Definition 2). The aforementioned formula is also known as the mixed-product property of Kronecker product (Golub and Van Loan 2013).

DEFINITION 2 (KRONECKER PRODUCT (GOLUB AND VAN LOAN 2013)). For any matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$, the Kronecker product between \mathbf{A} and \mathbf{B} is given by

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2n}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{(mp) \times (nq)}, \quad (18)$$

where the resulting matrix is of size $(mp) \times (nq)$ with $m \times n$ blocks, and each block is the multiplication between the matrix \mathbf{B} and the entry of \mathbf{A} .

LEMMA 2. *Given any partially observed matrix $\mathbf{Y} \in \mathbb{R}^{N \times T}$ with the observed index set Ω , the closed-form solution to (16) in the form of vectorized \mathbf{X} is given by*

$$\text{vec}(\mathbf{X}) = \left(\mathbf{S} + \gamma \sum_{k \in [0, d]} \sum_{h \in [0, d]} (\Psi_k \otimes \mathbf{A}_k)^\top (\Psi_h \otimes \mathbf{A}_h) + \rho \mathbf{I}_{RT} \right)^{-1} \text{vec}(\mathbf{W}\mathcal{P}_\Omega(\mathbf{Y})), \quad (19)$$

where the block matrix

$$\mathbf{S} \triangleq \begin{bmatrix} \mathbf{S}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{S}_T \end{bmatrix} \in \mathbb{R}^{(RT) \times (RT)},$$

has a sequence of blocks on the diagonal such that

$$\mathbf{S}_t \triangleq \sum_{n:(n,t) \in \Omega} \mathbf{w}_n \mathbf{w}_n^\top \in \mathbb{R}^{R \times R}, \forall t \in [T],$$

where the notation $\sum_{n:(n,t) \in \Omega}$ implies the sum over all n in the observed index set Ω with the fixed index t .

REMARK 2. If we let the first term of the left-hand side in (16) as

$$\mathbf{C} \triangleq \mathbf{W} \mathcal{P}_\Omega (\mathbf{W}^\top \mathbf{X}) \in \mathbb{R}^{R \times T}, \quad (20)$$

then the t -th column vector $\mathbf{c}_t \in \mathbb{R}^R$ of the matrix \mathbf{C} is given by

$$\mathbf{c}_t = \sum_{n:(n,t) \in \Omega} \mathbf{w}_n \mathbf{w}_n^\top \mathbf{x}_t = \mathbf{S}_t \mathbf{x}_t \Rightarrow \text{vec}(\mathbf{C}) = \mathbf{S} \text{vec}(\mathbf{X}). \quad (21)$$

In the meantime, if we let the VAR components in (16) be

$$\mathbf{E} \triangleq \sum_{k \in [0,d]} \mathbf{A}_k^\top \left(\sum_{h \in [0,d]} \mathbf{A}_h \mathbf{X} \Psi_h^\top \right) \Psi_k \in \mathbb{R}^{R \times T}, \quad (22)$$

then according to the mixed-product property of Kronecker product in Lemma 2, we have

$$\begin{aligned} \text{vec}(\mathbf{E}) &= \sum_{k \in [0,d]} (\Psi_k^\top \otimes \mathbf{A}_k^\top) \text{vec} \left(\sum_{h \in [0,d]} \mathbf{A}_h \mathbf{X} \Psi_h^\top \right) \\ &= \sum_{k \in [0,d]} \sum_{h \in [0,d]} (\Psi_k \otimes \mathbf{A}_k)^\top (\Psi_h \otimes \mathbf{A}_h) \text{vec}(\mathbf{X}). \end{aligned} \quad (23)$$

As a consequence, (16) is equivalent to

$$\left(\mathbf{S} + \gamma \sum_{k \in [0,d]} \sum_{h \in [0,d]} (\Psi_k \otimes \mathbf{A}_k)^\top (\Psi_h \otimes \mathbf{A}_h) + \rho \mathbf{I}_{RT} \right) \text{vec}(\mathbf{X}) = \text{vec}(\mathbf{W} \mathcal{P}_\Omega (\mathbf{Y})), \quad (24)$$

and thus leading to (19) as claimed.

However, the vectorized closed-form solution in (19) is memory-consuming and computationally expensive for large problems. In large applications, as suggested by Rao et al. (2015), Chi et al. (2019), the alternating minimization with conjugate gradient for solving matrix factorization problems is both efficient and scalable. The conjugate gradient method allows one to search for the approximated solution to a system of linear equations with a relatively small number of iterations (e.g., 5 or 10). Therefore, to estimate the temporal

factor matrix \mathbf{X} , we develop an efficient conjugate gradient routine. We first define the left-hand side of (24) as the function $\theta_x: \mathbb{R}^{R \times T} \rightarrow \mathbb{R}^{RT}$, whose formula is given by

$$\theta_x(\mathbf{X}) \triangleq \left(\mathbf{S} + \gamma \sum_{k \in [0, d]} \sum_{h \in [0, d]} (\boldsymbol{\Psi}_k \otimes \mathbf{A}_k)^\top (\boldsymbol{\Psi}_h \otimes \mathbf{A}_h) + \rho \mathbf{I}_{RT} \right) \text{vec}(\mathbf{X}), \quad (25)$$

which is equivalent to

$$\theta_x(\mathbf{X}) = \text{vec} \left(\mathbf{W} \mathcal{P}_\Omega(\mathbf{W}^\top \mathbf{X}) + \gamma \sum_{k \in [0, d]} \mathbf{A}_k^\top \left(\sum_{h \in [0, d]} \mathbf{A}_h \mathbf{X} \boldsymbol{\Psi}_h^\top \right) \boldsymbol{\Psi}_k + \rho \mathbf{X} \right). \quad (26)$$

Accordingly, we let the right-hand side of (24) be

$$\psi_x \triangleq \text{vec}(\mathbf{W} \mathcal{P}_\Omega(\mathbf{Y})). \quad (27)$$

As mentioned above, (24) is in the form of a system of linear equations, and its system matrix $\mathbf{S} + \gamma \sum_{k \in [0, d]} \sum_{h \in [0, d]} (\boldsymbol{\Psi}_k \otimes \mathbf{A}_k)^\top (\boldsymbol{\Psi}_h \otimes \mathbf{A}_h) + \rho \mathbf{I}_{RT}$ is a symmetric positive-definite matrix. The conjugate gradient method is well-suited to such Kronecker-structured linear equations. Algorithm 1 summarizes the estimation procedure for approximating the least squares solution to \mathbf{X} in (19). Alternatively, we can use the conjugate gradient to solve (13) when estimating the spatial factor matrix \mathbf{W} .

3.6. Coefficient Matrix Estimation

The coefficient matrices $\{\mathbf{A}_k\}_{k \in [d]}$ in NoTMF represent temporal correlations of the temporal factors $\{\mathbf{x}_t\}_{t \in [T]}$. According to the property of the Kronecker product, one can rewrite the formula such that

$$\begin{aligned} \sum_{k \in [d]} \mathbf{A}_k \mathbf{X} \boldsymbol{\Psi}_k^\top &= \mathbf{A}_1 \mathbf{X} \boldsymbol{\Psi}_1^\top + \mathbf{A}_2 \mathbf{X} \boldsymbol{\Psi}_2^\top + \cdots + \mathbf{A}_d \mathbf{X} \boldsymbol{\Psi}_d^\top \\ &= \underbrace{\begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_d \end{bmatrix}}_{=\mathbf{I}_d \otimes \mathbf{X}} \begin{bmatrix} \mathbf{X} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{X} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{X} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}_1^\top \\ \boldsymbol{\Psi}_2^\top \\ \vdots \\ \boldsymbol{\Psi}_d^\top \end{bmatrix}, \end{aligned} \quad (28)$$

and as a consequence, the subproblem with respect to the coefficient matrices can be formulated as

$$\min_{\mathbf{A}} \frac{\gamma}{2} \|\mathbf{X} \boldsymbol{\Psi}_0^\top - \mathbf{A}(\mathbf{I}_d \otimes \mathbf{X}) \boldsymbol{\Psi}^\top\|_F^2, \quad (29)$$

Algorithm 1 Conjugate Gradient for Approximating \mathbf{X}

Input: Data matrix \mathbf{Y} with the observed index set Ω , spatial factor matrix \mathbf{W} , initialized temporal factor matrix \mathbf{X} , coefficient matrices $\{\mathbf{A}_k\}_{k \in [d]}$, and maximum iteration ℓ_{\max} (e.g., $\ell_{\max} = 5$).

