Dynamic Autoregressive Tensor Factorization for Pattern Discovery of Spatiotemporal Systems

Xinyu Chen, Dingyi Zhuang, HanQin Cai, *Senior Member, IEEE*, Shenhao Wang, and Jinhua Zhao

***Abstract*—Spatiotemporal systems are ubiquitous in a large number of scientific areas, representing underlying knowledge and patterns in the data. Here, a fundamental question usually arises as how to understand and characterize these spatiotemporal systems with a certain data-driven machine learning framework. In this work, we introduce an unsupervised pattern discovery framework, namely, dynamic autoregressive tensor factorization. Our framework is essentially built on the fact that the spatiotemporal systems can be well described by the time-varying autoregression on multivariate or even multidimensional data. In the modeling process, tensor factorization is seamlessly integrated into the time-varying autoregression for discovering spatial and temporal modes/patterns from the spatiotemporal systems in which the spatial factor matrix is assumed to be orthogonal. To evaluate the framework, we apply it to several real-world spatiotemporal datasets, including fluid flow dynamics, international import/export merchandise trade, and urban human mobility. On the international trade dataset with dimensions {country/region, product type, year}, our framework can produce interpretable import/export patterns of countries/regions, while the low-dimensional product patterns are also important for classifying import/export merchandise and understanding systematical differences between import and export. On the ridesharing mobility dataset with dimensions {origin, destination, time}, our framework is helpful for identifying the shift of spatial patterns of urban human mobility that changed between 2019 and 2022. Empirical experiments demonstrate that our framework can discover interpretable and meaningful patterns from the spatiotemporal systems that are both time-varying and multidimensional.**

***Index Terms*—Spatiotemporal data, multidimensional data, dynamical systems, pattern discovery, unsupervised learning, time-varying autoregression, tensor factorization.**

I. Introduction

DISCOVERING underlying spatial and temporal patterns, e.g., geospatial patterns and time-varying system behaviors, from spatiotemporal systems is of great significance for understanding data systems, performing predictions, and guiding decision-making processes, as nowadays, large amounts of spatiotemporal data are readily available in several scientific areas such as fluid flow dynamics [1], [2], climate [3], weather [4], mobility [5], traffic flow [6], and neuroscience [7], to name but a few. In the past decades, the advent of data-driven machine learning methods has prompted our understanding of complicated data in these scientific areas [8]. In the field of dynamical systems, substantial progress has been made in characterizing the spatial and temporal modes with dynamic mode decomposition (DMD [1], [2], [3]—reduced-rank vector autoregression on time series)—in which the modes are usually referred to as the data patterns from the low frequency to the high frequency. However, real-world systems are usually time-varying, e.g., sea surface temperature system [9], [10], as a result, exploiting spatial and temporal patterns from these systems is very different from what DMD has done on the time-invariant systems.

To characterize the time-varying systems, the classical time series autoregression can be reformulated within the DMD framework and demonstrate broad applications to fluid flow [5], sea surface temperature [5], [9], [10], neuroscience signals [10], climate [5], and human mobility [5]. For example, the multiresolution DMD takes recursive decomposition on the state of the system, allowing one to characterize nonlinear dynamical systems [9]. The online DMD provides a windowed method for computing DMD in real time and updating the approximation of a system's dynamics as time evolves [11], making it well-suited to time-varying systems. In contrast, the time-varying autoregression model in [10] integrates tensor factorization into a sequence of autoregressions over separated time windows. However, the window setting in both aforementioned models often makes the temporal dynamics overly smooth and prevents the description from fully time-varying. To address this issue, one recent time-varying autoregression model in [5] assumes fully time-varying coefficients and uses tensor factorization to capture both spatial patterns and temporal dynamics. Although tensor factorization can discover spatial and temporal patterns from the time-varying system, the mathematical or physical principles (e.g., orthogonality in physics-informed DMD [12]) at least on the spatial modes/patterns are overlooked in [5], [10].

On the multidimensional data (e.g., international relation data [13], [14] and human mobility [5]), it poses great methodological challenges for formulating the system within the DMD framework. In this work, we start from the assumptions of time-varying systems and the multidimensionality of spatiotemporal data. To discover the underlying data patterns, we reformulate

Received 29 May 2024; revised 6 May 2025; accepted 1 June 2025. Date of publication 4 June 2025; date of current version 5 September 2025. The work of HanQin Cai was partially supported by NSF under Grant DMS 2304489. This research was supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Vehicle Technology Program under Grant DE-EE0009211 and Grant DEEE0011186. Recommended for acceptance by W. Liu. *(Corresponding author: Jinhua Zhao.)*

Xinyu Chen, Dingyi Zhuang, and Jinhua Zhao are with the Department of Urban Studies and Planning, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: chenxy346@gmail.com; dingyi@mit.edu; jinhua@mit.edu).

HanQin Cai is with the Department of Statistics and Data Science, Department of Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: hqcai@ucf.edu).

Shenhao Wang is with the Department of Urban and Regional Planning, University of Florida, Gainesville, FL 32611-5701 USA (e-mail: shenhaowang@ufl.edu).

Digital Object Identifier 10.1109/TPAMI.2025.3576719

the time-varying autoregression with tensor factorization and build up the dynamic autoregressive tensor factorization (DATF) framework for complicated spatiotemporal systems. Overall, the contribution of this work is two-fold:

- We propose a DATF framework that utilizes tensor factorization formulas to capture interpretable and meaningful spatial/temporal modes from the time-varying spatiotemporal systems. On the one hand, we formulate the orthogonal variable estimation in the optimization as an orthogonal Procrustes problem, thus leading to the closed-form solution with the orthogonal projection. On the other hand, the proposed framework can characterize the time-varying system behavior of both vector- and matrix-variate dynamical systems, referring to the systems in the form of matrices and tensors, respectively. The proposal formula on the matrix-variate spatiotemporal systems provides insight into the appropriateness of use, given that the usefulness and generalizability of our framework are of great significance to real-world complicated data.
- We empirically demonstrate how the spatiotemporal systems—in the form of matrices and tensors—can be analyzed with the proposed framework. The experiments show spatial and temporal patterns discovered by the proposed framework on several real-world datasets, including multiresolution fluid flow, international import/export trade, and urban human mobility. The interpretable patterns (e.g., dominant and secondary patterns) are meaningful for understanding complicated spatiotemporal systems from the low frequency to the high frequency.

The remainder of this paper is constructed as follows. Section II summarizes the related literature, while Section III introduces the basic notations, tensor factorization, and time-varying autoregression. Sections IV and V present the DATF framework on the multivariate and multidimensional systems, respectively. In Section VI, we evaluate the proposed framework on several real-world spatiotemporal datasets. Finally, we conclude this study in Section VII.

II. Related Work

In this section, we connect some existing methods such as matrix/tensor factorization, autoregression, and DMD with the proposed DATF framework. Although the proposed framework takes a time-varying autoregression formula and uses tensor factorization to represent the spatial/temporal modes, it differs from the prior studies and demonstrates capabilities for discovering dynamic patterns from complicated spatiotemporal systems. In what follows, we start the discussion from the spatiotemporal system $\{\boldsymbol{y}_t\}_{t\in[T]}$ (of T time steps) in which each vector $\boldsymbol{y}_t \in \mathbb{R}^N$ has N spatial variables.

A. Matrix/Tensor Factorization

Singular value decomposition (SVD) is the cornerstone of many machine learning methods (e.g., principal component analysis [8]) and shows great potential for real-world applications including dimensionality reduction [8], [15], data compression [8], [15], matrix/tensor completion [16], [17], [18], [19], and pattern discovery [20], to name but a few. For finding interpretable patterns from data, non-negative matrix factorization (NNMF) takes a factorization structure such that $\boldsymbol{y}_t = \boldsymbol{W}^\top \boldsymbol{x}_t + \boldsymbol{\epsilon}_t$ ($\boldsymbol{\epsilon}_t$ is the error term) on the non-negative data $\{\boldsymbol{y}_t\}_{t\in[T]}$ with non-negative factors $\boldsymbol{W}$ and $\{\boldsymbol{x}_t\}_{t\in[T]}$ [21], [22]. As NNMF enforces non-negative constraints for the factorized structures, it often represents more interpretable and meaningful patterns than those obtained from SVD due to the non-negativity property of most real-world data.

In the case of multidimensional data, tensor factorization is an efficient dimensionality reduction tool for learning data patterns from tensors [18], [23]. In this area, higher-order SVD, as a fundamental approach, is a natural generalization of SVD to tensors [18]. The existing matrix factorization methods can be properly adapted to the tensor factorization (e.g., tensor principal component analysis [24], [25]) if not mentioning the methodological challenges such as nonconvex optimization, uniqueness of decomposition, and computations. For example, tensor factorization formulas such as Tucker decomposition have been applied to pattern discovery from international relations [13], [14]. Tensor factorization with non-negativity, tensor mode biases, and nonlinearity allows pattern analysis and missing value imputation/prediction [26], [27], [28]. However, the aforementioned matrix/tensor factorization methods are incapable of exploiting spatial/temporal dependencies when learning the data patterns. For instance, the factor matrices obtained by NNMF are invariant to the permutation of rows and columns in the data $\{\boldsymbol{y}_t\}_{t\in[T]}$. This means that these methods are incapable of discovering the dynamic patterns from time series data or spatiotemporal data.

B. Autoregression

In statistics, vector autoregression is a classical method that is widely used to characterize the co-evolution patterns of multiple variables in time series data [29]. In the past decade, integrating low-rank structure into vector autoregression, e.g., DMD, provides a descriptive and predictive framework to reinforce the interpretability of vector autoregression, showing to be well-suited to time-invariant dynamical systems. On the basis of autoregressive time series models, some multidimensional time series models emerge from the matrix autoregression model on matrix-variate time series [30]. In theory, matrix autoregression takes a bilinear structure to correlate time series and involves two coefficient matrices [30], [31], rather different from the vector autoregression in the multivariate analysis [29]. According to the principles of tensor computations, one can generalize the modeling idea of matrix autoregression to tensor-variate time series (i.e., tensor time series), and the counterparts include tensor autoregression with multilinear coefficient matrices [32] and low-rank tensor autoregression [33]. Although bilinear and multilinear expressions in these models preserve the intrinsic representation of multidimensional data beyond the classical time series autoregression models, they are incapable of discovering the time series dynamics and interpretable patterns. Overall, these autoregression expressions can build the temporal correlations of time series data and provide basic frameworks for

characterizing dynamical systems, if not mentioning the nonlinear autoregressive formula for complicated time series [34].

However, one essential assumption behind the vector autoregression is the time-invariant coefficients as the system evolves. This is not always true as the phase transition in time series could be abrupt, for instance in health monitoring with the phase transition from a healthy psychological phenomena to an unhealthy one (or vice versa) [35]. Therefore, time-varying vector autoregressive models are of great significance for capturing the parameter changes in response to interventions [36], [37], [38], e.g., estimating the policy covariates due to the COVID-19 pandemic [39], [40] and monitoring brain activities [41].

C. Dynamic Mode Decomposition

The straightforward way to understand dynamic systems is by reformulating the evolution of a system over time with certain mathematical or physical principles [42], [43], [44], [45]. Typically, DMD takes a classical first-order autoregression formula that is linear time-invariant, namely, $\boldsymbol{y}_{t+1} = \boldsymbol{A}\boldsymbol{y}_t + \boldsymbol{\epsilon}_t$ ($\boldsymbol{\epsilon}_t$ is the error term) on the system $\{\boldsymbol{y}_t\}_{t\in[T]}$ [1], [2], [3], [42], in which the truncated SVD on the coefficient matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ provides insights into how principles could be utilized to capture spatial modes and low-dimensional dynamics of the system (e.g., from the low frequency to the high frequency). To reinforce DMD with some fundamental physical principles, physics-informed DMD formulates principles such as conservation, self-adjointness, localization, causality, and shift-equivariance as Procrustes problems (e.g., orthogonal Procrustes problem) in the optimization, thus demonstrating closed-form solutions and efficient algorithms [12]. However, the time-invariant formulas behind these DMD models are not well-suited to characterize time-varying systems whose properties or behavior change over time [5], [10], [11]. Following the time-varying autoregression methods in [5], [10], the underlying system formula is hard to be properly built in the multidimensional spatiotemporal systems.

