Annual Review of Statistics and Its Application

Analysis of Tensor Time Series

Stevenson Bolivar, Shuo-Chieh Huang, and Rong Chen

Department of Statistics, Rutgers University, Piscataway, New Jersey, USA;
email: sb2230@stat.rutgers.edu, sh1976@stat.rutgers.edu, rongchen@stat.rutgers.edu

Annu. Rev. Stat. Appl. 2026. 13:369–98

First published as a Review in Advance on October 14, 2025

The *Annual Review of Statistics and Its Application* is online at statistics.annualreviews.org

https://doi.org/10.1146/annurev-statistics-042424-063308

Keywords

autoregressive models, factor models, tensor analysis, rank determination, spectral analysis

Abstract

This article provides a comprehensive overview of statistical methods developed for the analysis of tensor time series data, which have become increasingly prevalent across various fields such as economics, finance, biology, engineering, and the social sciences. The review focuses on three primary approaches: autoregressive modeling, factor modeling, and segmentation approaches. These methods leverage the inherent tensor structure to offer advantages such as dimension reduction, enhanced interpretability, and computational efficiency. The review focuses on model settings and their potential interpretations, discussing various estimation techniques for these models and their associated theoretical properties. In addition, we outline various applications using these models and discuss potential directions for future developments.

1. INTRODUCTION

The study of tensor time series—where each observation at a given time point takes the form of a tensor or multidimensional array—has gained increasing attention due to its relevance across a wide range of contemporary applications (Auddy et al. 2025; Banin et al. 2025; Barigozzi et al. 2025a; Chen et al. 2021a,b; Hsu et al. 2021; Li & Xiao 2021). This surge in interest is driven by the growing prevalence of structured, high-dimensional data collected over time in fields such as economics, finance, biology, engineering, and the social sciences.

Advancements in computing and data acquisition technologies over the past decade have enabled the collection of complex temporal data sets with inherent multiway structures. In many cases, each observation is more naturally represented as a tensor, rather than a vector, due to the presence of multiple classification dimensions (e.g., products, country, and economic indicator). When such tensor-valued observations are observed over time, they form a tensor time series, which generalizes the classical multivariate time series framework.

For example, a panel of economic indicators measured across multiple countries over time yields a matrix-valued time series (i.e., a second-order tensor) illustrated in Example 1 below. The import–export volumes across multiple product categories among several countries is an example of an order-3 tensor time series (Chen et al. 2021b). Example 2 shows the matrix time series version of the case, using the total volumes. Additional examples include the dynamic evolution of social networks across different interaction pathways, giving rise to an order-3 tensor time series (Goldenberg et al. 2010, Hanneke et al. 2010, Ji & Jin 2016, Kolaczyk & Csárdi 2014, Phan & Airoldi 2015, Snijders 2001, Zhao et al. 2012), as well as the temporal variation of wind speeds on a spatial grid (Hsu et al. 2021), which can naturally be organized into a matrix time series.

Example 1. Figure 1*a* shows a set of economic indicators for multiple countries observed over time, forming a matrix (order-2 tensor) time series (Wang et al. 2019). The data contain quarterly observations on four economic indicators for five countries. An illustrative analysis of this data set is given in Section 3.5.

Example 2. Figure 1*b* displays the monthly import–export volumes between several countries in several product categories (Chen & Chen 2022, Chen et al. 2021b). A simplified version of this data set is analyzed in Section 4.5 to illustrate the tensor factor models.

The analysis of tensor time series lies at the intersection of two major methodological threads: the extension of vector time series models to tensor-valued observations and the adaptation of tensor data analysis techniques to temporally dependent settings. Recent years have witnessed a surge of methodological developments to address the unique challenges posed by these data structures, including high dimensionality, multiway dependence, and complex dynamics.

This article provides a concise review of statistical methodologies for analyzing tensor time series. Section 2 begins with a brief overview of foundational models for vector time series and independent tensor data. We then discuss two primary approaches for analyzing tensor time series: tensor autoregressive (TenAR) models (Section 3) and tensor factor models (Section 4). Dynamic tensor factor models, which are combinations of the factor models with the autoregressive (AR) framework, are reviewed in Section 5. Additionally, a segmentation method that partitions tensor data to reduce the number of parameters is presented in Section 6. Section 7 surveys some R packages designed for tensor time series analysis. Finally, Section 8 concludes with an outline of potential directions for future research.

Throughout this article, we adopt the following notation. Bold lowercase letters denote vectors, bold uppercase letters denote matrices, and script letters signify tensors of order 3 or higher. We use $\odot$ to denote the tensor outer product, $\otimes$ the Kronecker product, and $\times_k$ the mode-k product

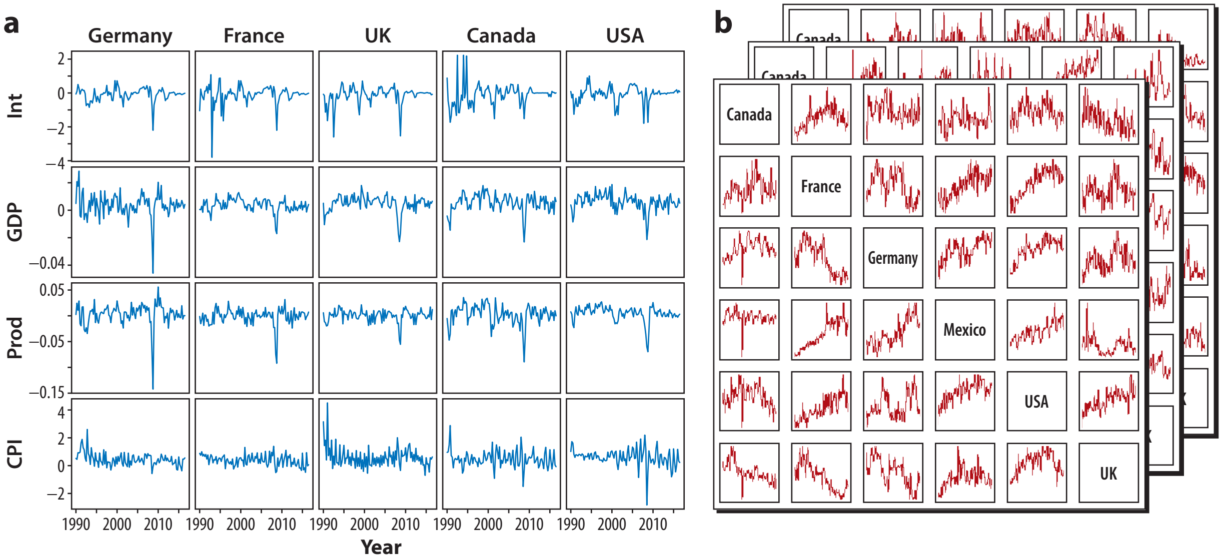

Figure 1

(*a*) A set of economic indicators of a group of countries (Germany, France, the United Kingdom, Canada, and the United States) (data from Wang et al. 2019). The economic indicators are three-month interbank interest rates (Int, first differenced), gross domestic product growth (GDP, first log differenced), total manufacturing production growth (Prod, first log differenced), and consumer price index (CPI, first log differenced) core inflation. (*b*) Import–export volume of several product categories among a group of countries (data from Chen et al. 2021b).

of a tensor and a matrix. For a K-mode tensor $\mathcal{X}$ with dimensions $d_1, d_2, \ldots, d_K$, its vectorization is denoted as $\mathrm{vec}(\mathcal{X})$, which is a vector of dimension $d_1 d_2 \ldots d_K$. The unfolding of $\mathcal{X}$ along mode k produces a matrix of dimension $d_k \times (d_1 \ldots d_{k-1} d_{k+1} \ldots d_K)$ and is denoted as $\mathrm{mat}_k(\mathcal{X})$. LSVD_m is the left singular matrix composed of the first m left singular vectors corresponding to the largest m singular values. For detailed definitions of these products and other basic tensor operations, we refer readers to Kolda & Bader (2009).

2. REVIEW OF VECTOR TIME SERIES AND TENSOR DATA ANALYSIS

Multivariate time series analysis has been studied extensively, primarily through two main approaches: the AR model and the factor model.

Vector autoregressive (VAR) models, formally expressed as

$$\boldsymbol{x}_t = \sum_{i=1}^{p} \boldsymbol{\Phi}_i \boldsymbol{x}_{t-i} + \boldsymbol{\varepsilon}_t, \qquad 1.$$

have been extensively studied and are widely applied in many fields (Hannan 1970, Lütkepohl 2005, Reinsel 1993, Shumway & Stoffer 2017, Tiao & Box 1981, Tsay 2013). VAR models facilitate multistep forecasting and impulse response analysis, both of which are well understood. However, since each coefficient matrix $\boldsymbol{\Phi}_i$ has dimension $d \times d$, where d is the dimension of $\boldsymbol{x}_t$, the application of VAR models is often limited to low-dimensional time series. To address this limitation, recent research has focused on developing regularized VAR methods to improve estimation and interpretability in high-dimensional settings (Basu & Michailidis 2015, Davis et al. 2016, Han et al. 2015, Kock & Callot 2015, Lin & Michailidis 2017, Loh & Wainwright 2012, Melnyk & Banerjee 2016, Nicholson et al. 2017).

An alternative approach to modeling high-dimensional time series is dynamic vector factor models (VFMs) (Ahn & Horenstein 2013; Bai 2003; Bai & Ng 2002, 2013, 2023; Bolivar et al. 2021; Caro & Peña 2025; Engle & Kroner 1995; Fan et al. 2013, 2022; Forni et al. 2000; Lam & Yao 2012; Onatski 2009, 2010; Peña & Box 1987; Peña & Poncela 2006; Stock & Watson 2016; Tiao & Tsay 1989). These models are based on the premise that noncausal correlations among variables arise from a few latent, unobserved factors that are causal of the observed variables. VFMs assume a linear relationship between the observed time series and a small number of latent factors:

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{f}_t + \boldsymbol{\varepsilon}_t, \qquad 2.$$

where $\boldsymbol{A}$ is the loading matrix, which relates the latent factors $\boldsymbol{f}_t$ to the observed vector time series $\boldsymbol{x}_t$. The VFM serves as an effective tool for dimension reduction while also providing an interpretable framework to explain correlations and comovements among the components of $\boldsymbol{x}_t$. Thus, it has been widely applied in economics, finance, and various other fields.

The study of tensors under the assumption of independent and identically distributed (i.i.d.) data has proven to be a powerful and versatile framework at the confluence of statistics, computer science, machine learning, and signal processing. Key research areas include tensor completion (Liu et al. 2012; Xia & Yuan 2019; Xia et al. 2021; Yuan & Zhang 2016, 2017; Zhang 2019), tensor decomposition (Anandkumar et al. 2015; Kolda & Bader 2009; Liu et al. 2014, 2022; Sidiropoulos et al. 2017; Sun et al. 2017), tensor denoise (Xia & Zhou 2019), and tensor regression (Chen et al. 2019, Hoff 2015, Lock 2018, Raskutti et al. 2019, Zhang et al. 2020, Zhou et al. 2013). The foundation of these methods largely relies on two fundamental tensor decompositions: the Tucker decomposition, which expresses a tensor $\mathcal{X}$ as

$$\mathcal{X} = \mathcal{F} \times_1 \boldsymbol{A}_1 \times_2 \ldots \times_K \boldsymbol{A}_K,$$

and the canonical polyadic (CP) decomposition, which decomposes $\mathcal{X}$ as a sum of rank-1 tensors,

$$\mathcal{X} = \sum_{j=1}^{R} f_{jt}\boldsymbol{a}_{j1} \odot \boldsymbol{a}_{j2} \odot \cdots \odot \boldsymbol{a}_{jK}.$$

These decompositions provide essential tools for dimensionality reduction, structure learning, and efficient representation in high-dimensional settings.

3. TENSOR AUTOREGRESSIVE MODELS

3.1. Precursor: Matrix Autoregressive Models

To motivate the AR model for higher-order tensors, we first introduce the matrix autoregressive (MAR) model of order 1, denoted as MAR(1), which was studied by Chen et al. (2021a). This model extends the VAR model to matrix-valued time series by incorporating a bilinear form:

$$\boldsymbol{X}_t = \boldsymbol{A}_1\boldsymbol{X}_{t-1}\boldsymbol{A}_2^\top + \boldsymbol{E}_t, \qquad 3.$$

where $\boldsymbol{X}_t$ is a $d_1 \times d_2$ matrix observed at time t; $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are $d_1 \times d_1$ and $d_2 \times d_2$ AR coefficient matrices, respectively; and $\boldsymbol{E}_t$ is a $d_1 \times d_2$ matrix white noise process.