Output: Estimated temporal factor matrix \mathbf{X} .

- 1: Initialize \mathbf{x}_0 by the vectorized \mathbf{X} .
- 2: Compute residual vector $\mathbf{r}_0 = \psi_x - \theta_x(\mathbf{X})$, and let $\mathbf{d}_0 = \mathbf{r}_0$.
- 3: **for** $\ell = 0$ to $\ell_{\max} - 1$ **do**
- 4: Convert vector \mathbf{d}_ℓ into matrix \mathbf{D}_ℓ .
- 5: Compute $\alpha_\ell = \frac{\mathbf{r}_\ell^\top \mathbf{r}_\ell}{\mathbf{d}_\ell^\top \theta_x(\mathbf{D}_\ell)}$.
- 6: Update $\mathbf{x}_{\ell+1} = \mathbf{x}_\ell + \alpha_\ell \mathbf{d}_\ell$.
- 7: Update $\mathbf{r}_{\ell+1} = \mathbf{r}_\ell - \alpha_\ell \theta_x(\mathbf{D}_\ell)$.
- 8: Compute $\beta_\ell = \frac{\mathbf{r}_{\ell+1}^\top \mathbf{r}_{\ell+1}}{\mathbf{r}_\ell^\top \mathbf{r}_\ell}$.
- 9: Update $\mathbf{d}_{\ell+1} = \mathbf{r}_{\ell+1} + \beta_\ell \mathbf{d}_\ell$.
- 10: **end for**
- 11: Convert vector $\mathbf{x}_{\ell_{\max}}$ into matrix \mathbf{X} .

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_d \end{bmatrix} \in \mathbb{R}^{R \times (dR)},$$

$$\mathbf{\Psi} = \begin{bmatrix} \mathbf{\Psi}_1 & \mathbf{\Psi}_2 & \cdots & \mathbf{\Psi}_d \end{bmatrix} \in \mathbb{R}^{(T-d-m) \times (dT)},$$

are augmented matrices. As each matrix has d sub-matrices, \mathbf{A} and $\mathbf{\Psi}$ show dR and dT columns, respectively. Since we build correlations by VAR on the temporal factors, the subproblem would only demonstrate the VAR component. The decision variable of this subproblem is the augmented coefficient matrix \mathbf{A} . The partial derivative of f with respect to the matrix \mathbf{A} is given by

$$\frac{\partial f}{\partial \mathbf{A}} = -\gamma \mathbf{X} \mathbf{\Psi}_0^\top \mathbf{\Psi} (\mathbf{I}_d \otimes \mathbf{X})^\top + \gamma \mathbf{A} (\mathbf{I}_d \otimes \mathbf{X}) \mathbf{\Psi}^\top \mathbf{\Psi} (\mathbf{I}_d \otimes \mathbf{X})^\top. \quad (30)$$

Let $\frac{\partial f}{\partial \mathbf{A}} = \mathbf{0}$, then we have a least squares solution as

$$\mathbf{A} = \mathbf{X} \mathbf{\Psi}_0^\top ((\mathbf{I}_d \otimes \mathbf{X}) \mathbf{\Psi}^\top)^\dagger, \quad (31)$$

where \cdot^\dagger denotes the Moore-Penrose pseudo-inverse.

3.7. Solution Algorithm of NoTMF

As mentioned above, the alternating minimization scheme of NoTMF possesses the closed-form solutions to all three variables $\{\mathbf{W}, \mathbf{X}, \mathbf{A}\}$. For the spatial factor matrix \mathbf{W} , it is not difficult to write down the least squares solutions to the column vectors $\{\mathbf{w}_n\}_{n \in [N]}$ (see (14)), but one can also compute the approximated solution to the matrix \mathbf{W} from (13) with conjugate gradient. For the temporal factor matrix \mathbf{X} , computing from (24) with least squares is not as efficient as computing from (16) with conjugate gradient (see Algorithm 1). Algorithm 2 summarizes the implementation of NoTMF. In fact, the alternating minimization of NoMTF is the block coordinate minimization with three blocks/variables $\{\mathbf{W}, \mathbf{X}, \mathbf{A}\}$. One can justify our algorithm decreases the objective function monotonically in the iterative process and the NoTMF algorithm can be converged.

Algorithm 2 NoTMF($\mathbf{Y}, \Omega, d, R, \gamma, \rho, m$)

Input: Data matrix \mathbf{Y} with the observed index set Ω , order $d \in \mathbb{Z}^+$ and season $m \in \mathbb{Z}^+$ of the VAR, rank $R \in \mathbb{Z}^+$, and hyperparameters $\{\gamma, \rho\}$.

Output: Spatial factor matrix \mathbf{W} , temporal factor matrix \mathbf{X} , and coefficient matrix \mathbf{A} .

- 1: Initialize the variables $\{\mathbf{W}, \mathbf{X}, \mathbf{A}\}$.
 - 2: Generate temporal operator matrices $\{\Psi_k\}_{k \in [0, d]}$.
 - 3: **repeat**
 - 4: Compute \mathbf{W} as the least squares solution (or conjugate gradient).
 - 5: Compute \mathbf{X} from (16) with conjugate gradient.
 - 6: Compute \mathbf{A} by the least squares solution.
 - 7: **until** convergence
-

3.8. Rolling Forecasting with Dictionary Learning

In the following, we introduce the forecasting scheme for NoTMF. For example, Figure 2 shows how NoTMF works on the single-step movement speed forecasting in the latent spaces. At the t -th time step, as shown in Figure 2(a), we can first estimate the spatial factor matrix, the temporal factor matrix, and the coefficient matrix on the sparse input data $\mathbf{Y}_t \in \mathbb{R}^{N \times t}$ (see Algorithm 2), then we can forecast the next-step temporal factors as $\hat{\mathbf{x}}_{t+1}$ by using seasonal differenced VAR. As we have both spatial factor matrix \mathbf{W} and temporal factors $\hat{\mathbf{x}}_{t+1}$, it is not hard to generate the future movement speed values

as $\hat{\mathbf{y}}_{t+1} = \mathbf{W}^\top \hat{\mathbf{x}}_{t+1}$. Since we already have the spatial factor matrix \mathbf{W} , we can fix this variable as a dictionary matrix and instantly compute the temporal factor matrix and the coefficient matrix at the $t+1$ -th time step, referring to the online dictionary learning as mentioned by [Gultekin and Paisley \(2018\)](#). In Figure 2(b), we can first use input data $\mathbf{Y}_{t+1} \in \mathbb{R}^{N \times (t+1)}$ to update the temporal factor matrix and the coefficient matrix. Then, the seasonal differenced VAR allows one to forecast the next step temporal factors as $\hat{\mathbf{x}}_{t+2}$ and generate the future movement speed values as $\hat{\mathbf{y}}_{t+2} = \mathbf{W}^\top \hat{\mathbf{x}}_{t+2}$.