III. Preliminaries

A. Notation

In this work, we summarize the basic symbols and notations in Table I. Herein, $\mathbb{R}$ denotes the set of real numbers, while $\mathbb{Z}^+$ refers to the set of positive integers. The definitions of tensor unfolding, Kronecker product, and mode-k product are well elaborated in [18], [46].

B. Tensor Factorization

Tensor factorization is a powerful mathematical tool for analyzing multidimensional data or variables in the form of tensors, already demonstrated broad applications in various scientific areas such as data science, machine learning, and signal processing [15], [18], [23], [46]. Typically, factorization on any given tensor produces a sequence of low-dimensional components (e.g., factor matrices), following certain factorization structures such as Tucker decomposition and CANDECOMP/PARAFAC (CP) decomposition [18]. These models have been verified to be efficient for dimensionality reduction, missing data imputation, and pattern discovery [23]. But such factorization is invariant to the permutation of fibers and slices, as a result, spatiotemporal systems in the form of tensors cannot be well characterized by the low-dimensional factors. For instance, the tensor factorization on spatiotemporal systems would inevitably lose the sequential dependencies when checking out the interpretability.

TABLE I
Summary of the Basic Notation

Notation	Description
$x \in \mathbb{R}$	Scalar
$\boldsymbol{x} \in \mathbb{R}^n$	Vector of length n
$\boldsymbol{X} \in \mathbb{R}^{m\times n}$	Matrix of size $m \times n$
$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{m\times n\times t}$	Tensor of size $m \times n \times t$
$\boldsymbol{\mathcal{X}}_{(1)} \in \mathbb{R}^{m\times(nt)}$	Mode-1 unfolding matrix of $\boldsymbol{\mathcal{X}}$
$\partial f/\partial \boldsymbol{x}$	Partial derivative of f with respect to $\boldsymbol{x}$
$\partial f/\partial \boldsymbol{X}$	Partial derivative of f with respect to $\boldsymbol{X}$
$\boldsymbol{\epsilon}_t \in \mathbb{R}^N, \forall t$	Gaussian error term with $\mathbb{E}[\boldsymbol{\epsilon}_t \mid \cdot] = \boldsymbol{0}$
$[i]$	Positive integer set $\{1, 2, \ldots, i\}$, $i \in \mathbb{Z}^+$
$\|\cdot\|_2$	ℓ_2-norm of vector
$\|\cdot\|_F$	Frobenius norm of matrix
$\mathrm{tr}(\cdot)$	Trace of square matrix
$\otimes$	Kronecker product
$\times_k, \forall k \in \mathbb{Z}^+$	Mode-k product between tensor and matrix
$R \in \mathbb{Z}^+$	Low Tucker rank, see [18], [46]

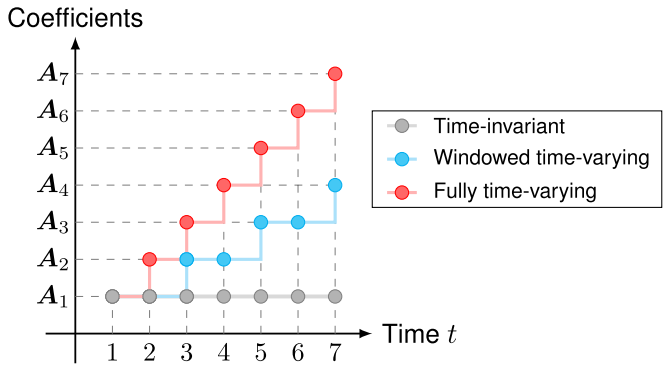
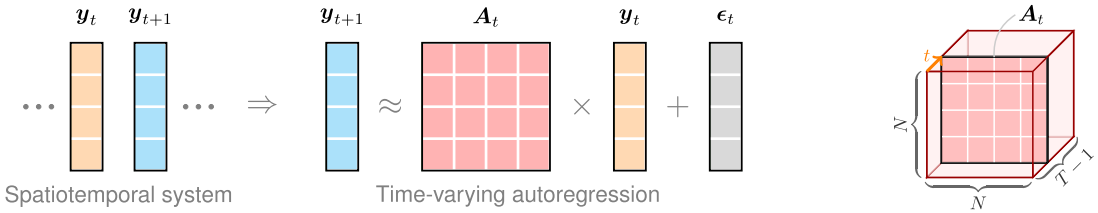


Fig. 1. Diagram for demonstrating the difference among vector autoregressions that are time-invariant (e.g., DMD [1], [2], [3]), windowed time-varying (see e.g., [10]), and fully time-varying (see e.g., [5]) in the literature (within the scope of dynamical system modeling). The autoregression coefficients on the system $\{\boldsymbol{y}_t\}_{t\in[T]}$ are in the form of matrices, e.g., $\boldsymbol{A}_t$.

C. Time-Varying Autoregression

For any multivariate time series data (e.g., spatiotemporal data with spatial and temporal dimensions [5]) $\boldsymbol{Y} \in \mathbb{R}^{N\times T}$ whose time snapshots are $\boldsymbol{y}_t \in \mathbb{R}^N, \forall t \in [T]$ with N variables, the first-order vector autoregression takes a linear time-invariant system. Although the formula and its variants such as DMD are important for characterizing the time-evolving patterns of time series, the time-invariant coefficient matrix is not well-suited to the time-varying system $\{\boldsymbol{y}_t\}_{t\in[T]}$ [3], [5], [10]. As shown in Fig. 1, when handling real-world time-varying systems and discovering dynamic patterns, the fully time-varying autoregression is more flexible for characterizing spatial/temporal pattern transitions than the time-invariant and windowed time-varying methods.



(a) Time-varying autoregression on $\{\boldsymbol{y}_t\}_{t\in[T]}$ (b) Coefficient tensor $\boldsymbol{\mathcal{A}}$

Fig. 2. Illustration of time-varying autoregression on the spatiotemporal system $\{\boldsymbol{y}_t\}_{t\in[T]}$. A sequence of resultant N-by-N coefficient matrices $\{\boldsymbol{A}_t\}_{t\in[T-1]}$ are the frontal slices of the coefficient tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{N\times N\times(T-1)}$.

To characterize the time-varying system behavior, as shown in Fig. 2(a), one can consider the time-varying coefficient matrices $\boldsymbol{A}_t \in \mathbb{R}^{N\times N}$, $\forall t \in [T-1]$ in the vector autoregression [5], [10], namely,

$$\boldsymbol{y}_{t+1} = \boldsymbol{A}_t\boldsymbol{y}_t + \boldsymbol{\epsilon}_t, \tag{1}$$

where the time-varying coefficient matrices $\{\boldsymbol{A}_t\}_{t\in[T-1]}$ can be represented as a tensor of size $N \times N \times (T-1)$, see Fig. 2(b) for an illustration. The aforementioned formula provides a fundamental idea for modeling real-world time-varying systems, but as can be seen, the overparameterization issue arises in the meanwhile because $N^2(T-1)$ parameters in $\boldsymbol{\mathcal{A}}$ would exceed NT observations in $\boldsymbol{Y}$ in most cases.

IV. Time-Varying Autoregression on Spatiotemporal Matrix

In this section, we introduce a time-varying autoregression method for discovering spatial and temporal patterns from the data in the form of a matrix, i.e., the system $\{\boldsymbol{y}_t\}_{t\in[T]}$. The coefficients in the autoregression are represented as a third-order tensor and in this case, tensor factorization allows one to capture interpretable spatial and temporal modes/patterns of spatiotemporal systems.

A. Model Description

In the time-varying autoregression, tensor factorization with a low Tucker rank R can be used to compress the coefficient tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{N\times N\times(T-1)}$ and discover the underlying spatial and temporal patterns from the system. In this case, the number of parameters would be reduced from $N^2(T-1)$ to $(R^2+2N+T-1)R$. Formally, the tensor factorization on $\boldsymbol{\mathcal{A}}$ is given by

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{W} \times_2 \boldsymbol{V} \times_3 \boldsymbol{X}, \tag{2}$$

where $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R\times R\times R}$ is the core tensor. In this factorization formula, $\boldsymbol{W} \in \mathbb{R}^{N\times R}$, $\boldsymbol{V} \in \mathbb{R}^{N\times R}$, and $\boldsymbol{X} \in \mathbb{R}^{(T-1)\times R}$ are factor matrices that represent the modes or patterns of the spatiotemporal systems. Specifically, on the spatiotemporal data, the factor matrix $\boldsymbol{W}$ refers to the spatial modes (i.e., the columns of $\boldsymbol{W}$) of number R. In contrast, the factor matrix $\boldsymbol{X}$ refers to the temporal modes (i.e., the columns of $\boldsymbol{X}$).

According to the property of the mode-k product in tensor computations [18], [46], the formula on the tensor $\boldsymbol{\mathcal{A}}$ can be rewritten on the frontal slices of $\boldsymbol{\mathcal{A}}$, i.e., coefficient matrices $\{\boldsymbol{A}_t\}_{t\in[T-1]}$. On any coefficient matrix $\boldsymbol{A}_t$, (2) is equivalent to the following one:

$$\boldsymbol{A}_t = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{W} \times_2 \boldsymbol{V} \times_3 \boldsymbol{x}_t^\top = \boldsymbol{W}\boldsymbol{G}\left(\boldsymbol{x}_t \otimes \boldsymbol{V}^\top\right), \tag{3}$$

where $\boldsymbol{G} \triangleq \boldsymbol{\mathcal{G}}_{(1)} \in \mathbb{R}^{R\times R^2}$ is the mode-1 unfolding matrix of the core tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R\times R\times R}$. Introducing the Kronecker product can convert the mode-k product into matrix multiplications. As we have tensor factorization on the coefficient tensor, the time-varying autoregression can be reformulated as follows,

$$\begin{cases} \boldsymbol{y}_{t+1} = \boldsymbol{A}_t\boldsymbol{y}_t + \boldsymbol{\epsilon}_t, & \text{(Autoregression)} \\ \boldsymbol{A}_t = \boldsymbol{W}\boldsymbol{G}(\boldsymbol{x}_t \otimes \boldsymbol{V}^\top), & \text{(Tensor factorization)} \end{cases} \tag{4}$$

where $\boldsymbol{y}_t \in \mathbb{R}^N$ and $\boldsymbol{x}_t \in \mathbb{R}^R$, $R < N$ are the high-dimensional time series snapshot and low-dimensional temporal factors, respectively. The aforementioned formula includes a time-varying autoregression on the spatiotemporal system $\{\boldsymbol{y}_t\}_{t\in[T]}$ and a tensor factorization equation on the coefficient matrices $\{\boldsymbol{A}_t\}_{t\in[T-1]}$, showing an observation equation and a latent equation. Notably, the coefficient matrices are replaced by the low-dimensional components. As a consequence, the optimization problem of the DATF framework can be written as follows,

$$\min_{\boldsymbol{G},\boldsymbol{W},\boldsymbol{V},\boldsymbol{X}} \frac{1}{2}\sum_{t=1}^{T-1}\left\|\boldsymbol{y}_{t+1} - \boldsymbol{W}\boldsymbol{G}(\boldsymbol{x}_t \otimes \boldsymbol{V}^\top)\boldsymbol{y}_t\right\|_2^2 \tag{5}$$

where the objective function is the accumulated errors of autoregression over all $t \in [T-1]$. On the spatiotemporal system $\{\boldsymbol{y}_t\}_{t\in[T]}$, we assume the orthogonal factor matrix $\boldsymbol{W}$ as the constraint in the optimization problem for discovering interpretable spatial modes, namely, $\boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{I}_R$. In this DATF framework, we aim to capture both spatial and temporal modes from spatiotemporal systems. Since the time-varying factors $\{\boldsymbol{x}_t\}_{t\in[T-1]}$ would evolve over time t, we do not impose an orthogonality assumption.