The model in Equation 3 is an AR model, as $\boldsymbol{X}_t$ is a function of $\boldsymbol{X}_{t-1}$, with an additive white noise innovation. Hence, multistep predictions and impulse response functions can be easily derived. Note that the model can be written in the VAR form $\mathrm{vec}(\boldsymbol{X}_t) = (\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)\mathrm{vec}(\boldsymbol{X}_{t-1}) + \mathrm{vec}(\boldsymbol{E}_t)$. When compared with the VAR model in Equation 1 with $p = 1$, the coefficient matrix $\boldsymbol{\Phi}_1$, which has $d_1^2 d_2^2$ parameters, is constrained to the form of $(\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)$ with just $d_1^2 + d_2^2$ parameters, achieving significant dimension reduction. More importantly, it retains the matrix form of the data $\boldsymbol{X}_t$, facilitating the interpretation of the coefficient matrices. For example, if $\boldsymbol{A}_2 = \boldsymbol{I}$, the identity matrix, then each column of $\boldsymbol{X}_t$ follows the same VAR model, $\boldsymbol{X}_{t,k} = \boldsymbol{A}_1\boldsymbol{X}_{t-1,k} + \boldsymbol{E}_{t,k}$. If the first row of $\boldsymbol{A}_1$ is zero, then $\boldsymbol{X}_t$ does not depend on the first row of $\boldsymbol{X}_{t-1}$.

The VAR formulation also shows that the MAR(1) process is stationary and causal if $\rho(\boldsymbol{A}_1)\rho(\boldsymbol{A}_2) < 1$, where $\rho(\boldsymbol{A})$ denotes the spectral radius of the matrix $\boldsymbol{A}$. Note that we need to impose $\|\boldsymbol{A}_1\|_F = 1$ for identification of the MAR(1) model.

Estimation of Equation 3 can be accomplished by projecting the least squares solution $\hat{\boldsymbol{\Phi}}$ of a general VAR(1) model onto the space of matrices that are Kronecker products of two matrices:

$$(\hat{\boldsymbol{A}}_1, \hat{\boldsymbol{A}}_2) = \arg\min_{\boldsymbol{A}_1, \boldsymbol{A}_2} \|\hat{\boldsymbol{\Phi}} - \boldsymbol{A}_2 \otimes \boldsymbol{A}_1\|_F^2, \tag{4.}$$

where $\hat{\boldsymbol{\Phi}} = \arg\min_{\boldsymbol{\Phi}} \sum_{t=2}^{T} \|\mathrm{vec}(\boldsymbol{X}_t) - \boldsymbol{\Phi}\mathrm{vec}(\boldsymbol{X}_{t-1})\|_F^2$ is the least squares estimator of VAR(1). This approach is computationally simple, relying on a rearrange operation followed by principal component analysis (PCA). However, it may require a large sample size for $\hat{\boldsymbol{\Phi}}$ to be sufficiently accurate. Alternatively, by iteratively updating $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ while holding the other fixed, an iterative least squares procedure can be used to minimize

$$\sum_{t=2}^{T} \left\|\boldsymbol{X}_t - \boldsymbol{A}_1\boldsymbol{X}_{t-1}\boldsymbol{A}_2^\top\right\|_F^2. \tag{5.}$$

Under some mild assumptions, Chen et al. (2021a) showed that both estimators are $\sqrt{T}$-asymptotically normal, for fixed d_1 and d_2. The iterative least squares method is computationally more efficient, though it may converge to a local minimum. In practice, it is advisable to initialize the iterative least squares method by the projection estimators.

For Bayesian estimation of the MAR model, Chan & Qi (2025) considered a Bayesian shrinkage prior and efficient estimation. In this approach, it is assumed that $\mathrm{vec}(\boldsymbol{E}_t) \sim \mathrm{N}(\boldsymbol{0}, \omega_t\boldsymbol{\Sigma}_c \otimes \boldsymbol{\Sigma}_r)$, where ω_t is the mixture variable with a specified distribution. This formulation makes the resulting MAR model flexible enough to accommodate time-varying volatility, non-Gaussian errors, and possible outliers. For instance, if $\omega_t = \exp(h_t)$, with $h_t = \phi h_{t-1} + u_t$, $u_t \sim \mathrm{N}(0, \sigma_h^2)$, then the error process follows the common stochastic volatility model (Carriero et al. 2016). Using the normal-inverse-Wishart prior for $(\boldsymbol{A}_1, \boldsymbol{\Sigma}_r)$ and $(\boldsymbol{A}_2, \boldsymbol{\Sigma}_c)$, Chan & Qi (2025) provide efficient algorithms for posterior sampling.

The MAR(1) model in Equation 3 can be extended to

$$\boldsymbol{X}_t = \sum_{k=1}^{R} \boldsymbol{A}_k\boldsymbol{X}_{t-1}\boldsymbol{B}_k^\top + \boldsymbol{E}_t. \tag{6.}$$

Under this model, $\mathrm{vec}(\boldsymbol{X}_t)$ assumes the VAR(1) form in Equation 1 with $\boldsymbol{\Phi}_1 = \sum_{k=1}^{R} \boldsymbol{B}_k \otimes \boldsymbol{A}_k$. It is noteworthy that any matrix $\boldsymbol{\Phi} \in \mathbb{R}^{(d_1d_2)\times(d_1d_2)}$ has the Kronecker decomposition $\boldsymbol{\Phi} = \sum_{k=1}^{d_1d_2} \boldsymbol{B}_k \otimes \boldsymbol{A}_k$. Hence, the model in Equation 6 can be viewed as an approximation of the general VAR(1) model by truncating the Kronecker decomposition to R terms.

The model in Equation 6 can be further extended to the AR(p) model,

$$\boldsymbol{X}_t = \sum_{i=1}^{p}\sum_{k=1}^{R_i} \boldsymbol{A}_{i,k}\boldsymbol{X}_{t-i}\boldsymbol{B}_{i,k}^\top + \boldsymbol{E}_t. \tag{7.}$$

In addition, MAR models can also be generalized to matrix autoregressive moving average (ARMA) models. For further details and an in-depth review of the matrix ARMA models, readers are directed to Tsay (2024).

To achieve further dimension reduction, Xiao et al. (2021) considered the case where the matrices $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ in Equation 3 are of low rank. Write $\boldsymbol{A}_i = \boldsymbol{A}_{i,l}\boldsymbol{A}_{i,c}^\top$, where $\boldsymbol{A}_{i,l}$ and $\boldsymbol{A}_{i,c}$ are

$(d_i \times r_i)$-dimensional matrices with full column rank, $r_i \leqslant d_i$, for $i = 1, 2$. In this formulation, the reduced-rank matrix autoregressive (RRMAR) model can be interpreted as follows. Define $\boldsymbol{F}_t = \boldsymbol{A}_{1,c}^\top \boldsymbol{X}_{t-1} \boldsymbol{A}_{2,c}$, which serves as a composite, lower-dimensional representation of $\boldsymbol{X}_{t-1}$. Then, Equation 3 can be written as

$$\boldsymbol{X}_t = \boldsymbol{A}_{1,l} \boldsymbol{F}_t \boldsymbol{A}_{2,l}^\top + \boldsymbol{E}_t,$$

which is akin to the matrix factor model (MFM), introduced by Wang et al. (2019) (see Section 4.1.1), with the additional generating mechanism $\boldsymbol{F}_t = \boldsymbol{A}_{1,c}^\top \boldsymbol{X}_{t-1} \boldsymbol{A}_{2,c}$, which can be viewed as a composite index summarizing the information in $\boldsymbol{X}_{t-1}$. Unlike the MFM, the RRMAR model is fully generative and thus can be used for prediction.

To estimate the RRMAR model, one can employ alternating minimization with rank constraints. The resulting algorithm is closely related to the classical reduced-rank regression model (Velu & Reinsel 2013). To illustrate the alternating minimization algorithm, suppose $\boldsymbol{A}_2$ is fixed. Then, the solution to $\min \sum_{t=2}^T \|\boldsymbol{X}_t - \boldsymbol{A}_1 \boldsymbol{X}_{t-1} \boldsymbol{A}_2^\top\|_F^2$ subject to the constraint $\mathrm{rank}(\boldsymbol{A}_1) = r_1$ is given by $\check{\boldsymbol{A}}_1 = \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{S}_{yx} \boldsymbol{S}_{xx}^{-1}$, where $\boldsymbol{S}_{xx} = \sum_{t=2}^T \boldsymbol{X}_{t-1} \boldsymbol{A}_2^\top \boldsymbol{A}_2 \boldsymbol{X}_{t-1}^\top$, $\boldsymbol{S}_{yx} = \sum_{t=2}^T \boldsymbol{X}_t \boldsymbol{A}_2 \boldsymbol{X}_{t-1}^\top$, and $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{r_1})$ contains the leading eigenvectors of $\boldsymbol{S}_{yx} \boldsymbol{S}_{xx}^{-1} \boldsymbol{S}_{xy}$. A similar expression for minimizing with respect to $\boldsymbol{A}_2$ can also be obtained (for details, see Xiao et al. 2021). In addition, when $\boldsymbol{E}_t$ admits a special covariance structure, namely $\mathrm{Cov}(\mathrm{vec}(\boldsymbol{E}_t)) = \boldsymbol{\Sigma}_2 \otimes \boldsymbol{\Sigma}_1$, where $\boldsymbol{\Sigma}_i \in \mathbb{R}^{d_i \times d_i}$, $i = 1$, 2, Xiao et al. (2021) also propose a maximum likelihood estimator. This estimator is connected to the classical canonical correlation analysis. Under some regularity assumptions, both estimators are shown to achieve $\sqrt{T}$-normality. Similar to the general MAR model, RRMAR can also be extended to Equation 7 with reduced-rank coefficient matrices.

3.2. Tensor Autoregressive Models

For AR models of dynamic tensors of order 3 or higher, it is natural to extend the MAR model introduced above. Li & Xiao (2021) extend the MAR model to multilinear TenAR models. Let $\{\mathcal{X}_t\}$ be a tensor time series, where at each time t an order-K tensor $\mathcal{X}_t \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_K}$ is observed. The TenAR model of order p, denoted by TenAR(p) [or, more precisely, TenAR(p, $R_1, \ldots, R_p$), where R_i is the number of multilinear terms for lag i], is given by

$$\mathcal{X}_t = \sum_{i=1}^{p} \sum_{r=1}^{R_i} \mathcal{X}_{t-i} \times_1 \boldsymbol{A}_1^{(ir)} \times_2 \cdots \times_K \boldsymbol{A}_K^{(ir)} + \mathcal{E}_t, \qquad 8.$$

where $\boldsymbol{A}_k^{(ir)} \in \mathbb{R}^{d_k \times d_k}$ is the coefficient matrix associated with mode k for the rth term at lag i, $\mathcal{E}_t \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_K}$ is a tensor white noise process satisfying $\mathrm{Cov}(\mathcal{E}_t, \mathcal{E}_s) = \boldsymbol{0}$ whenever $s \neq t$, and p is the AR order. Similar to MAR models, we have

$$\mathrm{vec}(\mathcal{X}_t) = \sum_{i=1}^{p} \boldsymbol{\Phi}_i \mathrm{vec}(\mathcal{X}_{t-i}) + \mathrm{vec}(\mathcal{E}_t),$$

where

$$\boldsymbol{\Phi}_i = \sum_{r=1}^{R_i} \left[\otimes_{k=K}^{1} \boldsymbol{A}_k^{(ir)} \right] := \sum_{r=1}^{R_i} \left[\boldsymbol{A}_K^{(ir)} \otimes \boldsymbol{A}_{K-1}^{(ir)} \otimes \cdots \otimes \boldsymbol{A}_1^{(ir)} \right].$$

Hence, the TenAR model is also a VAR whose coefficient matrices have the structure of a sum of a few Kronecker products. The causality condition for Equation 8 follows directly from the VAR representation: $\{\mathcal{X}_t\}$ is causal if $\det \boldsymbol{\Psi}(z) \neq 0$ for all $|z| \leqslant 1$, where $\boldsymbol{\Psi}(z) = \boldsymbol{I} - \sum_{i=1}^{p} (\otimes_{k=K}^{1} \boldsymbol{A}_k^{(ir)}) z^i$. The model in Equation 8 clearly suffers from a lack of identification. Interestingly, even for the TenAR(1) model, the identification issue is quite different from the matrix case. Li & Xiao (2021) give a sufficient condition for identification when $K \geqslant 3$, which states that $\sum_{k=1}^{K} \kappa(\mathbb{A}_k) \geqslant 2R + K - 1$, where $\mathbb{A}_k = (\mathrm{vec}(\mathbf{A}_k^{(1,1)}); \ldots; \mathrm{vec}(\mathbf{A}_k^{(1,R)}))$ and $\kappa(\boldsymbol{A})$ denotes the Kruskal rank of $\boldsymbol{A}$, defined as the maximum value κ such that any κ columns of $\boldsymbol{A}$ are linearly independent.