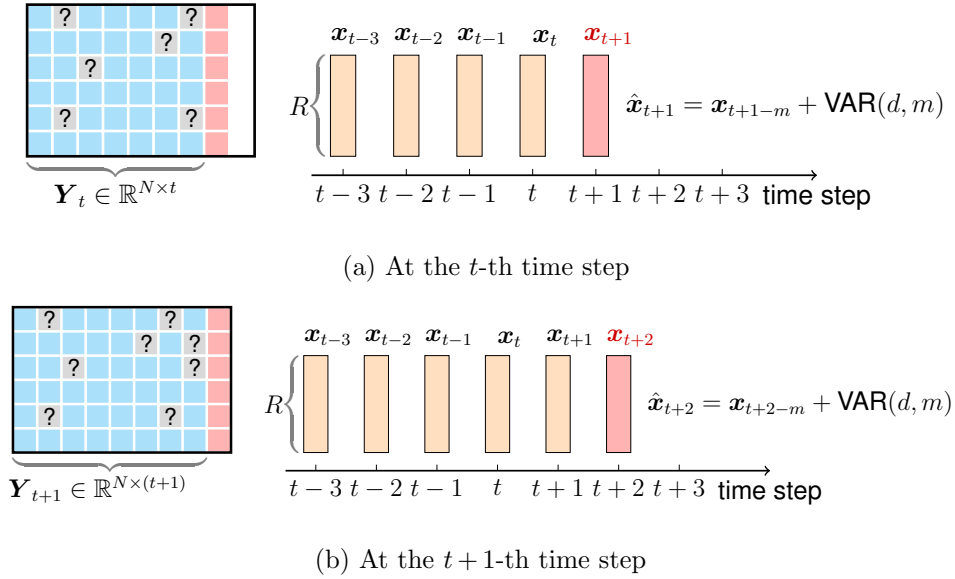


Figure 2 Sparse movement speed forecasting with NoTMF in which $\text{VAR}(d, m)$ is the VAR with the order d and the season m .

In this work, we consider δ -step forecasting on the sparse movement speed data. Algorithm 2 provides an efficient routine for estimating $\{\mathbf{W}, \mathbf{X}, \mathbf{A}\}$. We first perform seasonal differencing on temporal factors $\{\mathbf{x}_t\}_{t \in [T]}$ as $\dot{\mathbf{x}}_t^m = \mathbf{x}_t - \mathbf{x}_{t-m}$ for all $t \in [m+1, T]$, and then forecast $\dot{\mathbf{x}}_{T+1}^m, \dots, \dot{\mathbf{x}}_{T+\delta}^m$ through the seasonal differenced VAR with the time horizon $\delta \in \mathbb{Z}^+$. Finally, we generate the forecasts of movement speed values $\hat{\mathbf{y}}_{T+1}, \dots, \hat{\mathbf{y}}_{T+\delta}$ sequentially by

$$\hat{\mathbf{y}}_{T+1} = \mathbf{W}^\top \hat{\mathbf{x}}_{T+1}, \quad \hat{\mathbf{y}}_{T+2} = \mathbf{W}^\top \hat{\mathbf{x}}_{T+2}, \quad \dots, \quad \hat{\mathbf{y}}_{T+\delta} = \mathbf{W}^\top \hat{\mathbf{x}}_{T+\delta}. \quad (32)$$

When performing rolling forecasting on sparse movement speed data, we denote the incremental data at the κ -th rolling forecasting with the time horizon δ by $\mathbf{Y}_\kappa \in \mathbb{R}^{N \times (T + (\kappa-1)\delta)}$. NoTMF can process the entire data at once, but it is impractical to accommodate data arriving sequentially. Thus, we take into account the dictionary learning with

a fixed spatial factor matrix (also examined by [Gultekin and Paisley \(2018\)](#)). Such a collection of spatial factors enables the system to transform the incremental movement data to lower-dimensional temporal factors. At each rolling time, we first learn the temporal factor matrix through the conjugate gradient method. Then we update the coefficient matrix accordingly to capture time-evolving patterns. Algorithm 3 shows the implementation of rolling forecasting on the sparse movement speed data.

Algorithm 3 NoTMF Rolling Forecasting on Sparse Movement Speed Data

Input: Data matrix $\mathbf{Y} \in \mathbb{R}^{N \times T}$ with the observed index set Ω , order $d \in \mathbb{Z}^+$ and season $m \in \mathbb{Z}^+$ of VAR, rank $R \in \mathbb{Z}^+$, K rolling times, and forecasting time horizon $\delta \in \mathbb{Z}^+$.

Output: Movement speed forecasts $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times \delta \cdot K}$.

- 1: Train NoTMF on data matrix \mathbf{Y} as shown in Algorithm 2 and return the result:
 $\mathbf{W}, \mathbf{X}, \mathbf{A} = \text{NoTMF}(\mathbf{Y}, \Omega, d, R, \gamma, \rho, m)$.
 - 2: Generate the forecasts $\hat{\mathbf{y}}_{T+1}, \dots, \hat{\mathbf{y}}_{T+\delta}$ by (32) and stack them as the first δ column vectors of $\hat{\mathbf{Y}}$.
 - 3: **for** $\kappa = 2$ to K **do**
 - 4: Input the incremental movement speed data $\mathbf{Y}_\kappa \in \mathbb{R}^{N \times (T + (\kappa-1) \cdot \delta)}$ in which the last δ column vectors are the newly arriving data.
 - 5: Generate temporal operator matrices $\{\Psi_k\}_{k \in [0, d]}$.
 - 6: Compute \mathbf{X}_κ from data \mathbf{Y}_κ with conjugate gradient.
 - 7: Update the coefficient matrix \mathbf{A} .
 - 8: Generate the forecasts $\hat{\mathbf{y}}_{T+(\kappa-1) \cdot \delta + 1}, \dots, \hat{\mathbf{y}}_{T+(\kappa-1) \cdot \delta + \delta}$ and stack them to $\hat{\mathbf{Y}}$.
 - 9: **end for**
-

4. Experiments on Uber Movement Data

Within the context of this study, we conduct an empirical investigation for the NoTMF model through forecasting sparse movement speeds in urban road networks. Specifically, we evaluate NoTMF based on two large-scale movement datasets: the NYC Uber movement speed dataset and the Seattle Uber movement speed dataset. Both datasets are characterized by hourly movement speed values, derived from the trajectories of ridesharing vehicles across various road segments. But if one road segment at a certain hour does not cover at least five unique trips/traces, then the corresponding road-level traffic speed would be

ignored. In light of this, both datasets involve a large amount of missing/unobserved values, thereby accentuating the datasets' inherent sparsity.

4.1. Data Analysis and Experiment Setup

The datasets originate from the Uber Movement project, which aims to equip cities with data and analytical tools to navigate and address urban transportation challenges. The project's movement speed data, indicative of average speeds across specified urban road segments, represent the high-dimensional and sparse nature of traffic data encountered in real-world settings. Figure 3 shows two cases of Uber movement speed data in NYC and Seattle, USA, respectively. The high-dimensional and sparse data inevitably result in difficulties in analyzing traffic states or supporting data-driven city planning and decision-making.

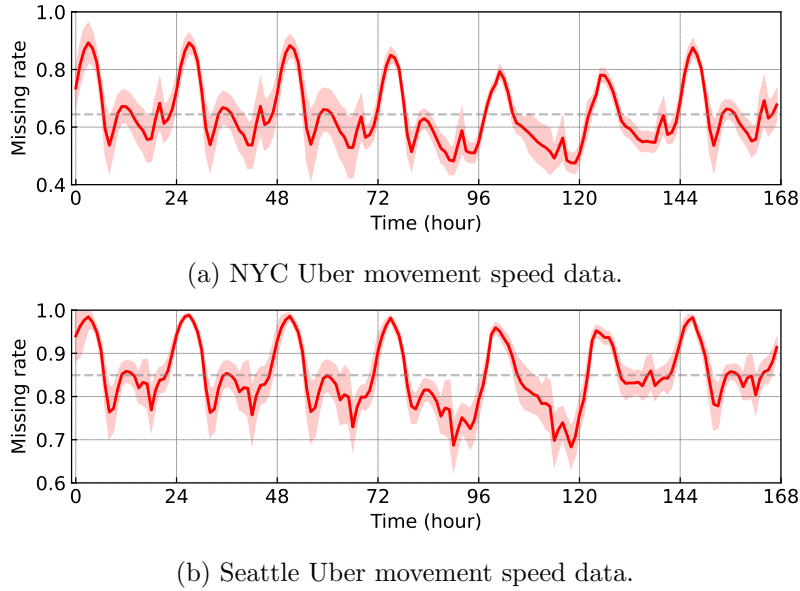


Figure 3 The missing rates of Uber movement speed data aggregated per week over the whole year of 2019. The red curve shows the average missing rates over all 52 weeks. The red area shows the standard deviation of missing rates in each hour over 52 weeks. The 168 time steps refer to 168 hours of Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, and Monday. (a) The dataset has 98,210 road segments, and the overall missing rate is 64.43%. (b) The dataset has 63,490 road segments, and the overall missing rate is 84.95%.