Recall that the alternating minimization (e.g., alternating least squares [46]) is a classical optimization scheme for solving nonconvex matrix/tensor factorization problems, we consider developing a solution algorithm with alternating minimization. Let $f(\boldsymbol{G},\boldsymbol{W},\boldsymbol{V},\boldsymbol{X})$ (also appearing as f in the following) be the objective function, the alternating minimization scheme is

summarized as follows,

$$\begin{cases} \boldsymbol{G} := \arg\min_{\boldsymbol{G}} \ f(\boldsymbol{G}, \boldsymbol{W}, \boldsymbol{V}, \boldsymbol{X}), \\ \boldsymbol{W} := \arg\min_{\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_R} f(\boldsymbol{G}, \boldsymbol{W}, \boldsymbol{V}, \boldsymbol{X}), \\ \boldsymbol{V} := \arg\min_{\boldsymbol{V}} \ f(\boldsymbol{G}, \boldsymbol{W}, \boldsymbol{V}, \boldsymbol{X}), \\ \boldsymbol{X} := \arg\min_{\boldsymbol{X}} \ f(\boldsymbol{G}, \boldsymbol{W}, \boldsymbol{V}, \boldsymbol{X}), \end{cases} \tag{6}$$

in which these variables can be updated iteratively. In these subproblems, we need to find the closed-form or approximated solutions to the variables. Note that the variable $\boldsymbol{G}$ has a least squares solution [5], which is given by

$$\boldsymbol{G} = \boldsymbol{W}^\dagger \left(\sum_{t=1}^{T-1} \boldsymbol{y}_{t+1} \boldsymbol{s}_t^\top \right) \left(\sum_{t=1}^{T-1} \boldsymbol{s}_t \boldsymbol{s}_t^\top \right)^{-1}, \tag{7}$$

where $\cdot^\dagger$ denotes the Moore-Penrose pseudoinverse. For notational convenience, in (7), we introduce

$$\boldsymbol{s}_t \triangleq \left(\boldsymbol{x}_t \otimes \boldsymbol{V}^\top \right) \boldsymbol{y}_t. \tag{8}$$

The subproblem with respect to each $\boldsymbol{x}_t, \forall t \in [T-1]$ is

$$\begin{aligned} \boldsymbol{x}_t :=& \arg\min_{\boldsymbol{x}_t} \ \frac{1}{2} \| \boldsymbol{y}_{t+1} - \boldsymbol{W}\boldsymbol{G}(\boldsymbol{x}_t \otimes \boldsymbol{V}^\top)\boldsymbol{y}_t \|_2^2 \\ =& \arg\min_{\boldsymbol{x}_t} \ \frac{1}{2} \| \boldsymbol{y}_{t+1} - \boldsymbol{W}\boldsymbol{G}(\boldsymbol{I}_R \otimes (\boldsymbol{V}^\top \boldsymbol{y}_t))\boldsymbol{x}_t \|_2^2 \\ =& \arg\min_{\boldsymbol{x}_t} \ \frac{1}{2} \| \boldsymbol{y}_{t+1} - \boldsymbol{H}_t \boldsymbol{x}_t \|_2^2 \\ =& \boldsymbol{H}_t^\dagger \boldsymbol{y}_{t+1}, \end{aligned} \tag{9}$$

where we let $\boldsymbol{H}_t \triangleq \boldsymbol{W}\boldsymbol{G}(\boldsymbol{I}_R \otimes (\boldsymbol{V}^\top \boldsymbol{y}_t))$.

In what follows, we introduce how to estimate the factor matrices $\boldsymbol{W}$ and $\boldsymbol{V}$ in the alternating minimization scheme.

B. Estimating the Variable $\boldsymbol{W}$

In the DATF framework, the orthogonal matrix $\boldsymbol{W}$ is expected to identify the independent spatial modes/patterns of spatiotemporal systems. With respect to the variable $\boldsymbol{W}$, the subproblem can be reformulated as a maximization problem on the matrix trace. Here, we reuse the definition of $\boldsymbol{s}_t$, $\forall t \in [T-1]$ in (8). The optimization is given by

$$\begin{aligned} \boldsymbol{W} :=& \arg\min_{\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_R} \ \frac{1}{2} \sum_{t=1}^{T-1} \| \boldsymbol{y}_{t+1} - \boldsymbol{W}\boldsymbol{G}\boldsymbol{s}_t \|_2^2 \\ =& \arg\min_{\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_R} \ \frac{1}{2} \sum_{t=1}^{T-1} (-2\boldsymbol{y}_{t+1}^\top \boldsymbol{W}\boldsymbol{G}\boldsymbol{s}_t \\ & + \boldsymbol{s}_t^\top \boldsymbol{G}^\top \boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{G}\boldsymbol{s}_t) \\ =& \arg\min_{\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_R} \ -\sum_{t=1}^{T-1} \boldsymbol{y}_{t+1}^\top \boldsymbol{W}\boldsymbol{G}\boldsymbol{s}_t \\ =& \arg\max_{\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_R} \ \sum_{t=1}^{T-1} \boldsymbol{y}_{t+1}^\top \boldsymbol{W}\boldsymbol{G}\boldsymbol{s}_t \\ =& \arg\max_{\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_R} \ \mathrm{tr}\left(\boldsymbol{W} \left(\sum_{t=1}^{T-1} \boldsymbol{y}_{t+1} \boldsymbol{s}_t^\top \boldsymbol{G}^\top \right)^\top \right), \end{aligned} \tag{10}$$

in which the minimization problem can be converted into a standard maximization problem of matrix multiplication with an orthogonal constraint, formally referred to as the orthogonal Procrustes problem [47], [48]. Therefore, we first introduce the following SVD:

$$\sum_{t=1}^{T-1} \boldsymbol{y}_{t+1} \boldsymbol{s}_t^\top \boldsymbol{G}^\top = \boldsymbol{B}_w \boldsymbol{\Sigma}_w \boldsymbol{D}_w^\top, \tag{11}$$

where $\boldsymbol{B}_w \in \mathbb{R}^{N \times R}$ and $\boldsymbol{D}_w \in \mathbb{R}^{R \times R}$ consist of the left and right singular vectors, respectively. The diagonal entries of $\boldsymbol{\Sigma}_w \in \mathbb{R}^{R \times R}$ are singular values. Then according to Theorem 1, the closed-form solution to the variable $\boldsymbol{W}$ is given by

$$\boldsymbol{W} := \boldsymbol{B}_w \boldsymbol{D}_w^\top, \tag{12}$$

in which the singular values in (11) are eliminated. The orthogonal spatial modes are of great significance in the dynamical system modeling [12].

Theorem 1. (Orthogonal Procrustes Rotation [47], [48], [49], [50]): For any matrices $\boldsymbol{F}, \boldsymbol{Q} \in \mathbb{R}^{m \times r}, m \geq r$, let the rank of $\boldsymbol{F}$ be r, the solution to the following optimization problem:

$$\begin{aligned} & \max_{\boldsymbol{F}} \ \mathrm{tr}(\boldsymbol{F}\boldsymbol{Q}^\top) \\ & \text{s.t. } \boldsymbol{F}^\top \boldsymbol{F} = \boldsymbol{I}_r, \end{aligned} \tag{13}$$

is given by

$$\boldsymbol{F} := \boldsymbol{B}\boldsymbol{D}^\top, \tag{14}$$

where the SVD of $\boldsymbol{Q}$ is $\boldsymbol{Q} = \boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{D}^\top$.

C. Estimating the Variable $\boldsymbol{V}$

The subproblem to the variable $\boldsymbol{V}$ is convex and it is possible to find the least squares solution. However, the challenge arises when writing down the partial derivative of f with respect to the variable $\boldsymbol{V}$ as follows,

$$\begin{aligned} \frac{\partial f}{\partial \boldsymbol{V}} =& -\sum_{t=1}^{T-1} \boldsymbol{y}_t \left(\boldsymbol{y}_{t+1}^\top - \boldsymbol{y}_t^\top (\boldsymbol{x}_t^\top \otimes \boldsymbol{V}) \boldsymbol{G}^\top \boldsymbol{W}^\top \right) \\ & \cdot \boldsymbol{W}\boldsymbol{G} \left(\boldsymbol{x}_t \otimes \boldsymbol{I}_R \right), \end{aligned} \tag{15}$$

where $\partial f / \partial \boldsymbol{V} = 0$ leads to a generalized Sylvester equation:

$$\begin{aligned} & \sum_{t=1}^{T-1} \boldsymbol{y}_t \boldsymbol{y}_t^\top (\boldsymbol{x}_t^\top \otimes \boldsymbol{V}) \boldsymbol{G}^\top \boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{G} \left(\boldsymbol{x}_t \otimes \boldsymbol{I}_R \right) \\ & \quad = \sum_{t=1}^{T-1} \boldsymbol{y}_t \boldsymbol{y}_{t+1}^\top \boldsymbol{W}\boldsymbol{G} \left(\boldsymbol{x}_t \otimes \boldsymbol{I}_R \right), \end{aligned} \tag{16}$$

which involves $T-1$ terms on the left- and right-hand sides, respectively. In what follows, we denote by $\theta_v : \mathbb{R}^{N \times R} \to \mathbb{R}^{NR}$ the function for describing the vectorization of the left-hand side and $\psi_v \in \mathbb{R}^{NR}$ the vectorization of the right-hand side of (16). The conjugate gradient method for estimating the variable $\boldsymbol{V}$ is summarized in Algorithm 1. Herein, $\boldsymbol{v}_{\ell+1}$, $\boldsymbol{r}_{\ell+1}$, and $\boldsymbol{d}_{\ell+1}$ are the optimizing variable, the residual vector, and the searching

Algorithm 1: Conjugate Gradient for Estimating $\boldsymbol{V}$.

Input: Data $\boldsymbol{Y} \in \mathbb{R}^{N\times T}$, known variables $\{\boldsymbol{G}, \boldsymbol{W}, \boldsymbol{X}\}$, initialized $\boldsymbol{V}$, and maximum iteration $\ell_{\max}$ (e.g., $\ell_{\max} = 5$).
Initialize $\boldsymbol{v}_0$ by the vectorized $\boldsymbol{V}$.
Compute residual vector $\boldsymbol{r}_0 = \psi_v - \theta_v(\boldsymbol{V})$, and let $\boldsymbol{d}_0 = \boldsymbol{r}_0$.
for $\ell = 0$ **to** $\ell_{\max} - 1$ **do**
  Convert vector $\boldsymbol{d}_\ell$ into matrix $\boldsymbol{D}_\ell$.
  Compute $\alpha_\ell = \frac{\boldsymbol{r}_\ell^\top \boldsymbol{r}_\ell}{\boldsymbol{d}_\ell^\top \theta_v(\boldsymbol{D}_\ell)}$.
  Update $\boldsymbol{v}_{\ell+1} = \boldsymbol{v}_\ell + \alpha_\ell \boldsymbol{d}_\ell$.
  Update $\boldsymbol{r}_{\ell+1} = \boldsymbol{r}_\ell - \alpha_\ell \theta_v(\boldsymbol{D}_\ell)$.
  Compute $\beta_\ell = \frac{\boldsymbol{r}_{\ell+1}^\top \boldsymbol{r}_{\ell+1}}{\boldsymbol{r}_\ell^\top \boldsymbol{r}_\ell}$.
  Update $\boldsymbol{d}_{\ell+1} = \boldsymbol{r}_{\ell+1} + \beta_\ell \boldsymbol{d}_\ell$.
  Convert $\boldsymbol{v}_{\ell+1}$ into matrix $\boldsymbol{V}_{\ell+1}$.
end for
Return $\boldsymbol{V}_{\ell_{\max}}$ as the estimate of $\boldsymbol{V}$.

direction vector, respectively. The implementation in this work differs from [5] via the use of the vectorization operation.