Estimation of the TenAR model is conceptually similar to that of the MAR model, using an extension of the projection estimator in Equation 4 and alternative least squares estimator in Equation 5. However, unlike the matrix case, the projection estimator for higher-order tensors requires computing the best low-rank approximation, which may not exist and is in general NP-hard. In practice, algorithms such as those proposed by Anandkumar et al. (2015) and Sun et al. (2017) can be employed to solve the CP decomposition with theoretical guarantees under some assumptions. Moreover, due to the increased model complexity, the alternative LS estimator for TenAR requires more accurate initial values to ensure reliable convergence (for details, see Li & Xiao 2021). These issues again highlight the difference between TenAR and MAR models.

Due to the model complexity of the TenAR model, it is crucial to have a model selection procedure. Li & Xiao (2021) proposed an extended Bayesian information criterion (EBIC) to jointly select the AR order p and the Kronecker ranks R_i, $i = 1, \ldots, p$. Specifically, for any given order p and rank vector $\boldsymbol{R}_p := (R_1, \ldots, R_p)$, define the information criterion (IC) as

$$\mathrm{EBIC}(p; \boldsymbol{R}_p) := \frac{1}{2} \log \left(\frac{1}{\mathrm{d}T} \sum_t \left\| \mathcal{X}_t - \sum_{i=1}^{p} \sum_{r=1}^{R_i} \mathcal{X}_{t-i} \times_1 \hat{\boldsymbol{A}}_1^{(ir)} \times_2 \cdots \times_K \hat{\boldsymbol{A}}_K^{(ir)} \right\|_F^2 \right) + g(d, T) \sum_{i=1}^{p} R_i,$$

where $\hat{\boldsymbol{A}}_k^{(ir)}$ are the estimates obtained for the given p and $\boldsymbol{R}_p$, and $g(d, T)$ is a penalty function. Under the conditions that $g(d, T) \to 0$ and $\frac{T}{d} g(d, T) \to \infty$ as $T \to \infty$, Li & Xiao (2021) show that this criterion achieves consistent selection of p and $\boldsymbol{R}_p$ under settings with fixed or diverging dimensions.

Prediction under a TenAR model follows the same principles as the univariate and VAR models. Specifically, the best linear h-step ahead prediction of $\mathcal{X}_{t+h}$, based on $\mathcal{X}_1, \ldots, \mathcal{X}_t$, can be obtained recursively using

$$\hat{\mathcal{X}}_t(h) = \sum_{i=1}^{p} \sum_{r=1}^{R_i} \hat{\mathcal{X}}_t(h-i) \times_1 \hat{\boldsymbol{A}}_1^{(ir)} \times_2 \cdots \times_K \hat{\boldsymbol{A}}_K^{(ir)},$$

where $\hat{\mathcal{X}}_t(h-i) = \mathcal{X}_{t+h-i}$ if $h \leqslant i$. The prediction standard errors can also be calculated recursively based on the estimated model parameters, allowing for uncertainty quantification in forecasting applications.

3.3. Low-Rank Tensor Autoregressive Model

The low-rank tensor AR model, as proposed by D. Wang et al. (2024), introduces an approach to modeling tensor-valued time series by leveraging a Tucker decomposition, effectively reducing the dimensionality and preserving the intrinsic multiway structure of the data. The model assumes that, with appropriate orthonormal matrices $\boldsymbol{U}_i \in \mathbb{R}^{d_i \times r_i}$, $i = 1, \ldots, 2K$, and $d_{i+K} = d_i$, the low-dimensional output factors $\mathcal{X}_t \times_{K+1} \mathbf{U}_{K+1}^\top \times_{K+2} \ldots \times_{2K} \mathbf{U}_{2K}^\top$ and the low-dimensional input factors $\mathcal{X}_{t-1} \times_1 \mathbf{U}_1^\top \times_2 \ldots \times_K \mathbf{U}_K^\top$ are linked via

$$\mathcal{X}_t \times_{i=K+1}^{2K} \mathbf{U}_i^\top = \langle \mathcal{G}, \mathcal{X}_{t-1} \times_{i=1}^{K} \mathbf{U}_i^\top \rangle + \mathcal{E}_t \times_{i=K+1}^{2K} \mathbf{U}_i^\top, \qquad 9.$$

where $\mathcal{G} \in \mathbb{R}^{r_1 \times \ldots \times r_K \times r_{K+1} \times \ldots \times r_{2K}}$ is an unrestricted coefficient tensor and $\mathcal{E}_t$ is a tensor white noise process. In Equation 9, the matrices $\boldsymbol{U}_i$, $i = 1, 2, \ldots, 2K$, and the tensor $\mathcal{G}$ are unknown, and the inner product is the contraction inner product, defined as follows. If $\mathcal{A} \in \mathbb{R}^{d_1 \times \ldots \times d_{2K}}$ and

$\mathcal{B} \in \mathbb{R}^{d_1 \times \ldots \times d_K}$, then $\langle \mathcal{A}, \mathcal{B} \rangle$ is the $d_{K+1} \times \ldots \times d_{2K}$ tensor whose entries are

$$\langle \mathcal{A}, \mathcal{B} \rangle [i_{K+1}, \ldots, i_{2K}] = \sum_{i_1=1}^{d_1} \ldots \sum_{i_K=1}^{d_K} \mathcal{A}[i_1, \ldots, i_K, i_{K+1}, \ldots, i_{2K}] \mathcal{B}[i_1, \ldots, i_K]$$

for $1 \leqslant i_{K+j} \leqslant d_{K+j}$, $1 \leqslant j \leqslant K$. Using the contraction inner product, Equation 9 can be equivalently written as

$$\mathcal{X}_t = \langle \mathcal{A}, \mathcal{X}_{t-1} \rangle + \mathcal{E}_t, \qquad 10.$$

where $\mathcal{A}$ is the $(d_1 \times \ldots \times d_K \times d_1 \times \ldots \times d_K)$ coefficient tensor, which admits a Tucker decomposition $\mathcal{A} = \mathcal{G} \times_{i=1}^{2K} \mathbf{U}_i$. By vectorizing both sides, Equation 10 can be cast into a VAR form, from which the conditions for causality can be derived (for details, see D. Wang et al. 2024).

Due to the nonconvex nature of directly estimating $\mathcal{G}$ and $\mathbf{U}_i$s, a penalized estimation approach is adopted based on the formulation in Equation 10. Specifically, the estimator

$$\hat{\mathcal{A}} = \arg\min_{\mathcal{A}} \sum_{t=2}^{T} \| \mathcal{X}_t - \langle \mathcal{A}, \mathcal{X}_{t-1} \rangle \|_F^2 + R(\mathcal{A})$$

is employed, where $R(\cdot)$ is a suitable low-rank-inducing penalty. D. Wang et al. (2024) proposed three different penalty functions based on the nuclear norms of matrix unfoldings of $\mathcal{A}$ and obtained their respective convergence rates under the high-dimensional setup and the approximate low-rank condition. These estimators all require at least $(\prod_{i=1}^{K} d_i)/T \to 0$ for consistency. However, by employing a nonconvex estimator, a faster convergence rate can be obtained, requiring only $(\sum_{i=1}^{2K} d_i r_i + \prod_{i=1}^{2K} r_i)/T \to 0$ for consistency. For the nonconvex approach, global optimality is not guaranteed, and thus careful initialization is needed.

In general, there is a statistical–computational gap of the tensor regression problem (Equation 10) (Luo & Zhang 2024), a significant departure from the vector regression models. This fundamental challenge highlights the need for further research to develop efficient algorithms and establish theoretical guarantees.

3.4. Extensions of Tensor Autoregressive Models and Other Related Models

In this subsection, we survey some recent extensions of the TenAR model, some of which nicely complement the basic model in certain applications, while others may point to new research directions.

3.4.1. Spatiotemporal data. In some applications, observations are collected on a spatial grid over time, naturally forming a matrix time series. The MAR model for such a time series can be employed to effectively capture the spatiotemporal dependencies. In these applications, dimension reduction can be achieved through additional structural assumptions. Hsu et al. (2021) employed an MAR(p) model for spatiotemporal data, leveraging the fixed-rank kriging technique (Cressie & Johannesson 2008), which assumes $\boldsymbol{\Phi} = \mathrm{Cov}(\mathrm{vec}(\boldsymbol{E}_t)) = \boldsymbol{F}\boldsymbol{M}\boldsymbol{F}^\top + \sigma_e^2 \boldsymbol{I}$, where $\boldsymbol{F}$ is a known $d_1 d_2 \times k$ matrix of basis functions and $\boldsymbol{M} \in \mathbb{R}^{k \times k}$, $k < d_1 d_2$, is unknown. In addition, since spatial dependence is strongest among nearby grid points, Hsu et al. (2021) incorporated additional structural assumptions on the coefficient matrices $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, enforcing banded or sparse structure. With these dimension reduction techniques, efficient algorithms were derived for alternating maximum likelihood estimation.

3.4.2. Nonlinear matrix autoregressive models. For nonlinear AR modeling, Yu et al. (2025b) proposed a two-way MAR model with thresholds, termed 2-MART. Let z_t and w_t be two threshold variables governing the nonlinear AR dynamics along the row and column dimensions, respectively. The 2-MART model is given by

$$\boldsymbol{X}_t = \sum_{i=1}^{2}\sum_{j=1}^{2} \boldsymbol{A}_i \boldsymbol{X}_{t-1} \boldsymbol{B}_j^\top I_{t,ij}(\boldsymbol{\tau}) + \boldsymbol{E}_t,$$

where $\boldsymbol{\tau} = (r, s)$, and $I_{t,11}(\boldsymbol{\tau}) = I(z_{t-1} \leqslant r, w_{t-1} \leqslant s)$, $I_{t,21}(\boldsymbol{\tau}) = I(z_{t-1} > r, w_{t-1} \leqslant s)$, $I_{t,12}(\boldsymbol{\tau}) = I(z_{t-1} \leqslant r, w_{t-1} > s)$, and $I_{t,22}(\boldsymbol{\tau}) = I(z_{t-1} > r, w_{t-1} > s)$ are indicator functions. Estimation of 2-MART is performed via an alternating least squares approach. Theoretical results are also obtained (for details, see Yu et al. 2025b).