For our experiments, we choose the data collected over the first ten weeks of 2019 from 98,210 road segments in NYC, comprising a dataset matrix size of 98210×1680 . Due to the insufficient sampling of ridesharing vehicles in urban road networks, the dataset contains

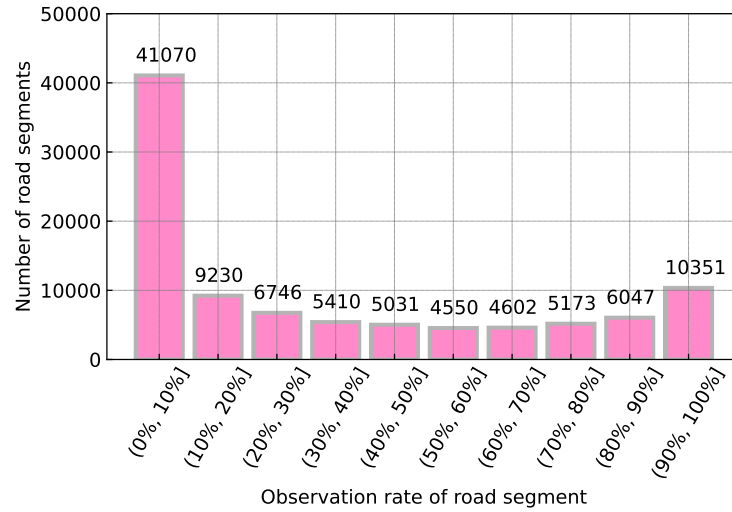


Figure 4 Histogram of observation rate of road segment in the NYC Uber movement speed dataset. Only a small fraction of road segments have an observation rate greater than 50%, i.e., $30723/98210 \approx 31\%$. For the observation rates greater than 20% and 80%, there are about 49% and 17% of road segments, respectively.

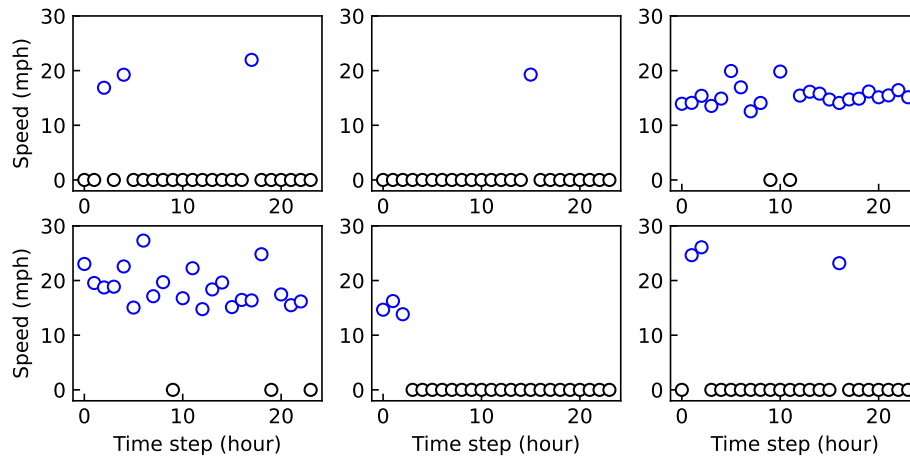


Figure 5 Movement speed of 6 road segments of January 1, 2019 (24 hours) in NYC. Blue points indicate the observed speed from the movement dataset, while black points indicate missing values (set to 0).

66.56% missing values. It can be seen that the time-evolving missing rates have periodicity patterns. At midnight, the missing rate can even reach $\sim 90\%$. This is due to limited ridesharing vehicles on the road network during nights. At some specific hours, only a small fraction of road segments have speed observations. Similarly, the Seattle Uber movement speed dataset covers the hourly movement speed data from 63,490 road segments during the first ten weeks of 2019. The data matrix is of size 63490×1680 and contains 87.35% missing values, showing that the sparsity issue in this dataset is more challenging to handle

than the NYC Uber movement speed dataset. Such sparsity underscores the critical need for sophisticated forecasting models capable of navigating these data gaps effectively.

Figure 4 shows the complementary histogram of the percentage of observed values in the NYC dataset. As can be seen, a large fraction of road segments only have very limited observations. This is especially interesting, as it shows that half of the road segments have less than 20% movement speed observations. Only 17% road segments have more than 80% movement speed observations. We can also see examples of incomplete movement speed observations from Figure 5.

In our experiment, we evaluate the model with 8-week data (from January 1st to February 25th) as the training set, one-week data (February 26th to March 4th) for validation, and one-week data (March 5th to March 11th) as the test set. We use rolling forecasting with time horizons $\delta = 1, 2, 3, 6$, corresponding to δ -hour-ahead forecasting. To confirm the importance of seasonal differencing, we evaluate the proposed NoTMF model and baseline models in the TMF framework. We consider the following TMF models in the literature:

- **TRMF** (Yu et al. 2016): TRMF achieves temporal modeling on latent temporal factors by applying a univariate autoregressive process.
- **BTMF** (Chen and Sun 2022): BTMF is a fully Bayesian TMF model with Gaussian assumption, which is solved by using the Markov chain Monte Carlo (MCMC) algorithm. This model has been empirically demonstrated to be state-of-the-art against some baseline models (e.g., matrix/tensor factorization) for both time series imputation and forecasting on sparse data.
- **BTRMF** (Chen and Sun 2022): Bayesian TRMF is a fully Bayesian treatment for the TRMF model with Gaussian assumption.
- **CTNNM** (Liu and Zhang 2022, Liu 2022): Circulant tensor nuclear norm minimization that is implemented by the fast Fourier transform.

4.2. Forecasting Performance

We evaluate the task of traffic time series forecasting in the presence of missing data on the NYC and Seattle Uber movement speed datasets. The preliminary experiment evaluates the NoTMF model with different ranks $R = 5, 10, 15, 20, 25, 30$. To set the hyperparameters $\{\gamma, \rho\}$, we first prescribe a collection of hyperparameters $\gamma \in \{10^2, 10^1, 10^0, 10^{-1}, 10^{-2}\}$ and $\rho \in \{10\gamma, 5\gamma, \gamma, 5 \times 10^{-1}\gamma, 10^{-1}\gamma\}$ and evaluate the NoTMF with these settings on the training and validation sets. Then, we find the best hyperparameter pair for testing the

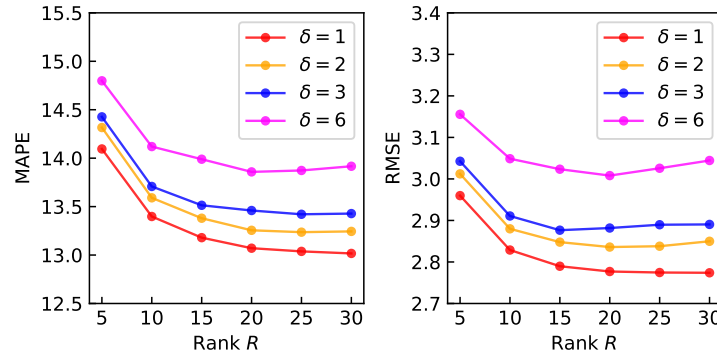


Figure 6 Performance of NoTMF with season-168 differencing (i.e., $m = 168$) and order $d = 6$ on the NYC dataset. The validated parameters are $\gamma = 1$ and $\rho = 5$.

model. In this case, the hyperparameters of NoTMF are properly given by $\gamma = 1$ and $\rho = 5$. As shown in Figure 6, NoTMF with a larger rank essentially provides higher accuracy but such improvement becomes marginal when $R > 15$. In the following experiments, we choose $R = 10$ for a good balance between performance and computational cost.