D. Solution Algorithm

As summarized in Algorithm 2, we present an unsupervised learning algorithm as the DATF on the spatiotemporal data $\boldsymbol{Y} \in \mathbb{R}^{N\times T}$ (or the system $\{\boldsymbol{y}_t\}_{t\in[T]}$). The algorithm allows one to automatically extract both spatial and temporal modes/patterns from the data, demonstrating to be meaningful for finding interpretable patterns in the case that the spatiotemporal system is time-varying.

At each step, the minimization over one variable (e.g., $\boldsymbol{G}$) decreases or leaves unchanged the objective function f:

$$f\left(\boldsymbol{G}^{\ell+1}, \boldsymbol{W}^{\ell}, \boldsymbol{V}^{\ell}, \boldsymbol{X}^{\ell}\right) \leq f\left(\boldsymbol{G}^{\ell}, \boldsymbol{W}^{\ell}, \boldsymbol{V}^{\ell}, \boldsymbol{X}^{\ell}\right), \quad (17)$$

which holds for the sequence of decision variables. Consequently, $f^{\ell+1} \leq f^{\ell}$ for all ℓ. Since f is lower bounded, the sequence of objective function values $\{f^{\ell}\}$ is non-increasing and converges to a finite value $f^{\star}$, namely, $\lim_{\ell\to\infty} f^{\ell} \to f^{\star}$. In the meantime, since the objective function f is block multiconvex, namely, being convex with respect to each block of variables. Since the partial derivatives (i.e., gradients) of f with respect to each block are zero, the iterative process of alternating minimization converges to a stationary point.

To analyze the empirical time complexity of DATF in Algorithm 2 (with 50 iterations by default), Fig. 3 shows the running time of DATF on the synthetic data of different spatial dimension N and temporal dimension T. As can be seen, The computational cost of DATF increase almost linearly with the increasing N (or T).

Algorithm 2: DATF on Data $\boldsymbol{Y} \in \mathbb{R}^{N\times T}$.

Input: Data $\boldsymbol{Y} \in \mathbb{R}^{N\times T}$, low Tucker rank $R \leq \min\{N, T-1\}$, and maximum iteration L.
Output: $\{\boldsymbol{G}, \boldsymbol{W}, \boldsymbol{V}, \boldsymbol{X}\}$.
1: Initialize factor matrices $\boldsymbol{W}, \boldsymbol{V}, \boldsymbol{X}$ by the truncated SVD on data samples.
2: **for** $\ell = 0$ to $L - 1$ **do**
3: Update $\boldsymbol{G}$ by (7).
4: Update $\boldsymbol{W}$ by (12).
5: Compute $\boldsymbol{V}$ with conjugate gradient in Algorithm 1.
6: **for** $t = 1$ to $T - 1$ **do**
7: Update $\boldsymbol{x}_t$ by (9).
8: **end for**
9: **end for**

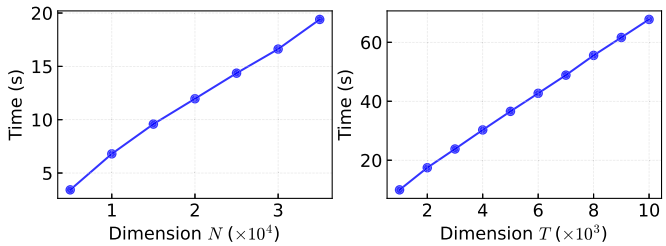


Fig. 3. Empirical time complexity of the DATF model (rank $R = 5$) on the synthetic data of different sizes. In the left panel, we set the data size as $N \in \{5 \times 10^3, 1 \times 10^4, \ldots, 3.5 \times 10^4\}$ and $T = 10$. In the right panel, we set the data size as $N = 10$ and $T \in \{1 \times 10^3, 2 \times 10^3, \ldots, 1 \times 10^4\}$. The reported running time is averaged over 20 random trials.

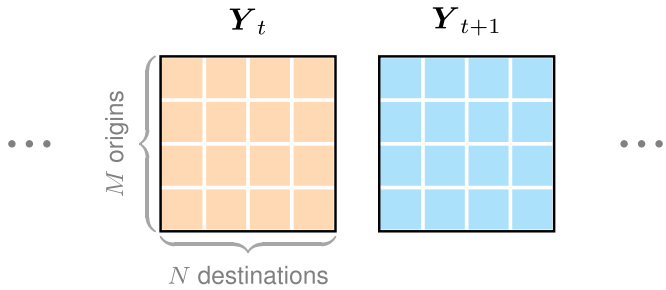


Fig. 4. Illustration of origin-destination matrices for representing human mobility over M origins and N destinations. The entries are traffic volumes or passenger counts.

V. Time-Varying Autoregression on Spatiotemporal Tensor

In this section, we elaborate on the time-varying autoregression on the matrix-variate time series data with time snapshots $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_T \in \mathbb{R}^{M\times N}$ (i.e., M-by-N variables in the spatial context, see an example of human mobility data in Fig. 4). The coefficients in the autoregression are represented as a fourth-order tensor and we introduce a tensor factorization formula to discover the interpretable spatial/temporal modes and patterns.

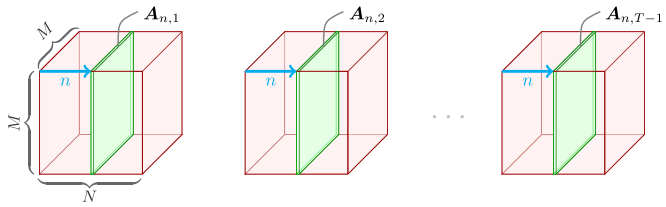


Fig. 5. Illustration of the fourth-order coefficient tensor $\mathcal{A} \in \mathbb{R}^{M\times M\times N\times(T-1)}$ in Eq. (18) in which the slices are the time-varying coefficient matrices $\{\boldsymbol{A}_{n,t}\}_{n\in[N],t\in[T-1]}$ of size $M \times M$.

A. Model Description

Matrix/tensor autoregression is a multi-dimensional generalization of vector autoregression [31], [51], often used to analyze matrix- or tensor-variate time series. On the spatiotemporal system $\{\boldsymbol{Y}_t\}_{t\in[T]}$ (or written in the form of tensor $\mathcal{Y} \in \mathbb{R}^{M\times N\times T}$), we propose a time-varying vector autoregression on the columns of $\boldsymbol{Y}_t \in \mathbb{R}^{M\times N}$ independently. The nth column of $\boldsymbol{Y}_t$ is denoted by $\boldsymbol{y}_{n,t} \in \mathbb{R}^M$, the resultant time-varying vector autoregression is given by

$$\boldsymbol{y}_{n,t+1} = \boldsymbol{A}_{n,t}\boldsymbol{y}_{n,t} + \boldsymbol{\epsilon}_{n,t}, \tag{18}$$

where $\boldsymbol{A}_{n,t} \in \mathbb{R}^{M\times M}$, $\forall n \in [N]$, $\forall t \in [T-1]$ is the coefficient matrix. By definition, the coefficient matrices $\{\boldsymbol{A}_{n,t}\}_{n\in[N],t\in[T-1]}$ are both t-varying (i.e., time-varying) and n-varying. Regarding these coefficient matrices, we let $\boldsymbol{A}_{n,t}$ be the (n,t)th slice of the fourth-order tensor $\mathcal{A} \in \mathbb{R}^{M\times M\times N\times(T-1)}$ as shown in Fig. 5.

In this case, tensor factorization on the coefficient tensor $\mathcal{A}$ produces five components such that

$$\mathcal{A} = \mathcal{G} \times_1 \boldsymbol{W} \times_2 \boldsymbol{U} \times_3 \boldsymbol{V} \times_4 \boldsymbol{X}, \tag{19}$$

where $\boldsymbol{W} \in \mathbb{R}^{M\times R}$, $\boldsymbol{U} \in \mathbb{R}^{M\times R}$, $\boldsymbol{V} \in \mathbb{R}^{N\times R}$, and $\boldsymbol{X} \in \mathbb{R}^{(T-1)\times R}$ are the factor matrices, while $\mathcal{G} \in \mathbb{R}^{R\times R\times R\times R}$ is the fourth-order core tensor in the tensor factorization. Here, we have $(2M + N + T - 1 + R^3)R$ parameters in total for estimation. According to the property of mode-k product in tensor computations [18], [46], we can write down the factorization formula on any coefficient matrix $\boldsymbol{A}_{n,t}$:

$$\begin{aligned}\boldsymbol{A}_{n,t} &= \mathcal{G} \times_1 \boldsymbol{W} \times_2 \boldsymbol{U} \times_3 \boldsymbol{v}_n^\top \times_4 \boldsymbol{x}_t^\top \\ &= \boldsymbol{W}\boldsymbol{G}\left(\boldsymbol{x}_t \otimes \boldsymbol{v}_n \otimes \boldsymbol{U}^\top\right),\end{aligned} \tag{20}$$

where $\boldsymbol{G} \triangleq \mathcal{G}_{(1)} \in \mathbb{R}^{R\times R^3}$ is the mode-1 unfolding of the core tensor $\mathcal{G} \in \mathbb{R}^{R\times R\times R\times R}$. Herein, $\boldsymbol{v}_n \in \mathbb{R}^R$, $\forall n \in [N]$ and $\boldsymbol{x}_t \in \mathbb{R}^R$, $\forall t \in [T-1]$ are the nth and tth rows of the factor matrices $\boldsymbol{V}$ and $\boldsymbol{X}$, respectively.

By introducing tensor factorization on the coefficient tensor and imposing the orthogonal constraint on the spatial factor matrix $\boldsymbol{W}$, the optimization problem of the time-varying autoregression within the DATF framework can be formulated as follows,

$$\begin{aligned}\min_{\boldsymbol{G},\boldsymbol{W},\boldsymbol{U},\boldsymbol{V},\boldsymbol{X}} \ & \frac{1}{2}\sum_{n=1}^{N}\sum_{t=1}^{T-1}\left\|\boldsymbol{y}_{n,t+1} - \boldsymbol{W}\boldsymbol{G}(\boldsymbol{x}_t\otimes\boldsymbol{v}_n\otimes\boldsymbol{U}^\top)\boldsymbol{y}_{n,t}\right\|_2^2 \\ \text{s.t.} \ & \boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{I}_R,\end{aligned} \tag{21}$$

In what follows, we let f be the objective function.