Another important aspect of matrix-valued time series is conditional heteroscedasticity. Yu et al. (2025a) introduced the matrix generalized AR conditional heteroscedasticity (GARCH) model, which generalizes the celebrated GARCH process (Bollerslev 1987). Generally, the matrix GARCH process takes the form

$$\boldsymbol{X}_t = \boldsymbol{U}_t^{1/2} \boldsymbol{Z}_t \boldsymbol{V}_t^{1/2},$$

where $\boldsymbol{U}_t \in \mathbb{R}^{d_1 \times d_1}$ and $\boldsymbol{V}_t \in \mathbb{R}^{d_2 \times d_2}$ are $\mathcal{F}_{t-1}$-measurable, with $\{\mathcal{F}_t\}$ being a filtration representing information up to time t. Here, $\boldsymbol{Z}_t$ is i.i.d., with mean zero and $\mathrm{Cov}(\mathrm{vec}(\boldsymbol{Z}_t)) = \boldsymbol{I}_{d_1 d_2}$. This model implies the following conditional second-moment properties:

$$\mathbb{E}(\boldsymbol{X}_t \boldsymbol{X}_t^\top | \mathcal{F}_{t-1}) = \mathrm{tr}(\boldsymbol{V}_t)\boldsymbol{U}_t, \quad \mathbb{E}(\boldsymbol{X}_t^\top \boldsymbol{X}_t | \mathcal{F}_{t-1}) = \mathrm{tr}(\boldsymbol{U}_t)\boldsymbol{V}_t.$$

Thus, the row and column conditional heteroscedasticity is entangled through $\boldsymbol{U}_t$ and $\boldsymbol{V}_t$, making it challenging to separately identify their contributions. To avoid some identification issues, Yu et al. (2025a) proposed the following model:

$$\begin{aligned}
\boldsymbol{S}_{1t} &= \boldsymbol{A}_0 \boldsymbol{A}_0^\top + \boldsymbol{A}_1 \boldsymbol{X}_{t-1} \boldsymbol{X}_{t-1}^\top \boldsymbol{A}_1^\top + \boldsymbol{A}_2 \boldsymbol{S}_{1,t-1} \boldsymbol{A}_2^\top,\\
\boldsymbol{S}_{2t} &= \boldsymbol{B}_0 \boldsymbol{B}_0^\top + \boldsymbol{B}_1 \boldsymbol{X}_{t-1}^\top \boldsymbol{X}_{t-1} \boldsymbol{B}_1^\top + \boldsymbol{B}_2 \boldsymbol{S}_{2,t-1} \boldsymbol{B}_2^\top,\\
y_t &= \omega + \alpha \mathrm{tr}(\boldsymbol{X}_{t-1} \boldsymbol{X}_{t-1}^\top) + \beta y_{t-1}.
\end{aligned}$$

The variance matrices are then

$$\boldsymbol{U}_t = \frac{\boldsymbol{S}_{1t}}{\mathrm{tr}(\boldsymbol{S}_{1t})} y_t, \quad \boldsymbol{V}_t = \frac{\boldsymbol{S}_{2t}}{\mathrm{tr}(\boldsymbol{S}_{2t})}.$$

Yu et al. (2025a) showed that if the density of $\boldsymbol{Z}_t$ is continuous and positive everywhere, and

$$\|\boldsymbol{A}_1\|^2 + \|\boldsymbol{B}_1\|^2 + \alpha + \max\{\|\boldsymbol{A}_2\|^2, \|\boldsymbol{B}_2\|^2, \beta\} < 1$$

holds, then there exists a strictly stationary solution $\{\boldsymbol{X}_t\}$ with a bounded second moment. Moreover, the solution is unique and ergodic. Estimation can be conducted using a quasi-maximum likelihood estimation (QMLE). Theoretical results, including $\sqrt{T}$-normality of the QMLE estimators, are provided by Yu et al. (2025a).

3.4.3. Nonstationary matrix autoregressive model. In practice, nonstationary time series are commonly encountered and have been extensively studied. Li & Xiao (2024) extend the RRMAR technique to analyze the cointegrated MAR model. The error-correction form of a cointegrated vector time series naturally generalizes to the matrix setting as

$$\Delta \boldsymbol{X}_t = \boldsymbol{A}_1 \boldsymbol{X}_{t-1} \boldsymbol{A}_2^\top + \sum_{i=1}^{k} \boldsymbol{B}_{i1} \Delta \boldsymbol{X}_{t-i} \boldsymbol{B}_{i2}^\top + \boldsymbol{D} + \boldsymbol{E}_t, \tag{11.}$$

where Δ is the difference operator with $\Delta \boldsymbol{X}_t = \boldsymbol{X}_t - \boldsymbol{X}_{t-1}$, $\boldsymbol{D}$ is a matrix constant, and $\boldsymbol{A}_j$ is of low rank $r_j \leqslant d_j$, $j = 1, 2$, while the $\boldsymbol{B}_{ij}$s are without rank constraints. This formulation implies the usual vector error-correction representation,

$$\Delta\text{vec}(\boldsymbol{X}_t) = (\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)\text{vec}(\boldsymbol{X}_{t-1}) + \sum_{i=1}^{K}(\boldsymbol{B}_{i2} \otimes \boldsymbol{B}_{i1})\Delta\text{vec}(\boldsymbol{X}_{t-i}) + \text{vec}(\boldsymbol{D}) + \text{vec}(\boldsymbol{E}_t),$$

where the coefficient follows a Kronecker product form. Thus, the cointegrating vector $\boldsymbol{\beta}$, which makes $\boldsymbol{\beta}^\top\text{vec}(\boldsymbol{X}_t)$ stationary, must satisfy $\boldsymbol{\beta} = \boldsymbol{\beta}_2 \otimes \boldsymbol{\beta}_1$, implying that $\boldsymbol{\beta}_1^\top \boldsymbol{X}_t \boldsymbol{\beta}_2$ is stationary. The least squares and maximum likelihood estimators introduced earlier remain applicable to Equation 11. We refer readers to Li & Xiao (2024) for a discussion of their asymptotic properties.

3.5. An Illustrative Example

To illustrate the practical applicability of TenAR models, we consider the multicountry economic indicator series displayed in **Figure 1*a*** and introduced in Example 1, which has been analyzed by Chen et al. (2021a). **Table 1** shows significant coefficients of an MAR(1) model in Equation 3 using the maximum likelihood estimator. Several interesting patterns emerge. For instance, the first column of the top panel shows that the US economic indicators from the previous quarter significantly influence all countries. In the bottom panel, the first column suggests that previous-quarter interest rate negatively affects gross domestic product (GDP) and production but has little impact on consumer price index (CPI). Meanwhile, the last column shows CPI only has a negative effect on interest rate and a positive effect on itself.

Figure 2*a* shows one-step-ahead out-of-sample rolling prediction performance of the last 20 quarters of the data. We compare the MAR(1) model, an EBIC-selected reduced-rank MAR(1) model with rank 3 for both $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ in Equation 3, and a VAR(1) model by stacking the matrix time series into a 20-dimensional vector time series. The figure plots the accumulated root mean square prediction errors for each series under the three models. The MAR models consistently outperform the VAR benchmark for most of the element series, especially the GDP and manufacturing production series.

The MAR framework is also able to facilitate structural analysis. In particular, one can obtain the impulse response function to study how shocks in one series propagate to other series (Chen et al. 2021a). **Figure 2*b*** plots the impulse response function with a unit-variance shock to US

Table 1 Significance for each entry of the estimated coefficient matrices

Country	USA	Germany	France	United Kingdom	Canada
USA	+	0	0	+	0
Germany	+	0	+	+	−
France	+	0	+	0	0
United Kingdom	+	0	+	+	0
Canada	+	0	0	+	+
Indicator	**Int**	**GDP**	**Prod**	**CPI**	
Int	+	+	+	−	
GDP	−	+	+	0	
Prod	−	+	+	0	
CPI	0	0	0	+	

(*Top*) $\hat{\boldsymbol{A}}_1$ and (*bottom*) $\hat{\boldsymbol{A}}_2$ from the model in Equation 3. + indicates positively significant, − indicates negatively significant, and 0 indicates not significant. Abbreviations: CPI, consumer price index; GDP, gross domestic product; Int, interest rate; Prod, total manufacturing production.

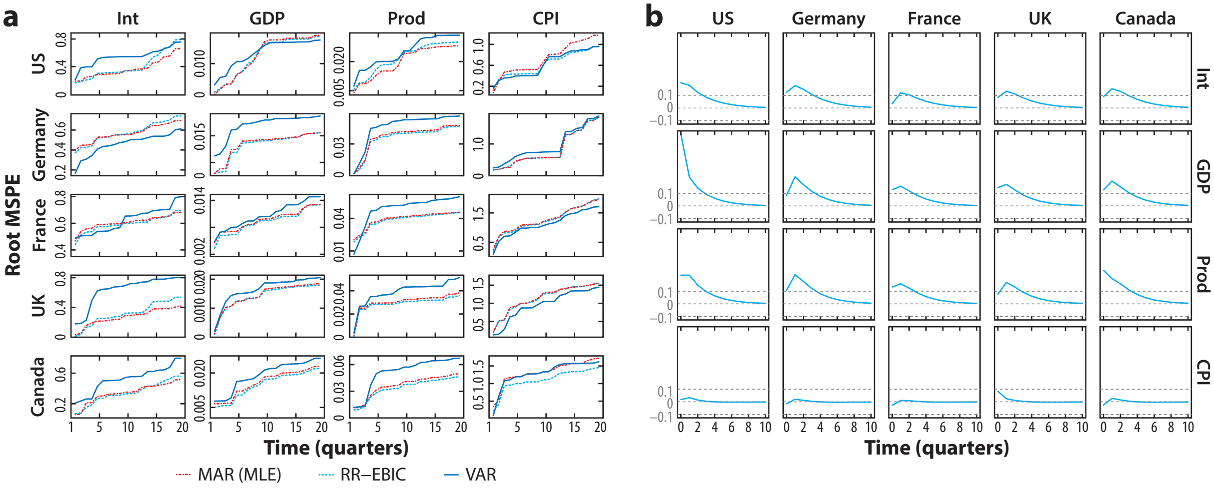

Figure 2

(*a*) Accumulated root MSPE of the VAR (*solid line*), MAR estimated by MLE (*red dashed line*), and RR models selected by EBIC (*blue dashed line*). (*b*) The impulse response functions of a unit-variance US GDP shock. The horizontal gray dashed lines mark the values (0.1, 0, −0.1) and the x-axis tick marks indicate time (0, 2, 4, 6, 8, 10) in quarters. Abbreviations: CPI, consumer price index; EBIC, extended Bayesian information criterion; GDP, gross domestic product; Int, interest rate; MAR, matrix autoregressive; MLE, maximum likelihood estimation; MSPE, mean square prediction errors; Prod, total manufacturing production; RR, reduced-rank; VAR, vector autoregressive.

GDP. The shock positively affects the GDP growth, production growth, and interest rate of all other countries, and the effect lasts about 10 quarters. In contrast, CPI is largely unaffected by the shock.

3.6. Applications of Tensor Autoregressive Models

Applications of TenAR and MAR models arise in many fields. In macroeconomics, data are often naturally grouped by economic indicators and geographic regions, leading to a matrix time series. For example, Chen et al. (2021a) analyzed quarterly economic indicators across five countries. In a similar vein, Chan & Qi (2025) modeled various macroeconomic time series across 50 US states.

In finance, assets can be categorized by many metrics. For instance, Li & Xiao (2021) treated asset returns as a three-way tensor in the TenAR model by categorizing them based on size, book-to-market ratio, and operating profitability. Yu et al. (2025a) modeled the volatility of daily log returns for credit default swaps of several financial institutions as a matrix time series. Other examples include the performance of different sectors across stock markets and the future prices of commodities over varying delivery months.

In environmental and climate studies, Yu et al. (2025b) modeled the air pollutant readings ($PM_{2.5}$, PM_{10}, and NO_2) at four stations as a matrix time series, while Hsu et al. (2021) arranged the 6-hr average east–west component of wind velocity data at 289 spatial locations in matrix form. These examples illustrate the versatility of TenAR and MAR models in capturing complex dependencies in high-dimensional time series in diverse disciplines.

4. TENSOR FACTOR MODELS

This section reviews tensor factor models as extensions of VFM in Equation 2 to tensor time series. Suppose the tensor time series $\mathcal{X}_t$ follows the decomposition $\mathcal{X}_t = \mathcal{M}_t + \mathcal{E}_t$, where the signal tensor $\mathcal{M}_t$ represents the underlying signal, and $\mathcal{E}_t$ is a white tensor noise. Chen et al. (2021b) introduced a Tucker-form tensor factor model (TFM-Tucker), modeling $\mathcal{M}_t$ using a multilinear structure akin to Tucker decompositions. Alternatively, Han et al. (2024c) proposed a CP-form tensor factor model (TFM-CP), where $\mathcal{M}_t$ follows a structure similar to CP decompositions.

The following sections provide a detailed discussion of these approaches, compare them, and explore related models and extensions.

4.1. Tensor Factor Model in Tucker Form

In this section, we present the TFM-Tucker, a flexible and widely used framework for modeling high-dimensional tensor-valued time series. This formulation extends traditional factor models by accommodating multiway structure through Tucker decomposition, enabling both dimension reduction and dynamic modeling across tensor modes.