Table 2 shows the forecasting performance of NoTMF and baseline models on the test set from the NYC dataset. We summarize the following findings from the results:

- With the increase in forecasting time horizons, the forecasting errors of all models increase. For each time horizon, as the order increases, the forecasting performance of NoTMF and TRMF is improved.
- CTNNM, by its definition, is capable of global time series trend modeling, therefore the forecasting performance with the increase of time horizons does not have a significant accuracy reduction. The overall performance of CTNNM is as competitive as BTMF.
- The NoTMF models with different differencing operations demonstrate a significant improvement over TRMF in terms of forecasting accuracy. In contrast to the univariate autoregressive process, there is a clear benefit from temporal modeling with a multivariate VAR process.
- On this dataset, we have two choices for setting the season. One is $m = 24$, corresponding to the daily differencing, while another is $m = 168$, corresponding to the weekly differencing. For both settings, NoTMF can achieve competitively accurate forecasts.

Figure 9 shows the predicted time series and the partially observed speed values of some road segments. It demonstrates that the forecasts produced by NoTMF are consistent with the temporal patterns underlying partially observed time series. Thus, the NoTMF model

Table 2 Forecasting performance (in MAPE/RMSE) on the test set of the NYC movement speed dataset. The rolling forecasting tasks include different time horizons, i.e., $\delta = 1, 2, 3, 6$. We consider the rank as $R = 10$. The best (γ, ρ) is found on the validation set to be $(1, 5)$ for NoTMF and TRMF. Since CTNNM does not have the order d , we only present the forecasting results along the first row at each time horizon. Note that the best test results are highlighted in bold fonts.

δ	d	NoTMF ($m = 24$)	NoTMF ($m = 168$)	TRMF	BTMF	BTRMF	CTNNM
1	1	13.63/2.88	13.53/2.86	14.50/3.12	14.94/3.13	15.93/3.33	15.79/3.12
	2	13.47/ 2.84	13.41/2.84	14.14/3.05	15.70/3.41	15.90/3.35	-/-
	3	13.46/2.84	13.39/2.83	13.87/2.96	15.80/3.34	16.08/3.43	-/-
	6	13.41/ 2.83	13.39/2.83	14.00/2.98	15.45/3.27	16.26/3.48	-/-
2	1	13.91/2.96	13.76/2.94	15.85/3.43	15.33/3.21	16.85/3.56	15.91/3.14
	2	13.77/2.92	13.63/2.89	15.04/3.31	15.87/3.32	17.27/3.71	-/-
	3	13.72/2.91	13.61/2.89	15.25/3.36	15.69/3.33	17.24/3.74	-/-
	6	13.59/ 2.87	13.57/2.88	14.92/3.24	15.91/3.39	18.18/3.97	-/-
3	1	14.30/3.05	14.06/3.02	17.52/3.83	15.86/3.32	18.61/3.91	16.02/3.16
	2	14.01/2.98	13.84/2.94	17.32/4.00	16.30/3.40	18.90/4.10	-/-
	3	13.95/2.97	13.79/2.93	16.91/3.71	16.56/3.49	18.68/4.05	-/-
	6	13.78/ 2.92	13.73/2.92	16.72/3.65	15.49/3.27	20.45/4.66	-/-
6	1	14.61/3.11	14.67/3.20	21.20/4.70	15.99/3.32	22.40/4.69	15.96/3.15
	2	14.30/3.03	14.33/3.09	20.87/5.01	16.04/3.33	23.56/5.63	-/-
	3	14.26/3.03	14.28/3.09	20.08/4.65	15.67/3.28	24.27/5.72	-/-
	6	14.06/2.97	14.16/3.06	20.40/4.35	16.38/3.50	26.34/6.60	-/-

can extract implicit temporal patterns (see Figure 10 for instance) from partially observed data and in the meanwhile perform forecasting.

On the Seattle dataset, we perform forecasting and compare the proposed NoTMF with some baseline models as shown in Table 3. Of these results, it seems that VAR in the NoTMF and BTMF models helps produce reliable forecasting performance with the increase of forecasting time horizons, while the univariate autoregression in the TRMF and BTRMF models fails to maintain the performance of the same level as the increase of forecasting time horizons. Notably, NoTMF consistently outperforms other models as shown

Table 3 Forecasting performance (MAPE/RMSE) on the test set of the Seattle movement speed dataset. The rolling forecasting tasks include different time horizons, i.e., $\delta = 1, 2, 3, 6$. We consider the rank as $R = 10$. The best (γ, ρ) is found on the validation set to be $(1, 5)$ for NoTMF and TRMF. Since CTNNM does not have the order d , we only present the forecasting results along the first row at each time horizon. Note that the best test results are highlighted in bold fonts.

δ	d	NoTMF ($m = 24$)	NoTMF ($m = 168$)	TRMF	BTMF	BTRMF	CTNNM
1	1	10.45/3.32	10.26/3.22	11.58/3.79	12.23/3.89	12.52/4.01	12.74/3.83
	2	10.53/3.34	10.29/3.23	10.92/3.51	12.95/4.18	13.16/4.31	-/-
	3	10.42/3.30	10.30/3.22	10.86/3.47	12.96/4.22	13.89/4.64	-/-
	6	10.50/3.32	10.21/3.21	10.99/3.51	12.91/4.18	13.90/4.67	-/-
2	1	10.90/3.55	10.32/3.25	12.07/4.02	12.74/4.06	13.31/4.32	12.97/3.90
	2	10.90/3.52	10.31/3.24	12.59/4.24	13.68/4.45	13.44/4.43	-/-
	3	10.81/3.49	10.31/3.24	12.01/3.96	13.55/4.46	13.66/4.56	-/-
	6	10.57/3.38	10.25/3.23	12.18/3.98	13.56/4.42	14.67/4.92	-/-
3	1	11.27/3.71	10.41/3.29	13.47/4.62	13.16/4.15	14.01/4.52	13.36/3.97
	2	11.26/3.71	10.30/3.27	14.48/5.19	13.63/4.37	14.39/4.76	-/-
	3	11.11/3.62	10.35/3.28	14.04/4.83	13.76/4.42	14.67/4.84	-/-
	6	10.96/3.55	10.30/3.26	13.32/4.51	13.28/4.29	15.64/5.31	-/-
6	1	11.88/3.97	10.63/3.43	15.59/5.32	13.63/4.30	16.39/5.28	13.82/4.04
	2	11.58/3.83	10.55/3.40	18.66/7.20	13.27/4.19	16.77/5.58	-/-
	3	11.54/3.81	10.57/3.39	17.94/6.32	13.88/4.36	17.35/5.70	-/-
	6	11.27/3.70	10.53/3.35	15.12/5.24	13.30/4.24	16.63/5.62	-/-

in Table 3. We set NoTMF with both daily differencing (i.e., $m = 24$) and weekly differencing (i.e., $m = 168$), and it shows that the weekly differencing is superior to the daily differencing. Therefore, proper seasonal differencing in NoTMF is important for improving the forecasting performance.