B. Estimating the Variable $\boldsymbol{G}$

The closed-form solution to the subproblem with respect to $\boldsymbol{G}$ is

$$\begin{aligned}\boldsymbol{G} &:= \arg\min_{\boldsymbol{G}} \frac{1}{2}\sum_{n=1}^{N}\sum_{t=1}^{T-1}\left\|\boldsymbol{y}_{n,t+1} - \boldsymbol{W}\boldsymbol{G}\boldsymbol{s}_{n,t}\right\|_2^2 \\ &= \boldsymbol{W}^\dagger\left(\sum_{n=1}^{N}\sum_{t=1}^{T-1}\boldsymbol{y}_{n,t+1}\boldsymbol{s}_{n,t}^\top\right)\left(\sum_{n=1}^{N}\sum_{t=1}^{T-1}\boldsymbol{s}_{n,t}\boldsymbol{s}_{n,t}^\top\right)^{-1},\end{aligned} \tag{22}$$

where we let

$$\boldsymbol{s}_{n,t} \triangleq \left(\boldsymbol{x}_t \otimes \boldsymbol{v}_n \otimes \boldsymbol{U}^\top\right)\boldsymbol{y}_{n,t}. \tag{23}$$

C. Estimating the Variable $\boldsymbol{W}$

With respect to the variable $\boldsymbol{W}$, if we reuse the definition of $\boldsymbol{s}_{n,t}$, $\forall n \in [N], t \in [T-1]$ in (23), then the closed-form solution can be written as follows,

$$\begin{aligned}\boldsymbol{W} &:= \arg\min_{\boldsymbol{W}^\top\boldsymbol{W}=\boldsymbol{I}_R} \frac{1}{2}\sum_{n=1}^{N}\sum_{t=1}^{T-1}\left\|\boldsymbol{y}_{n,t+1} - \boldsymbol{W}\boldsymbol{G}\boldsymbol{s}_{n,t}\right\|_2^2 \\ &= \arg\min_{\boldsymbol{W}^\top\boldsymbol{W}=\boldsymbol{I}_R} \frac{1}{2}\sum_{n=1}^{N}\sum_{t=1}^{T-1}(-2\boldsymbol{y}_{n,t+1}^\top\boldsymbol{W}\boldsymbol{G}\boldsymbol{s}_{n,t} \\ &\quad + \boldsymbol{s}_{n,t}^\top\boldsymbol{G}^\top\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{G}\boldsymbol{s}_{n,t}) \\ &= \arg\max_{\boldsymbol{W}^\top\boldsymbol{W}=\boldsymbol{I}_R} \sum_{n=1}^{N}\sum_{t=1}^{T-1}\boldsymbol{y}_{n,t+1}^\top\boldsymbol{W}\boldsymbol{G}\boldsymbol{s}_{n,t} \\ &= \arg\max_{\boldsymbol{W}^\top\boldsymbol{W}=\boldsymbol{I}_R} \operatorname{tr}\left(\boldsymbol{W}\left(\sum_{n=1}^{N}\sum_{t=1}^{T-1}\boldsymbol{y}_{n,t+1}\boldsymbol{s}_{n,t}^\top\boldsymbol{G}^\top\right)^\top\right) \\ &= \boldsymbol{B}_w\boldsymbol{D}_w,\end{aligned} \tag{24}$$

where

$$\sum_{n=1}^{N}\sum_{t=1}^{T-1}\boldsymbol{y}_{n,t+1}\boldsymbol{s}_{n,t}^\top\boldsymbol{G}^\top = \boldsymbol{B}_w\boldsymbol{\Sigma}_w\boldsymbol{D}_w^\top, \tag{25}$$

is the SVD, referring to Theorem 1.

D. Estimating the Variable $\boldsymbol{U}$

According to the property of Kronecker product, the partial derivative of f with respect to the variable $\boldsymbol{U}$ is given by

$$\begin{aligned}\frac{\partial f}{\partial \boldsymbol{U}} = & -\sum_{n=1}^{N}\sum_{t=1}^{T-1}\boldsymbol{y}_{n,t}\left(\boldsymbol{y}_{n,t+1}^\top - \boldsymbol{y}_{n,t}^\top\left(\boldsymbol{x}_t^\top\otimes\boldsymbol{v}_n^\top\otimes\boldsymbol{U}\right)\right. \\ & \left.\cdot\,\boldsymbol{G}^\top\boldsymbol{W}^\top\right)\boldsymbol{W}\boldsymbol{G}\left(\boldsymbol{x}_t\otimes\boldsymbol{v}_n\otimes\boldsymbol{I}_R\right),\end{aligned} \tag{26}$$

Algorithm 3: Conjugate Gradient for Estimating $\boldsymbol{U}$.

Input: Data $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{M\times N\times T}$, known variables $\{\boldsymbol{G}, \boldsymbol{W}, \boldsymbol{V}, \boldsymbol{X}\}$, initialized $\boldsymbol{U}$, and maximum iteration $\ell_{\max}$ (e.g., $\ell_{\max} = 5$).
Initialize $\boldsymbol{u}_0$ by the vectorized $\boldsymbol{U}$.
Compute residual vector $\boldsymbol{r}_0 = \psi_u - \theta_u(\boldsymbol{U})$, and let $\boldsymbol{d}_0 = \boldsymbol{r}_0$.
for $\ell = 0$ **to** $\ell_{\max} - 1$ **do**
  Convert vector $\boldsymbol{d}_\ell$ into matrix $\boldsymbol{D}_\ell$.
  Compute $\alpha_\ell = \frac{\boldsymbol{r}_\ell^\top \boldsymbol{r}_\ell}{\boldsymbol{d}_\ell^\top \theta_u(\boldsymbol{D}_\ell)}$.
  Update $\boldsymbol{u}_{\ell+1} = \boldsymbol{u}_\ell + \alpha_\ell \boldsymbol{d}_\ell$.
  Update $\boldsymbol{r}_{\ell+1} = \boldsymbol{r}_\ell - \alpha_\ell \theta_v(\boldsymbol{D}_\ell)$.
  Compute $\beta_\ell = \frac{\boldsymbol{r}_{\ell+1}^\top \boldsymbol{r}_{\ell+1}}{\boldsymbol{r}_\ell^\top \boldsymbol{r}_\ell}$.
  Update $\boldsymbol{d}_{\ell+1} = \boldsymbol{r}_{\ell+1} + \beta_\ell \boldsymbol{d}_\ell$.
  Convert $\boldsymbol{u}_{\ell+1}$ into matrix $\boldsymbol{U}_{\ell+1}$.
end for
Return $\boldsymbol{U}_{\ell_{\max}}$ as the estimate of $\boldsymbol{U}$.

Let $\partial f/\partial \boldsymbol{U} = \boldsymbol{0}$, we have a generalized Sylvester equation which can be solved by the conjugate gradient method:

$$\begin{aligned}&\sum_{n=1}^{N}\sum_{t=1}^{T-1} \boldsymbol{y}_{n,t}\boldsymbol{y}_{n,t}^\top \left(\boldsymbol{x}_t^\top \otimes \boldsymbol{v}_n^\top \otimes \boldsymbol{U}\right) \boldsymbol{G}^\top \boldsymbol{W}^\top \\ &\quad \cdot \boldsymbol{W}\boldsymbol{G}\left(\boldsymbol{x}_t \otimes \boldsymbol{v}_n \otimes \boldsymbol{I}_R\right) \\ &= \sum_{n=1}^{N}\sum_{t=1}^{T-1} \boldsymbol{y}_{n,t}\boldsymbol{y}_{n,t+1}^\top \boldsymbol{W}\boldsymbol{G}\left(\boldsymbol{x}_t \otimes \boldsymbol{v}_n \otimes \boldsymbol{I}_R\right). \end{aligned} \tag{27}$$

The routine for estimating $\boldsymbol{U}$ is summarized in Algorithm 3. Here, we denote by $\theta_u : \mathbb{R}^{M\times R} \to \mathbb{R}^{MR}$ the function for describing the vectorization of the left-hand side and $\psi_u \in \mathbb{R}^{MR}$ the vectorization of the right-hand side of (27).

E. Estimating the Variable $\boldsymbol{V}$

The subproblem with respect to each $\boldsymbol{v}_n, \forall n \in [N]$ is

$$\begin{aligned}\boldsymbol{v}_n &:= \arg\min_{\boldsymbol{v}_n} \frac{1}{2}\sum_{t=1}^{T-1} \left\|\boldsymbol{y}_{n,t+1} - \boldsymbol{W}\boldsymbol{G}(\boldsymbol{x}_t \otimes \boldsymbol{v}_n \otimes \boldsymbol{U}^\top)\boldsymbol{y}_{n,t}\right\|_2^2 \\ &= \arg\min_{\boldsymbol{v}_n} \frac{1}{2}\sum_{t=1}^{T-1} \left\|\boldsymbol{y}_{n,t+1} - \boldsymbol{Q}_{n,t}\boldsymbol{v}_n\right\|_2^2 \\ &= \left(\sum_{t=1}^{T-1} \boldsymbol{Q}_{n,t}^\top \boldsymbol{Q}_{n,t}\right)^{-1} \left(\sum_{t=1}^{T-1} \boldsymbol{Q}_{n,t}^\top \boldsymbol{y}_{n,t+1}\right), \end{aligned} \tag{28}$$

where we let

$$\boldsymbol{Q}_{n,t} \triangleq \boldsymbol{W}\boldsymbol{G}\left(\boldsymbol{x}_t \otimes \boldsymbol{I}_R \otimes \left(\boldsymbol{U}^\top \boldsymbol{y}_{n,t}\right)\right), \tag{29}$$

which is obtained from the property of Kronecker product such that

$$\begin{aligned}&(\boldsymbol{x}_t \otimes \boldsymbol{v}_n \otimes \boldsymbol{U}^\top)\boldsymbol{y}_{n,t} \\ &= ((\boldsymbol{x}_t \otimes \boldsymbol{v}_n) \otimes \boldsymbol{U}^\top)\boldsymbol{y}_{n,t} \\ &= \mathrm{vec}(\boldsymbol{U}^\top \boldsymbol{y}_{n,t}(\boldsymbol{x}_t \otimes \boldsymbol{v}_n)^\top) \\ &= \boldsymbol{x}_t \otimes \boldsymbol{v}_n \otimes (\boldsymbol{U}^\top \boldsymbol{y}_{n,t}) \\ &= (\boldsymbol{x}_t \otimes \boldsymbol{I}_R \otimes (\boldsymbol{U}^\top \boldsymbol{y}_{n,t}))\boldsymbol{v}_n. \end{aligned} \tag{30}$$

Algorithm 4: DATF on Data $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{M\times N\times T}$.

Input: Data $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{M\times N\times T}$, low Tucker rank $R \le \min\{\min\{M, N\}, T-1\}$, and maximum iteration L.
Output: $\{\boldsymbol{G}, \boldsymbol{W}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{X}\}$.
1: Initialize factor matrices $\boldsymbol{W}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{X}$ by the truncated SVD on data samples.
2: **for** $\ell = 0$ to $L - 1$ **do**
3: Update $\boldsymbol{G}$ by (22).
4: Update $\boldsymbol{W}$ by (24).
5: Compute $\boldsymbol{U}$ with conjugate gradient in Algorithm 3.
6: **for** $t = 1$ to $T - 1$ **do**
7: Compute $\boldsymbol{v}_n$ by (28).
8: **end for**
9: **for** $t = 1$ to $T - 1$ **do**
10: Compute $\boldsymbol{x}_t$ by (31).
11: **end for**
12: **end for**

F. Estimating the Variable $\boldsymbol{X}$

The subproblem with respect to each $\boldsymbol{x}_t, \forall t \in [T-1]$ is

$$\begin{aligned}\boldsymbol{x}_t &:= \arg\min_{\boldsymbol{x}_t} \frac{1}{2}\sum_{n=1}^{N} \left\|\boldsymbol{y}_{n,t+1} - \boldsymbol{W}\boldsymbol{G}(\boldsymbol{x}_t \otimes \boldsymbol{v}_n \otimes \boldsymbol{U}^\top)\boldsymbol{y}_{n,t}\right\|_2^2 \\ &= \arg\min_{\boldsymbol{x}_t} \frac{1}{2}\sum_{n=1}^{N} \left\|\boldsymbol{y}_{n,t+1} - \boldsymbol{H}_{n,t}\boldsymbol{x}_t\right\|_2^2 \\ &= \left(\sum_{n=1}^{N} \boldsymbol{H}_{n,t}^\top \boldsymbol{H}_{n,t}\right)^{-1} \left(\sum_{n=1}^{N} \boldsymbol{H}_{n,t}^\top \boldsymbol{y}_{n,t+1}\right), \end{aligned} \tag{31}$$

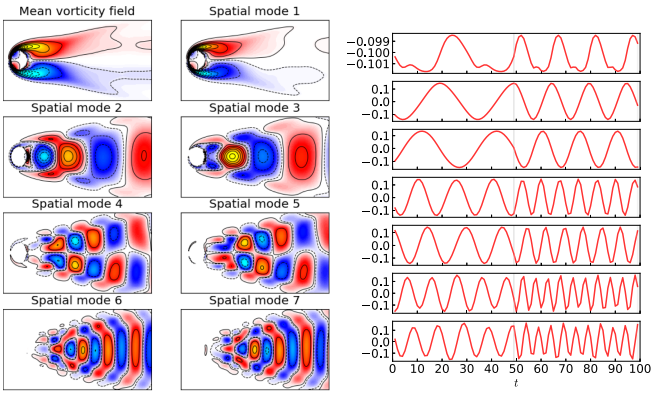
where we let

$$\boldsymbol{H}_{n,t} \triangleq \boldsymbol{W}\boldsymbol{G}(\boldsymbol{I}_R \otimes \boldsymbol{v}_n \otimes (\boldsymbol{U}^\top \boldsymbol{y}_{n,t})), \tag{32}$$

which follows the same property of Kronecker product as mentioned in (30).