4.1.1. The model. TFM-Tucker was initially introduced for matrix time series by Wang et al. (2019) and later extended to tensor time series by Chen et al. (2021b), incorporating new estimators and sharper error bounds. For an order-K tensor time series $\mathcal{X}_t$ of dimension $(d_1 \times \cdots \times d_K)$, the model assumes

$$\mathcal{X}_t = \mathcal{M}_t + \mathcal{E}_t = \mathcal{F}_t \times_1 \boldsymbol{A}_1 \times_2 \cdots \times_K \boldsymbol{A}_K + \mathcal{E}_t, \qquad 12.$$

where $\mathcal{F}_t$ is the core tensor of dimensions $(r_1 \times \cdots \times r_K)$ in the Tucker decomposition of $\mathcal{M}_t$, representing an unobserved latent factor process. The matrices $\boldsymbol{A}_k$ are mode-specific loading matrices of dimension $d_k \times r_k$. The noise term $\mathcal{E}_t$ is assumed to be white noise and uncorrelated with $\mathcal{F}_t$. Typically, r_k is much smaller than d_k, implying that the correlation (comovement) structure of the $d = d_1 \ldots d_K$ elements in $\mathcal{X}_t$ is controlled by a significantly smaller number of latent factors, $r = r_1 \ldots r_K$, in $\mathcal{F}_t$.

Equation 12 can be rewritten in vectorized form as

$$\mathrm{vec}(\mathcal{X}_t) = (\boldsymbol{A}_K \otimes \ldots \otimes \boldsymbol{A}_1)\mathrm{vec}(\mathcal{F}_t) + \mathrm{vec}(\mathcal{E}_t).$$

Comparing this with the VFM in Equation 2, it is evident that TFM-Tucker is a special case of VFM with a structured loading matrix. In a VFM with $r = r_1 \ldots r_K$ factors, the loading matrix $\boldsymbol{A}$ has rd parameters, whereas Equation 12 requires only $\sum_{k=1}^{K} r_k d_k$ parameters for the mode-specific loading matrices $\boldsymbol{A}_k$, $k = 1, \ldots, K$. This represents a significant dimension reduction. More importantly, TFM-Tucker explicitly leverages the tensor structure, preserving the classification of each mode. This structure provides meaningful interpretations, in contrast to the fully vectorized model, which disregards the underlying tensor organization and complicates interpretability.

For matrix time series ($K = 2$), the model simplifies to

$$\boldsymbol{X}_t = \boldsymbol{A}_1 \boldsymbol{F}_t \boldsymbol{A}_2^\top + \boldsymbol{E}_t. \qquad 13.$$

If $\boldsymbol{A}_2 = \boldsymbol{I}$, then $\boldsymbol{X}_t = \boldsymbol{A}_1 \boldsymbol{F}_t + \boldsymbol{E}_t$, so that each column of $\boldsymbol{X}_t$ follows $\boldsymbol{X}_{i,t} = \boldsymbol{A}_1 \boldsymbol{F}_{i,t} + \boldsymbol{E}_{i,t}$, which corresponds to a VFM with r_1 factors that share the same loading matrix $\boldsymbol{A}_1$, though with different factor processes. This can be interpreted as a column-wise factor structure, where $\boldsymbol{A}_1$ serves as the column loading matrix. However, this model is overly simplistic and restrictive, as it assumes the same loading matrix $\boldsymbol{A}_1$ for all columns.

A more general interpretation of Equation 13 arises from a hierarchical construction that incorporates both column-wise and row-wise factor structures. Specifically, suppose $\boldsymbol{X}_t = \boldsymbol{A}_1 \boldsymbol{G}_t + \boldsymbol{E}_t^{(c)}$, where $\boldsymbol{G}_t$ is a matrix factor process capturing column-wise dependencies. If $\boldsymbol{G}_t$ follows a row-wise factor structure, $\boldsymbol{G}_t = \boldsymbol{F}_t \boldsymbol{A}_2^\top + \boldsymbol{E}_t^{(r)}$, then substituting into the first equation yields $\boldsymbol{X}_t = \boldsymbol{A}_1 \boldsymbol{F}_t \boldsymbol{A}_2^\top + \boldsymbol{A}_1 \boldsymbol{E}_t^{(r)} + \boldsymbol{E}_t^{(c)}$. Since the combined error term can be rewritten as $\boldsymbol{E}_t = \boldsymbol{A}_1 \boldsymbol{E}_t^{(r)} + \boldsymbol{E}_t^{(c)}$, this hierarchical decomposition naturally leads back to Equation 13. This construction highlights how TFM-Tucker effectively utilizes the classification of each mode, structuring dependencies along both columns and rows rather than treating the data as a fully vectorized process.

Similar to the VFM in Equation 2, the model in Equation 12 is not identifiable due to rotational and scale ambiguities, although the projection space of each $\boldsymbol{A}_k$ is identifiable. To address this, orthonormal representatives of the projection space are selected for estimation, which leads to a canonical form given by

$$\mathcal{X}_t = \lambda \mathcal{F}_t^{(\mathrm{cano})} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \cdots \times_K \boldsymbol{U}_K, \qquad 14.$$

where the $\boldsymbol{U}_k$s are orthonormal matrices.

4.1.2. Estimation. The estimation procedures for TFM-Tucker proposed by Chen et al. (2021b) rely on the auto-moment matrix,

$$\widehat{\boldsymbol{\Sigma}}_h = \sum_{t=h+1}^{T} \frac{\mathcal{X}_{t-h} \odot \mathcal{X}_t}{T-h}. \qquad 15.$$

The use of the auto-moment matrix, rather than the second moment $\mathcal{X}_t \odot \mathcal{X}_t$, exploits the white noise assumption on $\boldsymbol{E}_t$, ensuring that $\mathbb{E}(\mathcal{X}_t \odot \mathcal{X}_{t-h}) = \mathcal{M}_t \odot \mathcal{M}_{t-h}$. This approach allows a more complex correlation structure within each $\mathcal{E}_t$.

Chen et al. (2021b) introduced two estimators: TOPUP (time series outer-product unfolding procedure) and TIPUP (time series inner-product unfolding procedure). Specifically, the TOPUP estimator for $\boldsymbol{U}_k$ in Equation 14 is given by $\widehat{U}_{k,r_k}^{\mathrm{TOPUP}}(\mathcal{X}_{1:T}) = \mathrm{LSVD}_{r_k}(\mathrm{TOPUP}_k(\mathcal{X}_{1:T}))$, where

$$\mathrm{TOPUP}_k(\mathcal{X}_{1:T}) := \left(\mathrm{mat}_k(\widehat{\boldsymbol{\Sigma}}_h), h = 1, \ldots, h_0\right).$$

Similarly, the TIPUP estimator is defined as $\widehat{U}_{k,r_k}^{\mathrm{TIPUP}}(\mathcal{X}_{1:T}) = \mathrm{LSVD}_{r_k}(\mathrm{TIPUP}_k(\mathcal{X}_{1:T}))$, where

$$\mathrm{TIPUP}_k(\mathcal{X}_{1:T}) := \left(\sum_{t=h+1}^{T} \frac{\mathrm{mat}_k(\mathcal{X}_{t-h})\mathrm{mat}_k^\top(\mathcal{X}_t)}{T-h}, h = 1, \ldots, h_0\right).$$

Here, $\mathrm{LSVD}_r(\boldsymbol{M})$ denotes the first r left singular vectors of $\boldsymbol{M}$.

The rationale behind these estimators stems from the following observation. Defining $\mathcal{F}_t^k = \mathcal{F}_t \times_{i \neq k} \boldsymbol{A}_i$ as the factors projected onto all loading matrices except for mode k, it follows that

$$\mathbb{E}[\mathrm{TOPUP}_k(\mathcal{X}_t)] = \boldsymbol{A}_k \mathbb{E}\left(\mathrm{mat}_1\left(\sum_{t=h+1}^{T} \frac{\mathrm{mat}_k(\mathcal{F}_{t-h}^k) \odot \mathrm{mat}_k(\mathcal{M}_t)}{T-h}\right), h = 1, \ldots, h_0\right)$$

and

$$\mathbb{E}[\mathrm{TIPUP}_k(\mathcal{X}_t)] = \boldsymbol{A}_k \mathbb{E}\left(\sum_{t=h+1}^{T} \frac{\mathrm{mat}_k(\mathcal{F}_{t-h}^k)\mathrm{mat}_k^\top(\mathcal{F}_t^k)}{T-h} \boldsymbol{A}_k^\top, h = 1, \ldots, h_0\right).$$

For each lag h, $\mathrm{TOPUP}_k(\mathcal{X}_t)$ constructs auto-cross-product matrices by computing inner products between all mode-k fibers of $\mathcal{X}_{t-h}$ and all elements of $\mathcal{X}_t$, followed by applying Hermitian squares before summation. This process results in a matrix of size $d_k \times d_{-k} d h_0$ where $d_{-k} = d/d_k$. In contrast, $\mathrm{TIPUP}_k(\mathcal{X}_t)$ first computes inner products between all mode-k fibers, then takes the sum, and then applies Hermitian squares, producing a matrix of size $d_k \times h_0 d_k$.

A key distinction between the two estimators lies in the order of summation and squaring. TIPUP, by summing before squaring, effectively reduces the noise level, but this early aggregation can lead to partial cancellation of the signal. Conversely, TOPUP preserves all signal components, but since it squares before summation, it does not benefit from the same noise reduction. As shown by Chen et al. (2021b), when signal cancellation is minimal or absent, TIPUP achieves a faster convergence rate than TOPUP.

The TOPUP and TIPUP estimators are noniterative and computationally efficient; however, their accuracy can be improved through iterative refinement. To address this, Han et al. (2024a) introduced iterative versions, iTOPUP and iTIPUP, which enhance estimation precision by updating the loading matrices through projection-based iterations.

Given initial estimates, the procedure updates the orthonormal matrix $\boldsymbol{U}_k$ at iteration j by first computing the projected tensor

$$\mathcal{Z}_{k,t}^{(j)} = \mathcal{X}_t \times_1 \widehat{\boldsymbol{U}}_1^{(j)\top} \times_2 \cdots \times_{k-1} \widehat{\boldsymbol{U}}_{k-1}^{(j)\top} \times_{k+1} \widehat{\boldsymbol{U}}_{k+1}^{(j-1)\top} \cdots \times_K \widehat{\boldsymbol{U}}_K^{(j-1)\top}.$$

The updated estimate of $\boldsymbol{U}_k$ is obtained by applying either TOPUP or TIPUP to the projected data $\mathcal{Z}_{k,t}^{(j)}$. This iterative approach refines the factor estimates while ensuring computational efficiency. The process continues iterating until convergence. The procedure leverages the fact that the projected data $\mathcal{Z}_{k,t}^{(j)}$ are a lower-dimensional tensor with dimensions $r_1 \times \cdots \times r_{k-1} \times d_k \times r_{k+1} \times \cdots \times r_K$, compared with the full dimensions $d_1 \times \cdots \times d_K$ of $\mathcal{X}_t$. Han et al. (2024a) demonstrate that, with proper initialization, iTOPUP and iTIPUP outperform their noniterative counterparts, achieving faster convergence rates.

Several alternative estimation procedures have been proposed for TFM-Tucker; these were primarily developed for matrix time series but are easily extendable to tensor data. Specifically, Chen & Fan (2021) introduced an α-PCA method, which relies on eigenanalysis of a weighted average of the sample mean and covariance matrices. Yu et al. (2022) proposed a projected estimation method that improves the efficiency of factor loading matrix estimation. Building on this, He et al. (2023a) interpreted the projected estimator within a least squares framework, analogous to PCA in VFMs. This framework was further extended to minimize the Huber loss function, leading to a weighted iterative projection method for robust parameter estimation in the presence of heavy-tailed errors.

Zhang et al. (2024) proposed an estimation method for $\boldsymbol{A}_k$ $(k = 1, \ldots, K)$ in Equation 12 using mode-wise PCA. This method relies on the mode cross-product of $\mathrm{mat}_k(\mathcal{Y})\mathrm{mat}_k(\mathcal{Y})^\top$, where $\mathcal{Y} = (\mathcal{X}_1, \ldots, \mathcal{X}_T)$ is the order-$(K+1)$ tensor. The approach is similar to the TOPUP and TIPUP procedures.