Figure 7 and 8 show the speed distribution of the ground truth data versus the forecasts achieved by NoTMF on NYC and Seattle datasets, respectively. For each missing rate range, we group the speed observations of the road segments whose missing rate of the test set is in that range. For road segments with relatively lower missing rates, e.g., $(0, 10\%]$,

the histograms present two peaks. For example on the Seattle dataset (see Figure 8(a-b)), one peak is around the speed of 20 mph, while another is around the speed of 60 mph. This implies that the road segments with relatively complete speed observations exhibit both lower speeds of congested traffic (possibly during peak hours) and higher speeds of free-flow traffic (possibly during off-peak hours), revealing the bi-modal traffic states. Of these results, NoTMF can accurately forecast both congested and free-flow traffic speeds. As a whole, we can summarize from Figure 7 and 8 that the forecasts produced by NoTMF are accurate when compared to the ground truth data.

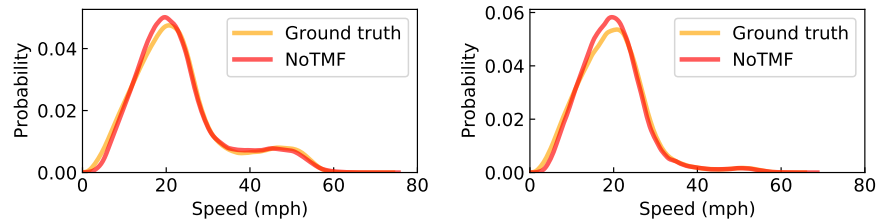
4.3. Nonstationarity Analysis

Nonstationarity is an important characteristic in real-world traffic time series data. Figure 10 illustrates some temporal factors of NoTMF with the following setting: $d = 1$, $R = 10$, and $m = 24$. As can be seen, temporal factors #3, #5, #6, #7, #8, and #9 show clear seasonality and periodicity. The long-term season is associated with the week, i.e., $7 \times 24 = 168$ time steps (hours), while the short-term season is associated with the day, i.e., 24 time steps (hours). For other temporal factors, they also show clear trends with weak seasonality and periodicity. Therefore, making use of seasonal differencing in nonstationary traffic state data can benefit the forecasting performance (as noted in Table 2 and 3). Our results further demonstrate the effectiveness of seasonal differencing in characterizing temporal process in the factor matrix \mathbf{X} .

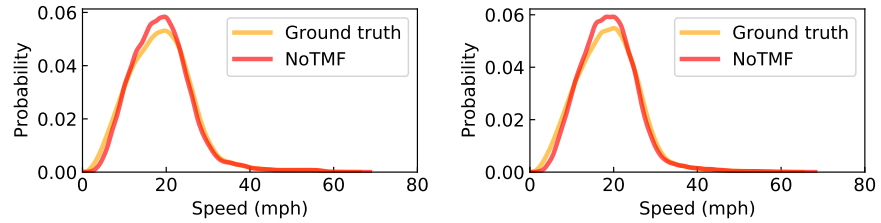
In particular, we visualize the coefficient matrix of NoTMF with the rank $R = 10$ and the order $d = 3$ in Figure 11. In these heatmaps, each is of size 10×10 and the diagonal entries represent the auto-correlations of each time series (i.e., temporal factor). In contrast to the NoTMF model without differencing operation, it demonstrates that the coefficient matrices of NoTMF with seasonal differencing show weak correlations. This implies the importance of seasonal differencing for stationarizing the time series and eliminating the correlations.

5. Concluding Remarks

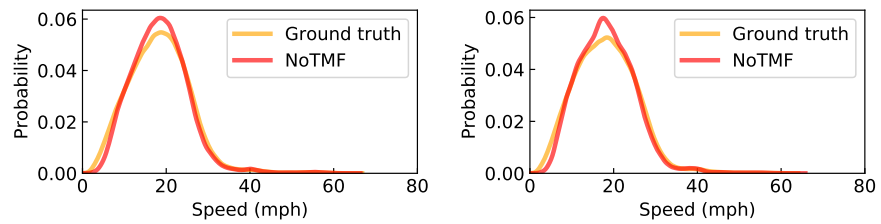
In this study, we propose a NoTMF model, specifically designed to forecast the complex dynamics of high-dimensional, sparse, and nonstationary movement speed data collected from Uber ridesharing vehicles. Distinguishing itself from traditional TMF models, NoTMF innovatively incorporates differencing operations on latent temporal factors, enhancing its



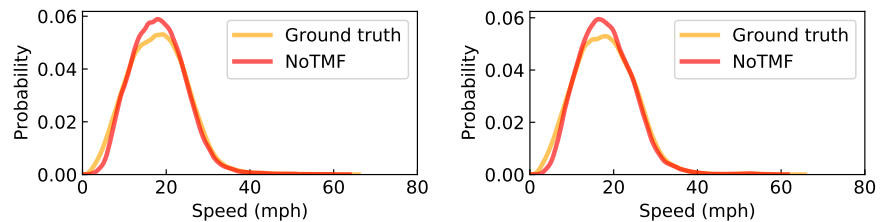
(a) Missing rate between 0 and 10%; (b) Missing rate between 10% and 20%; 7,833 road segments.



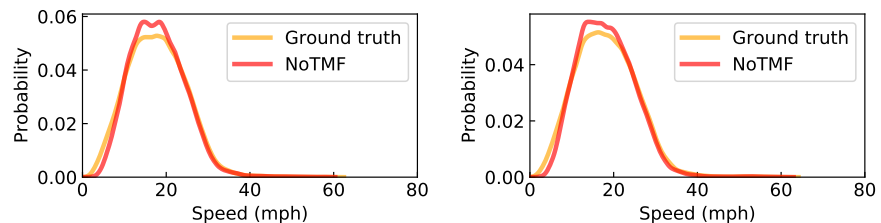
(c) Missing rate between 20% and 30%; 6,229 road segments. (d) Missing rate between 30% and 40%; 5,373 road segments.



(e) Missing rate between 40% and 50%; 5,608 road segments. (f) Missing rate between 50% and 60%; 4,981 road segments.

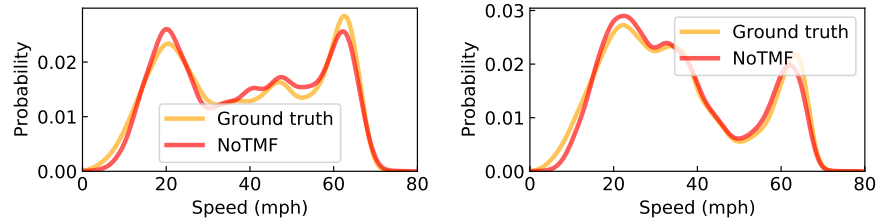


(g) Missing rate between 60% and 70%; 5,833 road segments. (h) Missing rate between 70% and 80%; 6,633 road segments.

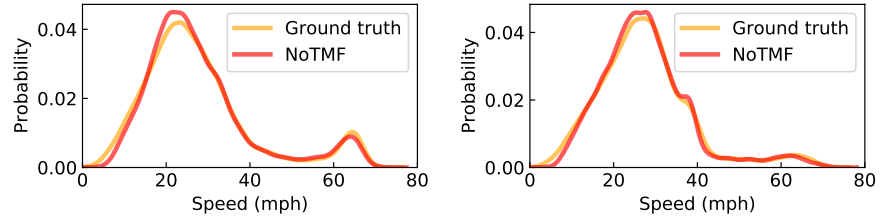


(i) Missing rate between 80% and 90%; 9,804 road segments. (j) Missing rate between 90% and 100%; 33,547 road segments.

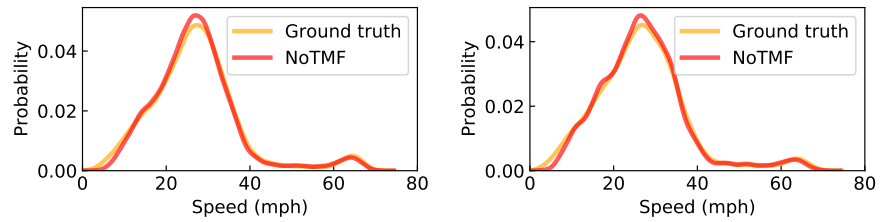
Figure 7 The histogram of ground truth data and forecasts achieved by NoTMF with $\delta = 6$ and $d = 6$ in the test set of the NYC dataset.