G. Solution Algorithm

As mentioned above, the proposed DATF framework handles both matrix and tensor time series, demonstrating a natural generalization from the system $\{\boldsymbol{y}_t\}_{t\in[T]}$ to the system $\{\boldsymbol{Y}_t\}_{t\in[T]}$. Algorithm 4 allows one to capture spatial/temporal modes and patterns from the system $\{\boldsymbol{Y}_t\}_{t\in[T]}$ that involves complicated dimension information. Overall, our framework provides valuable insight into multidimensional spatiotemporal system modeling.



(a) Spatial modes (b) Temporal modes

Fig. 6. Fluid flow and spatial/temporal modes to demonstrate the model capability. (a) Mean vorticity field and spatial modes of the fluid flow. Spatial modes are plotted by seven columns of $\boldsymbol{W}$. Note that the colorbars of all modes are on the same scale. (b) Temporal modes of the fluid flow in $\boldsymbol{X}$. Seven panels correspond to the rank $R = 7$.

VI. Experiments

In this section, we conduct extensive experiments on several real-world spatiotemporal systems, including fluid flow, international trade, and urban human mobility. For performance evaluation, we consider the pattern discovery from fluid flow as a benchmark to identify spatial and temporal modes with a multiresolution setting. Both international trade and urban human mobility demonstrate real-world applications and scenarios for verifying the model capability.

A. Fluid Flow

Fluid flow dataset[1] acts as a classical benchmark for evaluating the existing DMD methods and providing insight into the dynamical system modeling. The dataset has 150 time snapshots and each time snapshot is a matrix of size 199×449, representing 199-by-449 vorticity fields. In this work, we manually generate a multiresolution dataset on the fluid flow dataset with different frequencies as [5]. For numerical evaluation, we describe the multiresolution dataset as a spatiotemporal system $\{\boldsymbol{y}_t\}_{t\in[T]}$ and set the rank as $R = 7$ in Algorithm 2. Fig. 6(a) shows seven dominant spatial modes/patterns of the fluid flow achieved by our model. In these modes, the first mode is consistent with the mean vorticity field and the next modes change from low frequency to high frequency when observing the corresponding temporal modes in Fig. 6(b). In particular, our model can identify the time-varying system behavior (e.g., the changing point at $t = 50$ in Fig. 6(b)) in the multiresolution fluid flow. Notably, these spatial modes are consistent with the results in DMD [2], [3] and the existing time-varying model [5], while the temporal modes can be verified by comparing with the findings in [5].

[1]The cylinder wake dataset is collected from the fluid flow passing a circular cylinder using direct numerical simulations of the Navier-Stokes equations, see http://dmdbook.com.

B. International Trade

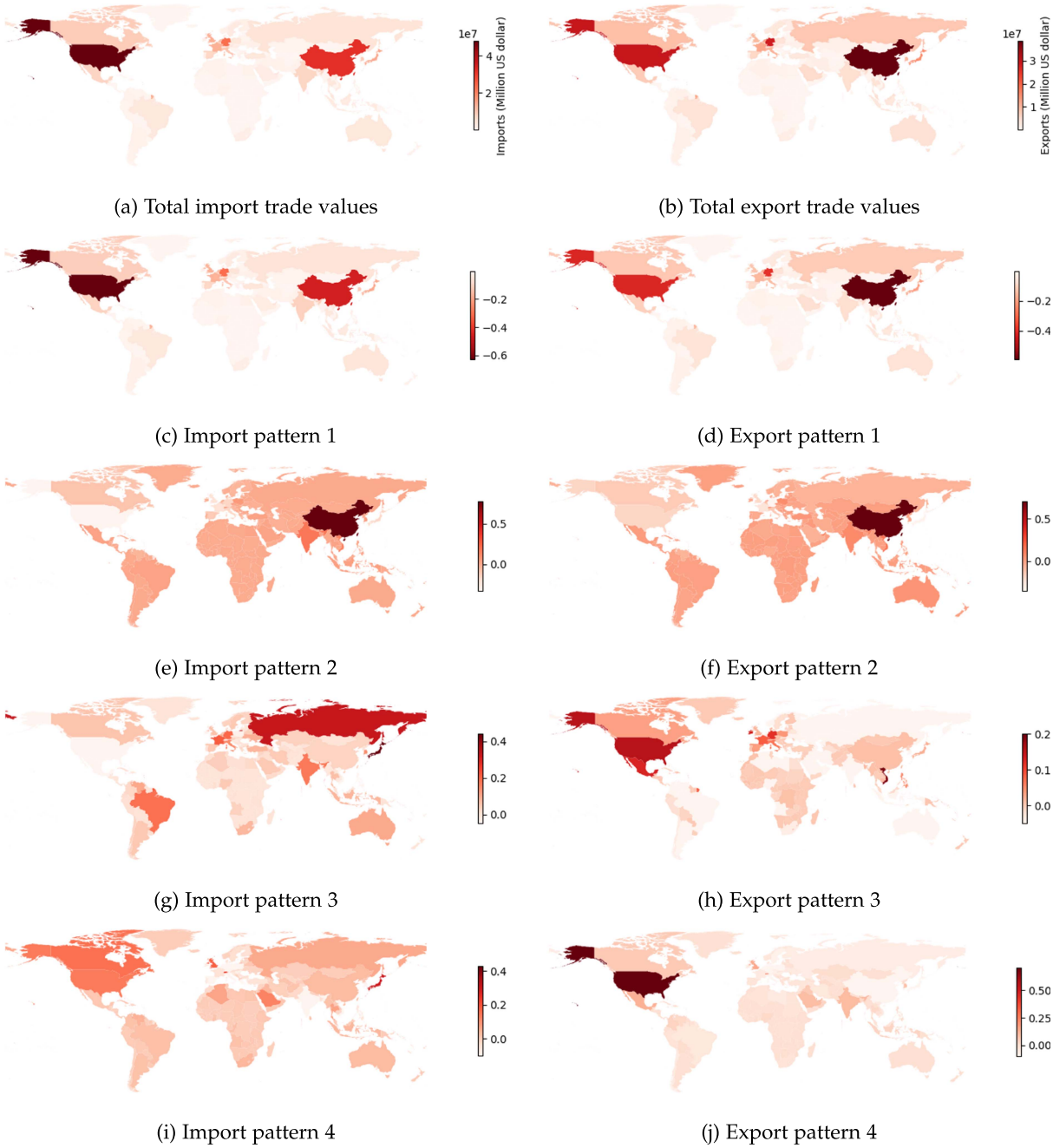
The international trade dataset used in this work is from the WTO Trade Statistics.[2] We consider the merchandise trade values annual dataset—including both imports and exports—during the period from 2000 to 2022 for the following experiments. There are 215 countries/regions—corresponding to the economy dimension—identified by the reporting economy ISO3A code and 17 product types, e.g., agricultural products, fuels and mining products, and manufactures. We summarize the evaluation as two categories: one numerical experiment is on the total merchandise values of imports/exports (i.e., the trade data with dimensions $\{\text{economy}, \text{year}\}$), while another is on the merchandise values of imports/exports with different product types (i.e., the trade data with dimensions $\{\text{economy}, \text{product}, \text{year}\}$). In what follows, we use Algorithms 2 and 4 to evaluate two categories, respectively.

1) On $\{Economy, Year\}$ Imports/Exports: We evaluate Algorithm 2 on the import data and the export data independently in which both data are in the form of matrices of size 215×23, namely, merchandise trade values of 215 countries/regions and 23 years. Fig. 7 shows the spatial patterns of both import and export trade values. As can be seen, Fig. 7(c) and (d) preserve the most important patterns of import and export trade data, respectively, demonstrating to be consistent with the trade statistics of import and export trade values in Fig. 7(a) and (b). Here, the first spatial patterns demonstrate the influential roles that China and the USA played in the international import and export trade. The second spatial patterns as shown in Fig. 7(e) and (f) highlight the import and export trade of China, in which the spatial patterns of import and export trade are consistent. Fig. 7(g) shows the third spatial pattern of the import trade and it reveals similar import between Japan and Russia. Fig. 7(h) reveals similar export patterns among Vietnam, USA, Mexico, and Germany. While the fourth import pattern in Fig. 7(i) highlights the pattern of Japan, Fig. 7(j) highlights the USA. Of these results, the dominant patterns achieved by our model allow one to understand international trade from an unsupervised learning perspective.

2) On $\{Economy, Product, Year\}$ Imports/Exports: We evaluate Algorithm 4 on the import data and the export data independently in which both data are in the form of tensors of size $215 \times 17 \times 23$. Using such tensor representation allows one to uncover the patterns of 17 product types because the coefficients $\{\boldsymbol{A}_{n,t}\}_{n\in[N],t\in[T]}$ in (18) are both time-varying and product-varying for analyzing the international trade data. Fig. 8(a) and (b) show the product patterns, enabling us to perform downstream tasks such as classification. In principle, the import and the export in international trade can be interpreted as "demand" and "supply", respectively. The structural differences between the import and export merchandise for these countries/regions can be distinguished by observing Fig. 8(a) and (b). Specifically, as shown in Fig. 8(a), some similar patterns of these import products can be easily found:

- *(Import group #1)* Agricultural products versus Food.

[2]The original data is available at https://stats.wto.org.



(a) Total import trade values (b) Total export trade values

(c) Import pattern 1 (d) Export pattern 1

(e) Import pattern 2 (f) Export pattern 2

(g) Import pattern 3 (h) Export pattern 3

(i) Import pattern 4 (j) Export pattern 4

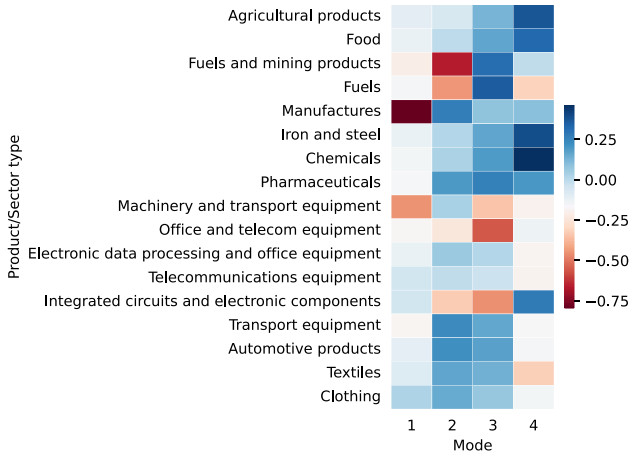
Fig. 7. International import and export trade values and their underlying spatial patterns. (a-b) The total imports/exports trade values of 215 countries/regions over the past 23 years from 2000 to 2022. (c-j) Spatial patterns are revealed by the factor matrix $\boldsymbol{W} \in \mathbb{R}^{215\times 4}$ in which we visualize import and export patterns independently.

- *(Import group #2)* Iron and steel versus Chemicals versus Pharmaceuticals.
- *(Import group #3)* Electronic data processing and office equipment versus Telecommunications equipment.
- *(Import group #4)* Transport equipment versus Automotive products.