Several studies have addressed weak serial and cross-sectional correlations in noise processes. Barigozzi et al. (2023b) developed a projection estimator for TFM-Tucker, improving signal-to-noise ratios (SNRs) through dimensionality reduction and eigenvalue ratio testing. Chen & Lam (2024b) introduced a preaveraging procedure for tensor time series data, utilizing random projections to improve factor loading estimation by identifying strong factor directions.

Robust factor modeling in tensor time series has received attention. Barigozzi et al. (2023a), He et al. (2023b), and Y. Wang et al. (2024) applied Huber loss for tail-robust estimation, while Barigozzi et al. (2025b) proposed a projection-based iterative procedure for Tucker decomposition, ensuring robust estimators under heavy-tailed distributions.

Bayesian approaches have also gained traction. Zhang (2024) introduced Bayesian dynamic factor models for high-dimensional matrix-valued time series, facilitating probabilistic factor estimation and uncertainty quantification.

Taking advantage of the ambiguity in the TFM-Tucker, Chen et al. (2021b) applied VARIMAX rotations to each estimated loading matrix in order to enhance sparsity and interoperability of the results.

4.1.3. Rank determination. The determination of the dimensions of $\mathcal{F}_t$ in Equation 12 is crucial for effectively constructing a TFM-Tucker in practice. This dimension is often referred to as the Tucker rank of the decomposition. The procedure for determining the Tucker rank, proposed by Han et al. (2022), involves analyzing a symmetric, semipositive definite matrix $\boldsymbol{W}$ of size $p \times p$ with rank r, whose eigenvalues satisfy $\lambda_1 \geqslant \cdots \geqslant \lambda_r > \lambda_{r+1} = \cdots = \lambda_p = 0$. Given a noisy estimate $\widehat{\boldsymbol{W}}$ of $\boldsymbol{W}$ and its eigenvalues $\widehat{\lambda}_1 > \cdots > \widehat{\lambda}_p$, Han et al. (2022) proposed two rank estimators based on the IC and eigenratio criterion (ER):

$$\mathrm{IC}(\widehat{\boldsymbol{W}}) = \arg \min_{0 \leqslant m \leqslant m^*} \left\{ \sum_{l=m+1}^{p} \widehat{\lambda}_l + mG(\widehat{\boldsymbol{W}}) \right\}, \text{ and } \mathrm{ER}(\widehat{\boldsymbol{W}}) = \arg \min_{1 \leqslant m \leqslant m^*} \frac{\widehat{\lambda}_{m+1} + H(\widehat{\boldsymbol{W}})}{\widehat{\lambda}_m + H(\widehat{\boldsymbol{W}})},$$

where m^* represents an upper bound on the rank.

The IC balances model fit (since $\sum_{l=m+1}^{p} \widehat{\lambda}_l$ can be viewed as the sum of squares of residuals) and a complexity penalty function $G(\widehat{\boldsymbol{W}})$. The ER, originally proposed by Ahn & Horenstein (2013) and Lam & Yao (2012) for rank determination in VFM, is based on the observation that $\lambda_r/\lambda_{r+1} = \infty$ if r is the true rank. The function $H(\widehat{\boldsymbol{W}})$ is employed to enhance the stability of the estimator. For $G(\cdot)$, with the tuning parameter ν, Han et al. (2022) used $\frac{h_0 d^{2-2\nu}}{T} \log(\frac{dT}{d+T})$, $h_0 d^{2-2\nu} \left(\frac{1}{T} + \frac{1}{d}\right) \log\left(\frac{dT}{d+T}\right)$, $\frac{h_0 d^{2-2\nu}}{T} \log(\min\{d, T\})$, $h_0 d^{2-2\nu} \left(\frac{1}{T} + \frac{1}{d}\right) \log(\min\{d, T\})$, and $h_0 d^{2-2\nu} \left(\frac{1}{T} + \frac{1}{d}\right) \log(\min\{d_k, T\})$. For $H(\cdot)$, they use $c_0 h_0$, $\frac{h_0 d^2}{T^2}$, $\frac{h_0 d^2}{T^2 d_k^2}$, $\frac{h_0 d^2}{T^2 d_k^2} + \frac{h_0 d_k^2}{T^2}$, and $\frac{h_0 d^2}{T^2 d_k} + \frac{h_0 d d_k}{T^2}$.

For determining the TFM-Tucker, Han et al. (2022) employ $\widehat{\boldsymbol{W}}$ in the form of $\boldsymbol{\Omega}\boldsymbol{\Omega}^\top$, where $\boldsymbol{\Omega}$ is defined as $\mathrm{TOPUP}_k(\mathcal{X}_{1:T})$ and $\mathrm{TIPUP}_k(\mathcal{X}_{1:T})$ for the TOPUP and TIPUP procedures, respectively. Within the iterative processes of iTOPUP and iTIPUP, $\boldsymbol{\Omega}$ is replaced by $\mathrm{TOPUP}_k(\widetilde{\mathcal{Z}}^{(j)}_{1:T,k})$ and $\mathrm{TIPUP}_k(\widetilde{\mathcal{Z}}^{(j)}_{1:T,k})$, respectively. The authors demonstrate that these estimators are consistent under certain conditions.

4.2. Tensor Factor Models in Canonical Polyadic Form

This section introduces the TFM-CP, an alternative to the Tucker-based approach that expresses a tensor as a sum of rank-1 components. The CP representation offers a more parsimonious structure and is particularly appealing for interpretability and identifiability in certain applications.

4.2.1. The model. The TFM-CP was introduced by Han et al. (2024c) and is expressed in the form

$$\mathcal{X}_t = \mathcal{M}_t + \mathcal{E}_t = \sum_{j=1}^{r} f_{jt} \boldsymbol{a}_{j1} \odot \boldsymbol{a}_{j2} \odot \cdots \odot \boldsymbol{a}_{jK} + \mathcal{E}_t, \qquad 16.$$

where $\boldsymbol{a}_{jk}$ are unit vectors of dimension d_k, f_{jt} are uncorrelated with $\mathcal{E}_{t-h}$, and $f_{j',t-h}$ at all lags h. The noise term $\mathcal{E}_t$ represents white noise. This formulation follows the CP decomposition of the signal tensor $\mathcal{M}_t$.

Note that the TFM-CP is a special case of the VFM, with the same number of factors, since under the model described in Equation 16, $\mathrm{vec}(\mathcal{X}_t) = \sum_{j=1}^{r} f_{jt} \mathrm{vec}(\boldsymbol{a}_{j1} \odot \boldsymbol{a}_{j2} \odot \cdots \odot \boldsymbol{a}_{jK}) + \mathrm{vec}(\mathcal{E}_t)$. The low-rank structure of TFM-CP improves estimation efficiency by reducing the number of free parameters compared with an unstructured VFM. This facilitates factor identification, reduces estimation variance, and preserves the multiway dependencies of the tensor, enhancing interpretability. Additionally, the structured decomposition allows for scalable computation, making it particularly suited for high-dimensional time series analysis.

The signal tensor $\mathcal{M}_t$ in the TFM-CP is a linear combination of r fixed rank-1 tensors, with the factors as the random (latent) combination weights. It enhances the interpretation of the factors. The rank-1 tensors are also easier to interpret. For example, for $K = 2$ (the matrix case), if $\boldsymbol{X}_t$ is a transport network where $\boldsymbol{X}_{ijt}$ is the number of people traveling from zone i to zone j at time t, and all elements of $\boldsymbol{a}_{jk}$ are positive, then one can normalize the model so that each $\boldsymbol{a}_{jk}$ sums to one (as proportions), and $\boldsymbol{a}_{j1}\boldsymbol{a}_{j2}$ is a probability contingency table, showing the traffic pattern (the proportion of people moving from zone i to zone j). In this case, the expectation of the sum of all elements in $\boldsymbol{X}_t$ (total number of people traveling at time t) equals the sum of the factors $f_{1t} + \ldots + f_{rt}$. Hence, f_{jt} can be interpreted as the number of people using traffic pattern j at time t.

One of the key assumptions of Han et al. (2024c) is that the factors f_{jt} are uncorrelated. However, B. Chen et al. (2024b) recently relaxed this assumption, allowing for correlated factors, thereby increasing the model's flexibility and applicability to a broader range of real-world scenarios.

4.2.2. Estimation. For estimation, both Han et al. (2024c) and B. Chen et al. (2024b) use an iterative procedure based on orthogonal projections with a special initial estimation. Define the cross-covariance matrix as

$$\boldsymbol{\Sigma}_h = \mathbb{E}\left[\mathcal{X}_t \odot \mathcal{X}_{t-h}\right] = \sum_{i,j=1}^{r} \theta_{ij}(h) \odot_{l=K}^{1} \boldsymbol{a}_{il} \odot_{l=K}^{1} \boldsymbol{a}_{jl} + \mathbb{E}\left[\mathcal{E}_{t-h} \odot \mathcal{E}_t\right], \qquad 17.$$

where $\theta_{ij}(h) := \mathbb{E}(f_{i,t-h} f_{jt})$.

Han et al. (2024c) assume the noise term $\boldsymbol{E}_t$ to be white, and the factor processes f_{it} are uncorrelated across all lags. For $h > 0$, $\boldsymbol{\Sigma}_h$ reduces to $\sum_i^r \theta_{ii}(h) \odot_{l=1}^{K} \boldsymbol{a}_{il} \odot_{l=1}^{K} \boldsymbol{a}_{il}$. The initial estimation of $\boldsymbol{a}_{ik}$ (referred to as composite PCA) is based on the first r singular vectors of $(\text{mat}_k(\widehat{\boldsymbol{\Sigma}}_h) + \text{mat}_k(\widehat{\boldsymbol{\Sigma}}_h)^\top)$, $k = 1, \ldots, K$, where $\widehat{\boldsymbol{\Sigma}}_h$ is the sample estimate of $\boldsymbol{\Sigma}_h$.

In contrast, B. Chen et al. (2024b) relax the assumption of uncorrelated factors, focusing instead on the contemporaneous cross-product matrix $\boldsymbol{\Sigma}_0$. The initialization is performed using a randomized composite PCA method, which incorporates randomized projections alongside composite PCA applied to the sample estimate of $\boldsymbol{\Sigma}_0$.

After initialization, both methods refine the estimates using iterative orthogonal projections, based on the following observation. Let

$$\boldsymbol{A}_k = (\boldsymbol{a}_{1k}, \ldots, \boldsymbol{a}_{rk}) \ \text{ and } \ \boldsymbol{B}_k = \boldsymbol{A}_k(\boldsymbol{A}_k^\top \boldsymbol{A}_k)^{-1} = (\boldsymbol{b}_{1k} \ldots, \boldsymbol{b}_{rk}),$$

where $\boldsymbol{A}_k$ contains the mode vectors for the kth mode and $\boldsymbol{B}_k$ represents the orthogonal projection of $\boldsymbol{A}_k$.

Next, define the updated factor tensor as

$$\boldsymbol{z}_{t,ik} = \mathcal{X}_t \times_1 \boldsymbol{b}_{i1}^\top \times_2 \cdots \times_{k-1} \boldsymbol{b}_{i,k-1}^\top \times_{k+1} \boldsymbol{b}_{i,k+1}^\top \times_{k+2} \cdots \times_K \widehat{\boldsymbol{b}}_{iK}^\top = f_{it}\boldsymbol{a}_{ik} + \mathcal{E}_{t,ik}^*,$$

where $\boldsymbol{z}_{t,ik}$ is a factorized version of $\mathcal{X}_t$, and $\mathcal{E}_{t,ik}^*$ represents the residual noise. This equation suggests that $\boldsymbol{z}_{t,ik}$ follows a VFM with a single factor.

Using the previously estimated $\boldsymbol{a}$ values, $\boldsymbol{a}_{ik}$ can be estimated using the estimated $\boldsymbol{z}_{t,ik}, t = 1, \ldots, T$, according to Lam & Yao (2012); using lag covariance matrices as done by Han et al. (2024c); or through the contemporaneous variance matrix, according to Bai & Ng (2002, 2013).

Chang et al. (2023) proposed an alternative estimation approach for the TFM-CP in the case of matrix time series ($K = 2$), leveraging the generalized eigenanalysis method (Golub & Van Loan 2013, Wilkinson 1965). This method eliminates the need for iterative procedures while maintaining strong theoretical properties. Chang et al. (2025a) further refined this approach, achieving faster convergence under weaker conditions.