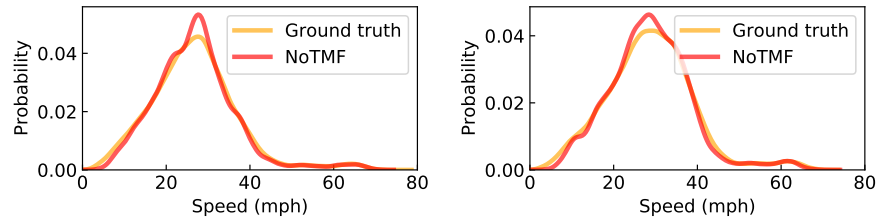
(a) Missing rate between 0 and 10%; (b) Missing rate between 10% and 20%; 1,557 road segments.



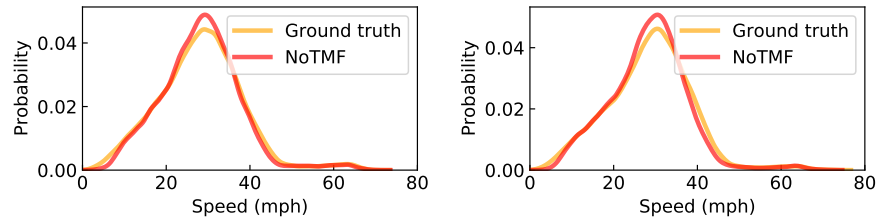
(c) Missing rate between 20% and 30%; 1,476 road segments. (d) Missing rate between 30% and 40%; 1,738 road segments.



(e) Missing rate between 40% and 50%; 1,849 road segments. (f) Missing rate between 50% and 60%; 1,649 road segments.



(g) Missing rate between 60% and 70%; 2,052 road segments. (h) Missing rate between 70% and 80%; 3,225 road segments.



(i) Missing rate between 80% and 90%; 6,392 road segments. (j) Missing rate between 90% and 100%; 42,574 road segments.

Figure 8 The histogram of ground truth data and forecasts achieved by NoTMF with $\delta = 6$ and $d = 6$ in the test set of the Seattle dataset.

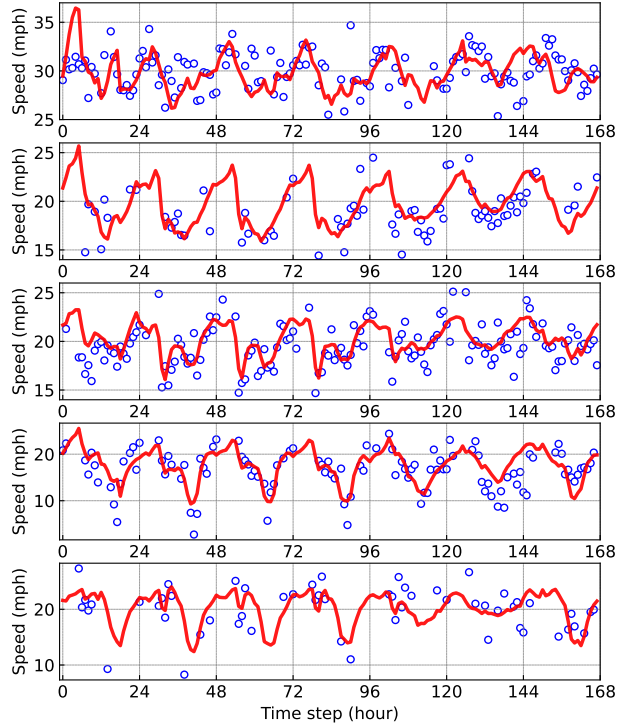


Figure 9 Five examples from the test set (corresponding to five road segments) for showing the forecasting results of the NoTMF model with time horizon $\delta = 6$. The red curves indicate the forecasts in the testing week, while the blue scatters indicate the ground truth speed data.

capability to model nonstationary time series data effectively. The reformulation of the temporal loss function through a sequence of matrix operators is a key innovation, enabling the application of the conjugate gradient method for solving the optimization problem. This method provides an efficient and scalable approach for approximating the closed-form least squares solution to the factor matrix from a generalized Sylvester equation, thus facilitating robust modeling and forecasting of large-scale, high-dimensional urban traffic data. Our comprehensive numerical analysis, conducted on two extensive Uber movement speed datasets, demonstrates NoTMF's superior forecasting performance compared to several baseline models.

Moreover, a comparative analysis with existing TMF models underscores NoTMF's ability to address nonstationary phenomena in time series data, particularly those exhibiting distinct seasonality and trends. The integration of seasonal differencing within the NoTMF framework is pivotal for elaborating the underlying seasonality and periodicity in urban road network speed data. Specifically, setting the seasonality parameter $m = 1$ transforms the temporal loss function to a first-order differenced VAR process, illustrating NoTMF's

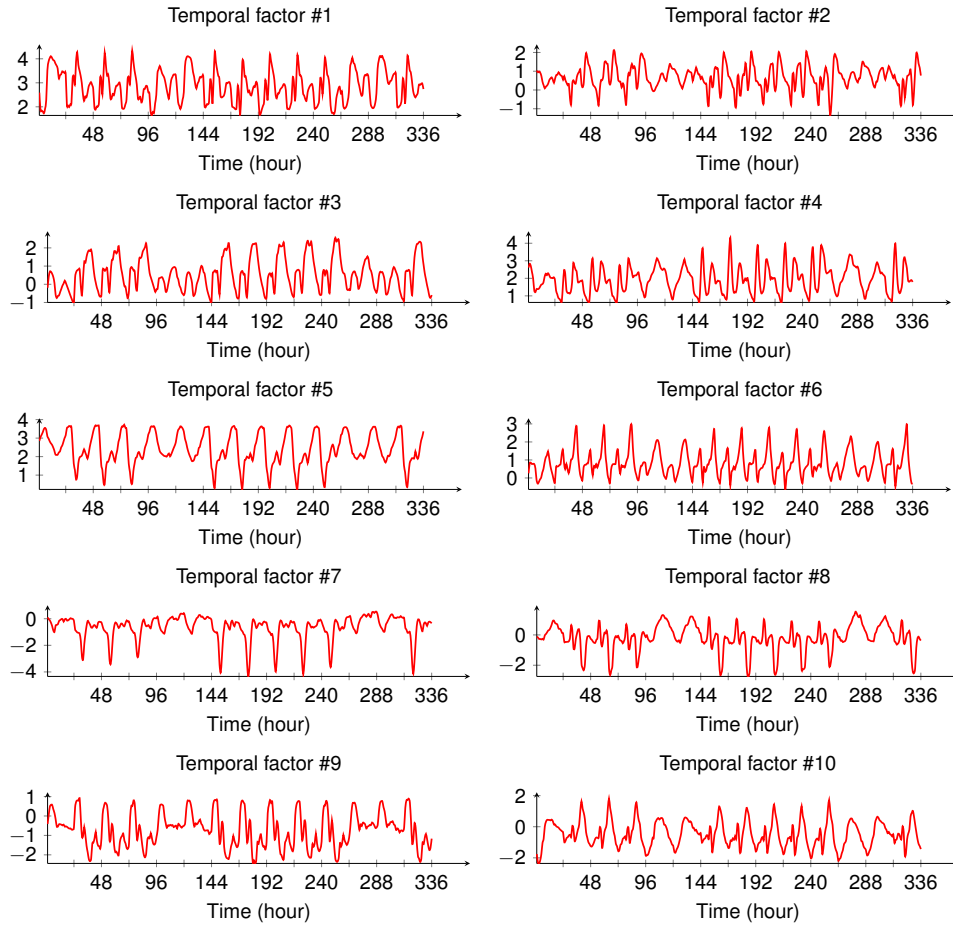


Figure 10 Temporal factors of NoTMF model ($R = 10$) on NYC movement speed data. The order and season of NoTMF are set as $d = 1$ and $m = 24$, respectively. The subfigures only show the temporal factors in the first two weeks.

flexibility in adapting to varying degrees of time series stationarity through the application of both first-order and seasonal differencing techniques.