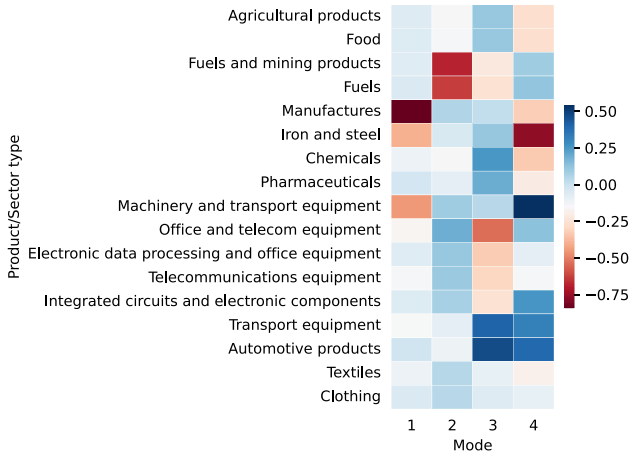
In contrast, Fig. 8(b) shows some similar patterns of the following export products:

- *(Export group #1)* Agricultural products versus Food.
- *(Export group #2)* Fuels and mining products versus Fuels.
- *(Export group #3)* Chemicals versus Pharmaceuticals.
- *(Export group #4)* Office and telecom equipment versus Electronic data processing and office equipment versus Telecommunications equipment.
- *(Export group #5)* Transport equipment versus Automotive products.
- *(Export group #6)* Textiles versus Clothing.

By comparison, one can summarize consistent patterns in the pair (agricultural products, food) between import and export merchandise, while another pair is (transport equipment, automotive products). This demonstrates that some merchandise items are highly correlated and show similar "demand"/"supply"



(a) Import product patterns



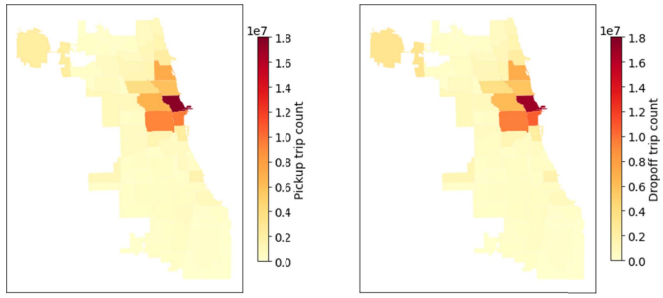
(b) Export product patterns

Fig. 8. Four patterns of 17 products that are revealed by the factor matrix $\boldsymbol{V} \in \mathbb{R}^{17\times 4}$ in the international trade.

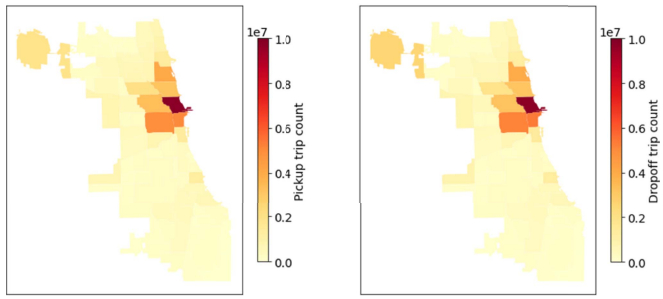
patterns. While the pair (electronic data processing and office equipment, telecommunications equipment) in the import patterns is identified (i.e., import group #3), an additional product "office and telecom equipment" is covered in the export group #4. The pair (chemicals, pharmaceuticals) in the export group #3 is consistent with the import group #2 if eliminating the product "iron and steel". In fact, due to structural differences between import and export, one can also find similar patterns of the export merchandise such as products in the export group #2 (i.e., natural resources) or the export group #6, but both groups are not significant in the imports. Overall, our model can automatically discover product patterns from three-dimensional international trade data and help us understand the structural differences between imports and exports.

C. Urban Human Mobility

In urban systems, human mobility data can be naturally represented by tensors because aggregated trips are usually associated with multiple dimensions such as origin, destination, and start time. For instance, at any time $t \in [T]$, the time snapshot $\boldsymbol{Y}_t \in$

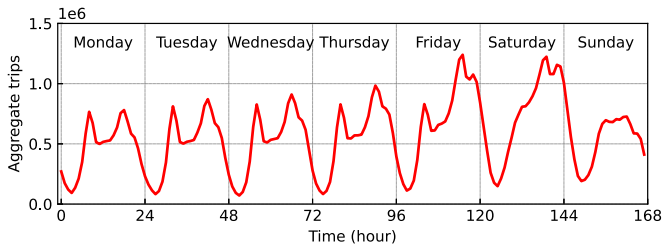


(a) In 2019

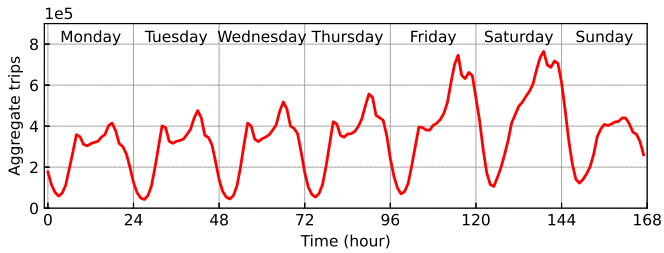


(b) In 2022

Fig. 9. Ridesharing pickup (left panels) and dropoff (right panels) trips in Chicago. There are 96,642,881 trips in the whole year of 2019 and 57,290,954 trips in 2022.



(a) Ridesharing trips in 2019

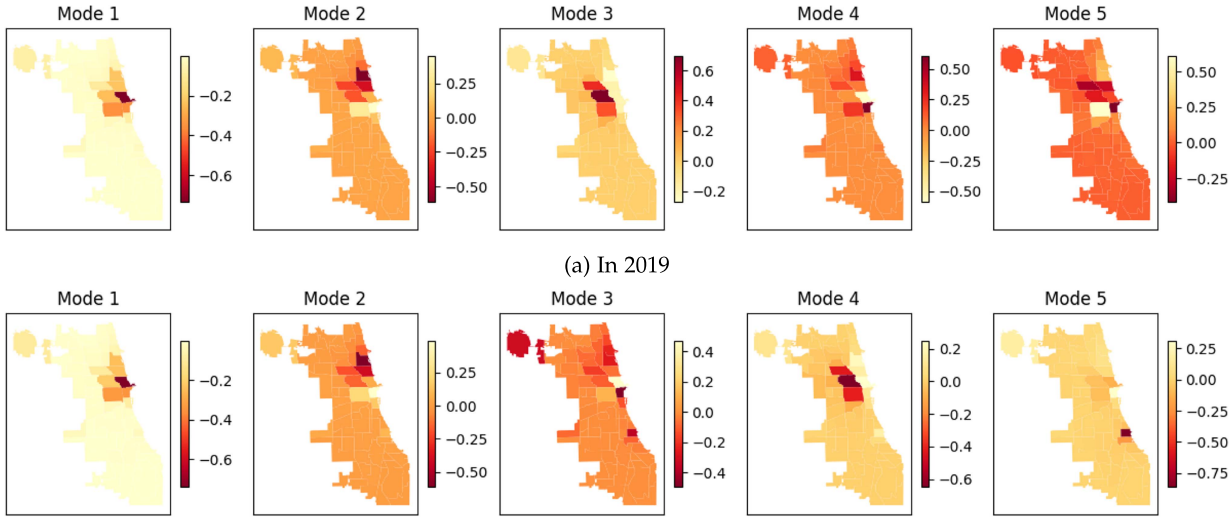


(b) Ridesharing trips in 2022

Fig. 10. Aggregate trips of ridesharing in Chicago.

$\mathbb{R}^{M\times N}$ of human mobility tensor is indeed an origin-destination matrix, showing the traffic volume or passenger flow from M origins to N destinations. To evaluate Algorithm 4, we consider the data of Transportation Network Providers (i.e., ridesharing companies such as Uber and Lyft) in Chicago.[3] As shown in Fig. 9, Chicago has 77 community areas, implying 77 pickup

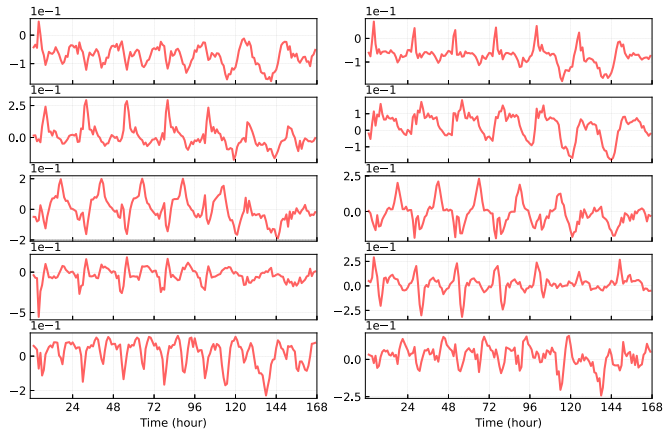
[3]The original data is available at https://data.cityofchicago.org/Transportation/Transportation-Network-Providers-Trips-2018-2022-/m6dm-c72p.



(a) In 2019

(b) In 2022

Fig. 11. Pickup modes/patterns revealed by the factor matrix $\boldsymbol{W} \in \mathbb{R}^{77\times 5}$.



(a) In 2019 (b) In 2022

Fig. 12. Temporal modes/patterns revealed by the factor matrix $\boldsymbol{X} \in \mathbb{R}^{167\times 5}$. The panels of each side from top to bottom correspond to the modes 1-5.

areas and 77 dropoff areas for the ridesharing trips. As a result, these mobility data can be represented as matrix-variate time series $\boldsymbol{Y}_t \in \mathbb{R}^{M\times N}, \forall t \in [T]$ in which $M = N = 77$.

As shown in Fig. 9, the most frequent areas of ridesharing trips are highlighted. There seems to be no significant pattern change by comparing the trips in 2019 with 2022, if not mentioning the remarkable reduction of total trips changing from 2019 to 2022. Fig. 10 presents the aggregated hourly ridesharing trips over 52 weeks in the whole year. The temporal trend of ridesharing trips is roughly consistent between 2019 and 2022. The peaks of ridesharing trips during weekdays (e.g., morning/afternoon peak hours) of 2022 become less remarkable when compared to 2019. In this case, the temporal dimension of mobility data is of length 168, corresponding to 7×24 hours.

Through evaluating Algorithm 4 on the mobility data in 2019 and 2022, we obtain the spatial patterns—from the dominant ones to the secondary ones—of pickup trips as shown in Fig. 11. The mobility data in 2019 and 2022 demonstrate very similar second patterns in which the high-income community areas near Downtown are highlighted. The third pattern in 2019 becomes less important in 2022—referring to the fourth pattern in Fig. 11(b)—because of mobility changes during COVID-19. In contrast, the fourth pattern in 2019 becomes more important in 2022 (see the third pattern in Fig. 11(b)). In fact, such pattern changes and shifts are hard to identify from the aggregated data.

In Fig. 12, the first patterns of ridesharing trips in 2019 and 2022 show the low-frequency trends in which the day-to-day similarity and differences between weekdays and weekends are identified. The second pattern of ridesharing trips in 2019 highlights the sharp peaks—around the morning peak hours—on the weekdays, and such peaks become marginal in the second pattern during 2022. The third patterns demonstrate the peaks for both morning peak hours (i.e., the mode with negative values) and afternoon peak hours (i.e., the mode with positive values), but such peaks are eliminated on the weekends. As shown in Fig. 12, the remaining patterns are with relatively higher frequencies. Overall, the proposal framework allows us to understand the spatial and temporal patterns from the multidimensional mobility data.