4.2.3. Determination of the number of factors in tensor factor models in canonical polyadic form. To determine the number of factors in TFM-CP, B. Chen et al. (2024b) propose two approaches based on the eigenratio method. The first approach applies the ER to the vectorized covariance matrix, $\widehat{\boldsymbol{\Sigma}} = T^{-1}\sum_{t=1}^{T} \mathrm{vec}(\mathcal{X}_t)\mathrm{vec}(\mathcal{X}_t)^\top$. The second approach evaluates the mode-wise cross-moment matrices, $\widehat{\boldsymbol{\Sigma}}_{(k)} = \frac{1}{T}\sum_{t=1}^{T} \mathrm{mat}(\mathcal{X}_t)\mathrm{mat}(\mathcal{X}_t)^\top$, $k = 1, \ldots, K$, and selects the maximum rank across all K modes.

The vectorized covariance matrix method captures global factor information, as TFM-CP is a special case of VFM with the same rank, as discussed earlier. However, it can be computationally intractable for large tensors due to the high dimensionality of $\widehat{\boldsymbol{\Sigma}}$. In contrast, the mode-wise approach leverages tensor-specific structures but may underestimate the number of factors since it requires at least one of the matrices $[\boldsymbol{a}_1 : \boldsymbol{a}_2 : \ldots : \boldsymbol{a}_r]$ to be full rank. B. Chen et al. (2024b) provide theoretical guarantees for both methods under different conditions. The choice between them depends on the data structure and computational feasibility. In practice, combining both approaches may yield the most reliable factor number determination.

4.3. Comparison of Tensor Factor Models in Tucker and Canonical Polyadic Form

The Tucker and CP formulations of tensor factor models offer distinct approaches to modeling tensor time series. TFM-Tucker provides flexibility by capturing complex factor interactions through its core tensor but suffers from rotational ambiguity, making interpretation challenging. In contrast, TFM-CP ensures a unique factorization under mild conditions, leading to a more straightforward interpretation of the factors. However, its estimation involves a nonconvex optimization problem, requiring iterative procedures and high-quality initialization for reliable results.

Typically, TFM-CP requires fewer factors than does TFM-Tucker. This follows from the fact that both models are special cases of the VFM, expressed as $\mathrm{vec}(\mathcal{X}_t) = \boldsymbol{A}\boldsymbol{f}_t + \mathrm{vec}(\mathcal{E}_t)$. In the case of TFM-Tucker, the loading matrix takes the Kronecker product form $\boldsymbol{A} = \boldsymbol{A}_K \otimes \cdots \otimes \boldsymbol{A}_1$, with the factor vector given by $\boldsymbol{f}_t = \mathrm{vec}(\mathcal{F}_t)$. For TFM-CP, each column of $\boldsymbol{A}$ is structured as $\mathrm{vec}(\boldsymbol{u}_{jK} \otimes \boldsymbol{u}_{j,K-1} \otimes \cdots \otimes \boldsymbol{u}_{j1})$, and the factor vector is $\boldsymbol{f}_t = (f_{1t}, \ldots, f_{rt})^\top$. Since $\mathcal{F}_t$ in TFM-Tucker must retain a tensor structure, its vectorized form $\mathrm{vec}(\mathcal{F}_t)$ generally has a larger dimension than r, implying that TFM-Tucker typically requires more factors than TFM-CP when both models provide equivalent representations.

Furthermore, TFM-CP can be rewritten as a special case of TFM-Tucker in the form $\boldsymbol{A}_k = (\boldsymbol{a}_{1k}, \ldots, \boldsymbol{a}_{rk})$ and $\mathcal{F}_t = \mathrm{diag}(f_{1t}, \ldots, f_{rt})$. In this representation, $\mathcal{F}_t$ is constrained to be diagonal, and the columns of $\boldsymbol{A}_k$ are not necessarily orthogonal. In contrast, the general TFM-Tucker imposes orthogonality on the columns of $\boldsymbol{A}_k$ and allows $\mathcal{F}_t$ to have an unrestricted structure.

Table 2 provides a theoretical comparison of estimators for the two models. Here, ψ_0 denotes the error bound for the TIPUP initialization estimator in TFM-Tucker and the composite PCA estimator in TFM-CP, while ψ_{ideal} represents the corresponding error bound for their iterative versions.

Table 2 Comparison of TFM-Tucker and TFM-CP

Parameter	TFM-Tucker	TFM-CP
Number of iterations	$O(\log(\psi_0/\psi_{\mathrm{ideal}}))$	$O(\log\log(\psi_0/\psi_{\mathrm{ideal}}))$
ψ_{ideal}	$\max_k\left(\frac{\sigma\sqrt{d_k}}{\lambda\sqrt{T}} + \frac{\sigma^2\sqrt{d_k}}{\lambda^2\sqrt{T}}\right)$	$\max_k\left(\frac{\sigma\sqrt{d_k}}{w_r\sqrt{T}} + \frac{\sigma^2\sqrt{d_k}}{w_r^2\sqrt{T}}\right)$
Statistical optimality	✓	✓
Computational optimality	✓	unknown

Abbreviations: TFM-CP, canonical polyadic–form tensor factor model; TFM-Tucker, Tucker-form tensor factor model.

4.4. Extension of Tensor Factor Models and Other Related Models

This section presents several extensions of tensor factor models that address more complex data structures and modeling goals. These include the incorporation of structural constraints and covariates (Section 4.4.1), adaptations to nonstationary and nonlinear dynamics (Section 4.4.2), applications to spatial and network time series (Section 4.4.3), and connections to non–time series tensor modeling approaches (Section 4.4.4).

4.4.1. Additional structural features and covariates. Extensions to tensor factor models have been explored to incorporate additional structural properties and covariate inclusion. Cen & Lam (2025a) proposed a test for detecting a Kronecker product structure in the factor loading matrix, a key feature of TFM-Tucker that characterizes its common component structure. Chen et al. (2020) introduced a constrained framework that integrates covariates as side information, enabling further dimensionality reduction. Additionally, E.Y. Chen et al. (2024) developed a semiparametric tensor factor analysis approach, incorporating auxiliary covariates into the loading matrices. This enhancement improves the model's ability to yield more accurate estimators and enables predictions with new covariates.

4.4.2. Nonstationary and nonlinear extensions. Chen et al. (2025) extended classical inference methods to high-dimensional settings by integrating common stochastic trends and multifactor error structures into matrix-valued time series. Liu & Chen (2022) proposed threshold matrix-variate factor models to capture regime-switching behavior in factor loadings. He et al. (2024a) developed a procedure specifically for online detection of changepoints in MFMs. Additionally, time-varying extensions, such as those introduced by B. Chen et al. (2024a), presented smoothly evolving MFMs, which are especially relevant for economic and financial data exhibiting structural shifts.

4.4.3. Spatial and network time series. Tensor time series are closely related to spatiotemporal models and dynamic networks. Barigozzi et al. (2025c) developed the general spatiotemporal factor model, which decomposes high-dimensional spatiotemporal data into latent factor-driven components, utilizing frequency-domain Fourier analysis. While Chen et al. (2021b) applied mode-based spatial components, their approach lacks an embedded spatial lag structure. Network-based factor models, such as those by Wang et al. (2020), reformulate VAR models into tensor representations, imposing parameter restrictions across multiple dimensions.

Lee & Zhang (2024) studied the TFM-Tucker assuming errors follow a martingale difference sequence. They proposed a metric to identify subspaces that preserve conditional mean information without making parametric assumptions. As their optimization targets conditional mean, the estimation enhances predictions, though prediction remains optional in their framework.

Factor strength estimation continues to be a crucial area of research. Chen & Lam (2024a) developed statistical tests to assess the significance of factors in vector and matrix time series models, building on the work of Gao & Tsay (2023), who introduced a two-way transformed factor model to capture dynamics in both row- and column-wise factors. Additionally, He et al. (2023a) explored the suitability of one-way versus two-way factor models in this context.

4.4.4. Non–time series approaches. Time series analysis aims to discover the dynamic evolution over time and build models with predictability. However, one can ignore the specialty of time and treat the order-K tensor time series $\mathcal{X}_t$ as an order-$(K+1)$ tensor observation $\mathcal{Y}$ by including the time dimension as an extra mode. A low-rank Tucker decomposition of $\mathcal{Y}$ can be expressed as

$$\mathcal{Y} = \mathcal{G} \times_{k=1}^{K+1} \boldsymbol{A}_k + \mathcal{E}, \qquad 18.$$

where $\mathcal{G}$ is the core tensor, $\boldsymbol{A}_k$ are the factor matrices, and $\mathcal{E}$ represents the error term.

Babii et al. (2025) treated $\boldsymbol{A}_{K+1}$ (the loading matrix corresponding to the time mode) as the factor matrix, with $\mathcal{G} \times_{k=1}^{K} \boldsymbol{A}_k$ representing the loading tensor. They introduced two tensor PCA algorithms for estimating the model in Equation 18. Their analysis showed that under strong factor settings, the simple one-step algorithm is optimal. However, when factors are weak, an iterative least squares algorithm is required to improve convergence rates.

Earlier, Babii et al. (2023) analyzed the CP decomposition of $\boldsymbol{Y}$ with orthogonal loadings. This is a special case within the generalized framework proposed by Babii et al. (2025), when the core tensor, $\mathcal{G} \in \mathbb{R}^{r_1 \times \cdots \times r_{K+1}}$, is diagonal, with $r_1 = r_2 = \cdots = r_{K+1}$.

4.4.5. Separable factor analysis. The separable factor analysis (SFA) model proposed by Fosdick & Hoff (2014) extends the model of Hoff et al. (2011). Let $\mathcal{Z} \in \mathbb{R}^{m_1 \times \cdots \times m_K}$, and suppose that $\mathrm{vec}(\mathcal{Z}) \sim \mathrm{Normal}(0, \mathrm{Cov}[\mathrm{vec}(\mathcal{Z})])$. SFA assumes that the covariance matrix is given by

$$\mathrm{Cov}[\mathrm{vec}(\mathcal{Z})] = \boldsymbol{\Sigma}_K \otimes \boldsymbol{\Sigma}_{K-1} \otimes \cdots \otimes \boldsymbol{\Sigma}_1. \qquad 19.$$

Under this assumption, each $\boldsymbol{\Sigma}_i$ can be interpreted as the covariance matrix of the ith mode fibers of the tensor. Specifically, define the standardized version of $\mathcal{Z}$ with respect to all modes except i as $\mathcal{Z}^i := \mathcal{Z} \times_{k \neq i} \boldsymbol{\Sigma}_k^{-1/2}$. The covariance of the ith mode fibers of $\mathcal{Z}^i$ is exactly $\boldsymbol{\Sigma}_i$ under the model in Equation 19.

Furthermore, Fosdick & Hoff (2014) assume that each covariance matrix $\boldsymbol{\Sigma}_i$ takes the form $\boldsymbol{\Sigma}_i = \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^T + \boldsymbol{D}_i^2$, where $\boldsymbol{\Lambda}_i \in \mathbb{R}^{m_i \times k_i}$ and $\boldsymbol{D}_i$ is a diagonal matrix with $0 \leqslant k_i < m_i$. This structure can be seen as the covariance matrix of a factor model $\mathrm{mat}_i(\mathcal{Z}^i) = \boldsymbol{\Lambda}_i \mathrm{mat}_i(\mathcal{F}^i) + \boldsymbol{D}_i \mathrm{mat}_i(\mathcal{E}^i)$, suggesting that the parameters $\{\boldsymbol{\Lambda}_i, \boldsymbol{D}_i\}$ represent the single-mode factor analysis parameters for the ith mode of the tensor when the covariance in all other modes has been removed.