Looking forward, the potential for integrating nonlinear temporal dependencies into the NoTMF model presents an exciting avenue for research. Such advancements could involve replacing the current linear latent temporal equation (7) with nonlinear functions, leveraging the computational power of deep learning to capture more complex temporal dynamics in urban traffic data (Prince 2023). This evolution of the NoTMF model would not only enhance its forecasting accuracy but also expand its applicability to a broader range of time series datasets characterized by intrinsic seasonal and trend patterns.

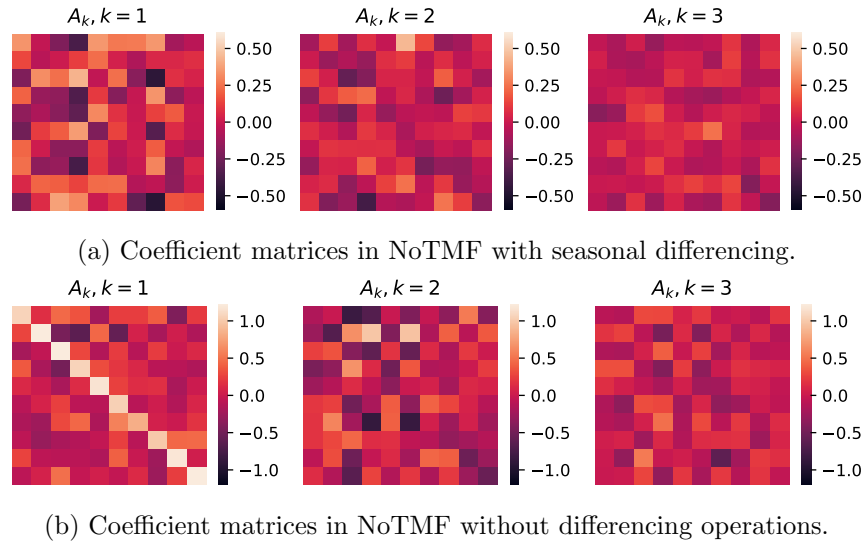


Figure 11 Heatmap of coefficient matrices A_k , $\forall k \in [d]$ in NoTMF. Note that we set the rank as $R = 10$ and the order as $d = 3$ for both models.

References

- Basu S, Li X, Michailidis G (2019) Low rank and structured modeling of high-dimensional vector autoregressions. *IEEE Transactions on Signal Processing* 67(5):1207–1222.
- Basu S, Michailidis G (2015) Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics* 43(4):1535–1567.
- Carriero A, Kapetanios G, Marcellino M (2016) Structural analysis with multivariate autoregressive index models. *Journal of Econometrics* 192(2):332–348.
- Che Z, Purushotham S, Cho K, Sontag D, Liu Y (2018) Recurrent neural networks for multivariate time series with missing values. *Scientific reports* 8(1):1–12.
- Chen X, Sun L (2022) Bayesian temporal factorization for multidimensional time series prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44(9):4659–4673.
- 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(4):904–918.
- Chi Y, Lu YM, Chen Y (2019) Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing* 67(20):5239–5269.
- Dunlavy DM, Kolda TG, Acar E (2011) Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 5(2):1–27.
- Forni M, Hallin M, Lippi M, Reichlin L (2000) The generalized dynamic-factor model: Identification and estimation. *Review of Economics and statistics* 82(4):540–554.
- Golub GH, Van Loan CF (2013) *Matrix Computations* (The Johns Hopkins University Press), fourth edition.

- Guerrero-Ibáñez J, Zeadally S, Contreras-Castillo J (2018) Sensor technologies for intelligent transportation systems. Sensors 18(4):1212.
- Gultekin S, Paisley J (2018) Online forecasting matrix factorization. IEEE Transactions on Signal Processing 67(5):1223–1236.
- Han F, Lu H, Liu H (2015) A direct estimation of high dimensional stationary vector autoregressions. Journal of Machine Learning Research .
- Hangos KM, Bokor J, Szederkényi G (2006) Analysis and control of nonlinear process systems (Springer Science & Business Media).
- Jain NK, Saini R, Mittal P (2019) A review on traffic monitoring system techniques. Soft computing: Theories and applications: Proceedings of SoCTA 2017 569–577.
- Janecek A, Valerio D, Hummel KA, Ricciato F, Hlavacs H (2015) The cellular network as a sensor: From mobile phone data to real-time road traffic monitoring. IEEE transactions on intelligent transportation systems 16(5):2551–2572.
- Kawabata K, Bhatia S, Liu R, Wadhwa M, Hooi B (2021) Ssmf: shifting seasonal matrix factorization. Advances in Neural Information Processing Systems 34:3863–3873.
- Koop G, Korobilis D, Pettenuzzo D (2019) Bayesian compressed vector autoregressions. Journal of Econometrics 210(1):135–154.
- Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37.
- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. nature 521(7553):436–444.
- Liu G (2022) Time series forecasting via learning convolutionally low-rank models. IEEE Transactions on Information Theory 68(5):3362–3380.
- Liu G, Zhang W (2022) Recovery of future data via convolution nuclear norm minimization. IEEE Transactions on Information Theory 69(1):650–665.
- Lütkepohl H (2013) Introduction to multiple time series analysis (Springer Science & Business Media).
- Prince SJ (2023) Understanding Deep Learning (MIT press).
- Rao N, Yu HF, Ravikumar P, Dhillon IS (2015) Collaborative filtering with graph information: Consistency and scalable methods. NIPS, volume 2, 7 (Citeseer).
- Stock JH, Watson MW (2016) Dynamic factor models, factor-augmented vector autoregressions, and structural vector autoregressions in macroeconomics. Handbook of macroeconomics, volume 2, 415–525 (Elsevier).
- Takeuchi K, Kashima H, Ueda N (2017) Autoregressive tensor factorization for spatio-temporal predictions. 2017 IEEE International Conference on Data Mining (ICDM), 1105–1110 (IEEE).

- Treiber M, Kesting A (2013) Traffic flow dynamics. Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg 983–1000.
- Velu R, Reinsel GC (1998) Multivariate reduced-rank regression: theory and applications, volume 136 (Springer Science & Business Media).
- Velu RP, Reinsel GC, Wichern DW (1986) Reduced rank models for multiple time series. Biometrika 73(1):105–118.
- Verleysen M, François D (2005) The curse of dimensionality in data mining and time series prediction. International work-conference on artificial neural networks, 758–770 (Springer).
- Wang D, Zheng Y, Lian H, Li G (2021) High-dimensional vector autoregressive time series modeling via tensor decomposition. Journal of the American Statistical Association 1–19.
- Xiong L, Chen X, Huang TK, Schneider J, Carbonell JG (2010) Temporal collaborative filtering with bayesian probabilistic tensor factorization. Proceedings of the 2010 SIAM international conference on data mining, 211–222 (SIAM).
- Yu HF, Rao N, Dhillon IS (2016) Temporal regularized matrix factorization for high-dimensional time series prediction. Advances in neural information processing systems 29:847–855.
- Zheng F, Jabari SE, Liu HX, Lin D (2018) Traffic state estimation using stochastic lagrangian dynamics. Transportation Research Part B: Methodological 115:143–165.
- Zheng Y (2015) Trajectory data mining: an overview. ACM Transactions on Intelligent Systems and Technology (TIST) 6(3):1–41.