VII. Conclusion

In this work, we investigate an important machine learning task—unsupervised spatiotemporal pattern discovery—for understanding system behaviors of various spatiotemporal data, including fluid flow, international trade, and urban human mobility. In particular, we present a DATF framework for spatiotemporal systems such as $\{\boldsymbol{y}_t\}_{t\in[T]}$ (i.e., multivariate time series) and $\{\boldsymbol{Y}_t\}_{t\in[T]}$ (i.e., matrix-variate time series). The underlying patterns of the spatiotemporal systems are characterized by the low-dimensional components of tensor factorization within time-varying autoregressions. On the fluid flow dataset, our framework can produce interpretable spatial patterns and

identify the system behavior with different frequencies. On the international trade dataset, our framework can be properly generalized from the multivariate data (i.e., in the form of matrices) to the multidimensional data (i.e., in the form of tensors). In that case, our framework can capture the most significant import/export trade patterns of the reported countries/regions. It is also possible to use product patterns to perform downstream tasks such as the classification of merchandise items and the variation analysis between import and export trade. On the urban human mobility dataset, our framework is well-suited to the spatial/temporal pattern discovery from multidimensional mobility, revealing the system changes in the long-term range.

Throughout this work, we demonstrate the importance of connecting time-varying autoregression with tensor factorization because: 1) time-invariant autoregression, such as DMD, struggles to capture dynamic patterns of spatiotemporal systems, and 2) purely tensor factorization without temporal correlations cannot preserve temporal dependencies. Although there are many time-varying autoregression methods in the literature, to the best of our knowledge, we advance the model development by generalizing the formula from vector autoregression to matrix- and tensor-variate autoregression. This substantial step—introducing tensor factorization to matrix-variate time-varying autoregressions—makes such methods well-suited to characterize spatiotemporal systems with multiple dimensions. In the future study, it is also remarkable to generalize the current framework and discover dominant patterns (in the low frequency) and anomalies/outliers (in the high frequency) from the international trade data with multiple dimensions as {export, import, product, year}. In terms of model development, the orthogonality constraint in the DATF framework is imposed on the factor matrix $\boldsymbol{W}$ as the orthogonal Procrustes rotation in Theorem 1 provides a closed-form solution to $\boldsymbol{W}$. However, it remains unclear how to efficiently solve for the orthogonal factor matrices across the remaining dimensions in DATF. Therefore, it is meaningful to develop an efficient algorithm and reinforce the model interpretability with orthogonal factor matrices.

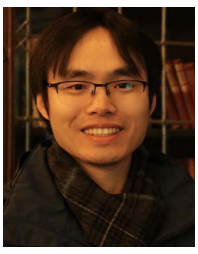
Acknowledgment

The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the United States Government. The Mens, Manus, and Machina (M3S) is an interdisciplinary research group (IRG) of the Singapore MIT Alliance for Research and Technology (SMART) center.

References

[1] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *J. Fluid Mechan.*, vol. 656, pp. 5–28, 2010.
[2] J. H. Tu, "Dynamic mode decomposition: Theory and applications," Ph.D. dissertation, Dept. Mech. Aerosp. Eng., Princeton Univ., Princeton, NJ, 2013.
[3] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. Philadelphia, PA, USA: SIAM, 2016.
[4] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, "Accurate medium-range global weather forecasting with 3D neural networks," *Nature*, vol. 619, no. 7970, pp. 533–538, 2023.
[5] X. Chen, C. Zhang, X. Chen, N. Saunier, and L. Sun, "Discovering dynamic patterns from spatiotemporal data with time-varying low-rank autoregression," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 2, pp. 504–517, Feb. 2024.
[6] M. Treiber and A. Kesting, "Traffic flow dynamics," in *Traffic Flow Dynamics: Data, Models and Simulation*, Berlin, Germany: Springer-Verlag, 2013, pp. 983–1000.
[7] S. Sanei and J. A. Chambers, *EEG Signal Processing and Machine Learning*. Hoboken, NJ, USA: Wiley, 2021.
[8] C. M. Bishop, "Pattern recognition and machine learning," *Springer Google Schola*, vol. 2, pp. 645–678, 2006.
[9] J. N. Kutz, X. Fu, and S. L. Brunton, "Multiresolution dynamic mode decomposition," *SIAM J. Appl. Dynamical Syst.*, vol. 15, no. 2, pp. 713–735, 2016.
[10] K. D. Harris, A. Aravkin, R. Rao, and B. W. Brunton, "Time-varying autoregression with low-rank tensors," *SIAM J. Appl. Dynamical Syst.*, vol. 20, no. 4, pp. 2335–2358, 2021.
[11] H. Zhang, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, "Online dynamic mode decomposition for time-varying systems," *SIAM J. Appl. Dynamical Syst.*, vol. 18, no. 3, pp. 1586–1609, 2019.
[12] P. J. Baddoo, B. Herrmann, B. J. McKeon, J. Nathan Kutz, and S. L. Brunton, "Physics-informed dynamic mode decomposition," in *Proc. Roy. Soc. A*, vol. 479, no. 2271, 2023, Art. no. 20220576.
[13] A. Schein, J. Paisley, D. M. Blei, and H. Wallach, "Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 1045–1054.
[14] A. Schein, M. Zhou, D. Blei, and H. Wallach, "Bayesian poisson tucker decomposition for learning the structure of international relations," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2016, pp. 2810–2819.
[15] J. Wright and Y. Ma, *High-Dimensional Data Analysis With Low-Dimensional Models: Principles, Computation, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2022.
[16] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.
[17] E. Candes and B. Recht, "Exact matrix completion via convex optimization," *Commun. ACM*, vol. 55, no. 6, pp. 111–119, 2012.
[18] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[19] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.
[20] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," in *A Practical Approach to Microarray Data Analysis*, Berlin, Germany: Springer, 2003, pp. 91–109.
[21] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
[22] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 535–541.
[23] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, Jul. 2017.
[24] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 18–39, Jan. 2008.
[25] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 925–938, Apr. 2020.
[26] X. Luo, H. Wu, H. Yuan, and M. Zhou, "Temporal pattern-aware QoS prediction via biased non-negative latent factorization of tensors," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1798–1809, May 2020.
[27] X. Luo, H. Wu, Z. Wang, J. Wang, and D. Meng, "A novel approach to large-scale dynamically weighted directed network representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9756–9773, Dec. 2022.
[28] X. Luo, H. Wu, and Z. Li, "Neulft: A novel approach to nonlinear canonical polyadic decomposition on high-dimensional incomplete tensors," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6148–6166, Jun. 2023.
[29] J. D. Hamilton, *Time Series Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 2020.
[30] P. D. Hoff, "Multilinear tensor regression for longitudinal relational data," *Ann. Appl. Statist.*, vol. 9, no. 3, pp. 1169–1193, 2015.

[31] R. Chen, H. Xiao, and D. Yang, "Autoregressive models for matrix-valued time series," *J. Econometrics*, vol. 222, no. 1, pp. 539–560, 2021.
[32] Z. Li and H. Xiao, "Multi-linear tensor autoregressive models," 2021, *arXiv:2110.00928*.
[33] D. Wang, Y. Zheng, and G. Li, "High-dimensional low-rank tensor autoregressive time series modeling," *J. Econometrics*, vol. 238, no. 1, 2024, Art. no. 105544.
[34] A. Patil, J. Viquerat, and E. Hachem, "Autoregressive transformers for data-driven spatiotemporal learning of turbulent flows," *APL Mach. Learn.*, vol. 1, no. 4, 2023, Art. no. 046101.
[35] C. J. Albers and L. F. Bringmann, "Inspecting gradual and abrupt changes in emotion dynamics with the time-varying change point autoregressive model," *Eur. J. Psychol. Assessment*, vol. 36, pp. 492–499, 2020.
[36] G. E. Primiceri, "Time varying structural vector autoregressions and monetary policy," *Rev. Econ. Stud.*, vol. 72, no. 3, pp. 821–852, 2005.
[37] J. M. Haslbeck, L. F. Bringmann, and L. J. Waldorp, "A tutorial on estimating time-varyingvector autoregressive models," *Multivariate Behav. Res.*, vol. 56, no. 1, pp. 120–149, 2021.
[38] L. F. Bringmann, E. L. Hamaker, D. E. Vigo, A. Aubert, D. Borsboom, and F. Tuerlinckx, "Changing dynamics: Time-varying autoregressive models using generalized additive modeling," *Psychol. Methods*, vol. 22, no. 3, pp. 409–425, 2017.
[39] P. Giudici, B. Tarantino, and A. Roy, "Bayesian time-varying autoregressive models of COVID-19 epidemics," *Biometrical J.*, vol. 65, no. 1, 2023, Art. no. 2200054.
[40] P. Congdon, "A spatio-temporal autoregressive model for monitoring and predicting COVID infection rates," *J. Geographical Syst.*, vol. 24, no. 4, pp. 583–610, 2022.
[41] R. P. Monti, P. Hellyer, D. Sharp, R. Leech, C. Anagnostopoulos, and G. Montana, "Estimating time-varying brain connectivity networks from functional MRI time series," *NeuroImage*, vol. 103, pp. 427–443, 2014.
[42] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM J. Appl. Dynamical Syst.*, vol. 15, no. 1, pp. 142–161, 2016.
[43] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Sci. Adv.*, vol. 3, no. 4, 2017, Art. no. e1602614.
[44] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of coordinates and governing equations," in *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 45, pp. 22445–22451, 2019.
[45] A. Towne, O. T. Schmidt, and T. Colonius, "Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis," *J. Fluid Mechan.*, vol. 847, pp. 821–867, 2018.
[46] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: JHU Press, 2013.
[47] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.
[48] J. C. Gower and G. B. Dijksterhuis, *Procrustes Problems*, vol. 30. Oxford, U.K.: OUP, 2004.
[49] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graphical Statist.*, vol. 15, no. 2, pp. 265–286, 2006.
[50] J.-F. Cai, H. Ji, Z. Shen, and G.-B. Ye, "Data-driven tight frame construction and image denoising," *Appl. Comput. Harmon. Anal.*, vol. 37, no. 1, pp. 89–105, 2014.
[51] J. Cates, R. C. Hoover, K. Caudle, C. Ozdemir, K. Braman, and D. Machette, "Forecasting multilinear data via transform-based tensor autoregression," 2022, *arXiv:2205.12201*.

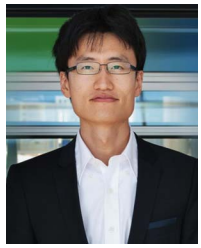


Xinyu Chen received the PhD degree from the University of Montreal, Montreal, QC, Canada. He is now a postdoctoral associate with the Massachusetts Institute of Technology, Cambridge, Massachusetts. His current research centers on machine learning, spatiotemporal data modeling, intelligent transportation systems, and urban science.

Dingyi Zhuang received the BS degree in mechanical engineering from Shanghai Jiao Tong University, in 2019, and the MEng degree in transportation engineering from McGill University, in 2021. He is currently working toward the PhD degree in transportation engineering with MIT Urban Mobility Lab. His research interests lie in deep learning, urban computing, and spatiotemporal data modeling.

HanQin Cai (Senior Member, IEEE) received the PhD degree in applied mathematics and computational sciences from the University of Iowa. He is currently the Paul N. Somerville Endowed assistant professor with the Department of Statistics and Data Science and the Department of Computer Science, University of Central Florida. He is also the director of Data Science Lab. His research interests include machine learning, data science, mathematical optimization, and applied harmonic analysis.

Shenhao Wang received the interdisciplinary PhD degree in computer and urban science from MIT, in 2020. He is an assistant professor with the University of Florida and research affiliate with MIT Urban Mobility Lab and Human Dynamics Group in Media Lab. His research focuses on developing interpretable, generalizable, and ethical deep learning models to analyze individual decision-making with applications to urban mobility.



Jinhua Zhao is currently the professor of Cities and Transportation with MIT. He brings behavioral science and transportation technology together to shape travel behavior, design mobility systems, and reform urban policies. He directs the MIT Urban Mobility Laboratory and Public Transit Laboratory.