Linton & Tang (2022a) compare the SFA approach with TFM-Tucker. They note that, assuming $\mathbb{E}[\mathrm{vec}(\mathcal{F}_t)[\mathrm{vec}(\mathcal{F}_t)]^\top] = \boldsymbol{I}_r$ and $\mathbb{E}[\mathrm{vec}(\mathcal{E}_t)[\mathrm{vec}(\mathcal{E}_t)]^\top] = \sigma^2 \boldsymbol{I}_d$, the TFM-Tucker simplifies to

$$\mathbb{E}[\mathrm{vec}(\mathcal{X}_t)[\mathrm{vec}(\mathcal{X}_t)]^\top] = \boldsymbol{A}_K \boldsymbol{A}_K^\top \otimes \boldsymbol{A}_{K-1} \boldsymbol{A}_{K-1}^\top \otimes \ldots \otimes \boldsymbol{A}_1 \boldsymbol{A}_1^\top + \sigma^2 \boldsymbol{I}_d,$$

which follows the framework of SFA, with $\boldsymbol{D}_i = \boldsymbol{I}_{d_i}$. This framework is also examined by Linton & Tang (2022b) without the $\sigma^2 \boldsymbol{I}_d$ term, assuming that d_k is fixed and K approaches infinity.

4.5. An Illustrative Example

Chen et al. (2021b) applied the TFM-Tucker to the monthly import–export trading volume series introduced in Example 2 of Section 1. To illustrate the TFM-Tucker and TFM-CPs, we consider a five-year monthly series (2012–2016) of total trading volumes across 22 countries, aggregated over all product categories. The resulting data set forms a matrix time series with 22×22 matrices observed monthly.

For the MFM-Tucker model in Equation 13, the factor dimensions are determined using the rank selection procedure of Han et al. (2022) (see Section 4.2.3), resulting in a 4×4 core tensor. Model estimation is performed via the iTIPUP algorithm, followed by VARIMAX rotation of each loading matrix to enhance interpretability.

Figure 3*a* shows the estimated core factor series $\boldsymbol{F}_t$ (with different scales). **Figure 3*b*** presents a hub interpretation of the MFM-Tucker structure, following the approach of Chen et al. (2021b). Specifically, the import–export matrix time series can be interpreted as a dynamic transport network, a network with directional edges with weights corresponding to the transport volumes. The low-dimensional 4×4 core factor $\boldsymbol{F}_t$ process can also be viewed as a dynamic transport network, or as a virtual trade flow network, quantifying the trading volumes between four latent export hubs and four latent import hubs. This structure implies that the observed export from country

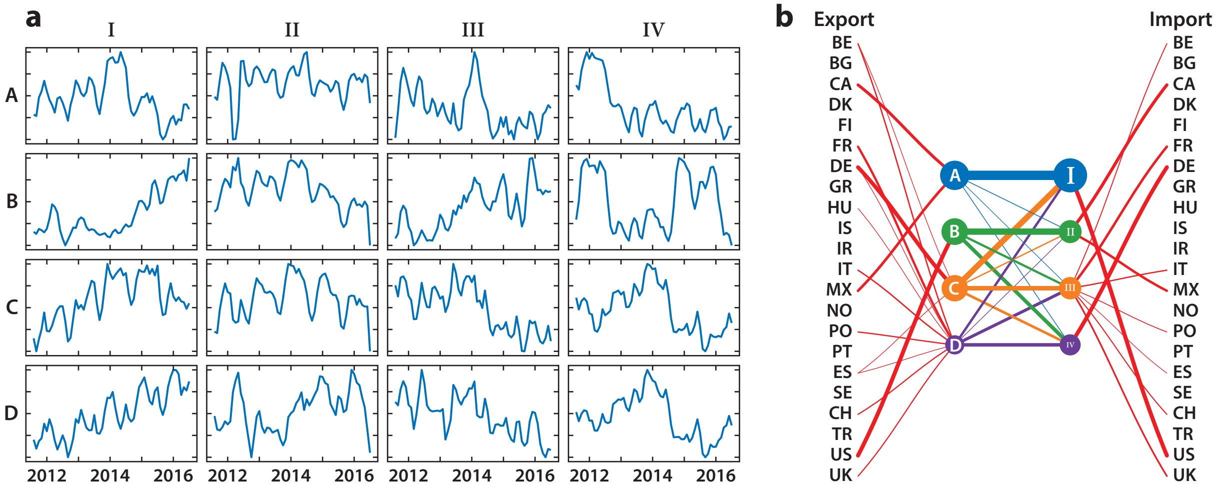

Figure 3

Estimated MFM-Tucker model for the import–export matrix time series example. (*a*) Estimated 4×4 core factor processes (with different scales). (*b*) A graph representation of the hub interpretation of the MFM-Tucker model. Exporting countries are shown on the left and importing countries on the right, illustrating directional trade flows captured by the model. Abbreviation: MFM, matrix factor model.

i to country j can be conceptualized as being routed via an export hub to an import hub before reaching its destination.

The columns of the loading matrix $\boldsymbol{A}_1$ characterize the contributions of each country to one of the export hubs, while the columns of $\boldsymbol{A}_2$ represent how each country receives trade from the four import hubs. In the network representation in **Figure 3*b***, the thickness of edges from individual countries to export hubs A, B, C, and D is proportional to the elements of $\boldsymbol{A}_1$, omitting contributions smaller than 0.15 for clarity. Similarly, the links from import hubs I, II, III, and IV to the receiving countries reflect the magnitudes of $\boldsymbol{A}_2$.

This representation reveals interpretable geographic and economic patterns. For instance, export hub B is predominantly associated with the United States, hub A with Canada and Mexico, and hub D with European countries. On the import side, hub I is primarily linked to the United States. The size of the dot representing each export hub reflects the total trade volume routed through it over the five years, calculated as the cumulative sum of the corresponding elements in the core factor series $\boldsymbol{F}_t$. For instance, the total volume associated with export hub A is given by the sum of $\boldsymbol{F}_{t,11} + \boldsymbol{F}_{t,12} + \boldsymbol{F}_{t,13} + \boldsymbol{F}_{t,14}$ over time. Similarly, the edge thickness from hub A to hub I reflects the total trading volume from hub A to hub I during the five years, as the sum of $\boldsymbol{F}_{t,11}$ over time. The remainder of the diagram may be interpreted similarly. Notably, most exports from Canada and Mexico are channeled through export hub A to import hub I, ultimately reaching the United States.

We also fit the TFM-CP in Equation 16 (Han et al. 2024c) to the same data set. The selected rank is 4. One of the interpretations of the model is that the global trading network mainly consists of the four subnetworks, each represented by a rank-1 matrix $\boldsymbol{a}_{i1}\boldsymbol{a}_{i2}^\top$ and associated with a time-varying factor f_{it} denoting the aggregate trading volume within subnetwork i at time t.

Figure 4 presents the four estimated subnetworks, constructed using $\widehat{\boldsymbol{a}}_{i1}\widehat{\boldsymbol{a}}_{i2}^\top$, with entries below 0.3 omitted for better clarity and interpretability, along with their respective factor processes f_{it}. The subnetwork in **Figure 4*a*** mainly captures the exports from Canada, Germany, and Mexico to the United States, while the subnetwork in **Figure 4*b*** reflects exports from several European

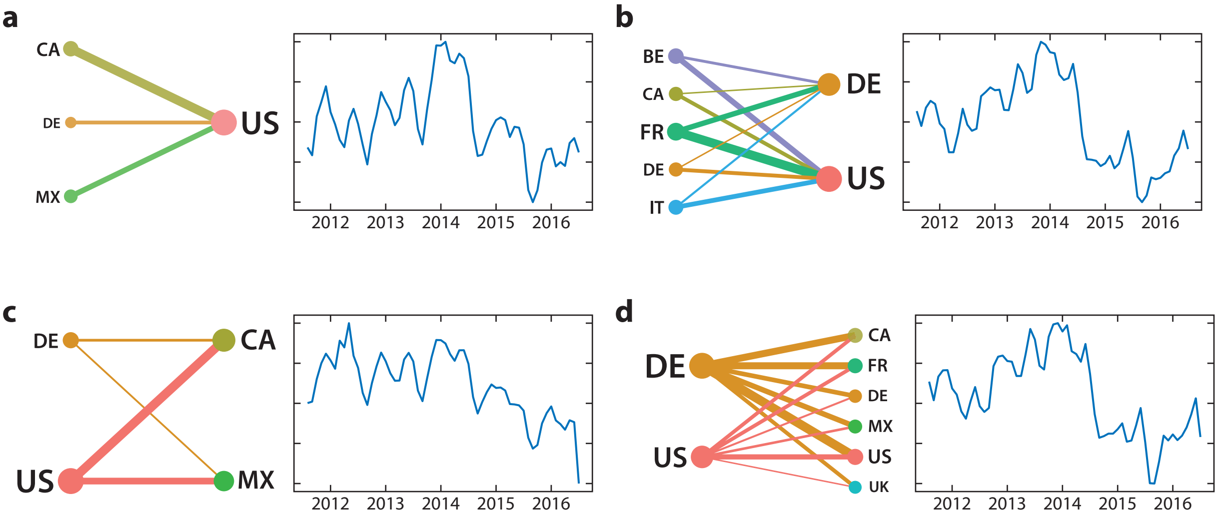

Figure 4

Estimated subnetworks and their corresponding estimated factor processes in a TFM-CP for the import–export matrix time series example. Abbreviation: TFM-CP, canonical polyadic–form tensor factor model.

countries to Germany and the United States. The temporal component f_{it} summarizes the total volume of trade occurring within subnetwork i over time.

4.6. Applications

Tensor factor models have been widely applied in the analysis of high-dimensional time series across various domains. In macroeconomics and finance, Chen & Chen (2022), Chen et al. (2021b), Barigozzi et al. (2023a), Zhang et al. (2024), and Barigozzi et al. (2023b) studied import/export data of a variety of commodities across multiple countries, and He et al. (2023b) and Y. Wang et al. (2024) considered financial portfolios and macroeconomic indices. The Fama–French 10×10 return series, initially analyzed by Wang et al. (2019), has been further investigated by Cen & Lam (2025b), Chang et al. (2025a), and Chang et al. (2023). Applications to macroeconomic networks include the analysis of US quarterly macroeconomic indicators (Wang et al. 2020) and factor-augmented network autoregressions (Barigozzi et al. 2025a). Wang et al. (2020) investigated US quarterly macroeconomic sequences, and Barigozzi et al. (2025a) examined macroeconomic networks.

Tensor-based approaches have also been applied to demographic studies, such as the analysis of human mortality data by Fosdick & Hoff (2014). In the transportation sector, tensor methods have been used to analyze New York City taxi data (Cen & Lam 2025b, Chen et al. 2021b, Han et al. 2024c, Lee & Zhang 2024). Lee & Zhang (2024) also investigate sales data. In energy forecasting, Banin et al. (2025) employed a three-mode tensor structure to model seasonal variations in energy demand. Furthermore, Yuan et al. (2023) applied tensor factor models to analyze air quality data.

Tensor factor models have extended their reach into machine learning and image analysis. For instance, Zhang et al. (2024) studied the Moving MNIST (Modified National Institute of Standards and Technology) data set, where tensor observations exhibit no serial dependence. In multidimensional panels, Babii et al. (2025) applied tensor decompositions to analyze firm characteristics, while Cen & Lam (2025b) utilized Tucker decompositions to model Organisation for Economic Co-operation and Development economic data and transportation networks.

These diverse applications highlight the increasing significance of tensor factor models in capturing complex dependencies in high-dimensional time series, offering a powerful framework for dimensionality reduction, latent factor estimation, and uncovering hidden patterns across various domains.

5. DYNAMIC TENSOR FACTOR MODELS

Although the TenAR models significantly reduce the dimensionality and complexity of VAR models by effectively utilizing the natural tensor structure and its mode classifications, they may still involve a large number of parameters, requiring a substantial sample size for accurate estimation. In contrast, the TFM effectively reduces dimensionality but does not adequately capture the temporal dynamics of the time series, limiting its predictive capability.

5.1. Dynamic Matrix Factor Model

In the matrix time series setting, Yu et al. (2024) introduced a unified framework that integrates dimensionality reduction with dynamic modeling to enhance both interpretability and predictive performance. Specifically, they proposed the dynamic matrix factor model (DMFM), which consists of two components:

$$\boldsymbol{X}_t = \lambda \boldsymbol{U}_1 \boldsymbol{F}_t \boldsymbol{U}_2^\top + \boldsymbol{E}_t, \qquad \text{(matrix factor model)} \qquad 20.$$

$$\boldsymbol{F}_t = \boldsymbol{A}_1 \boldsymbol{F}_{t-1} \boldsymbol{A}_2^\top + \boldsymbol{\xi}_t, \qquad \text{(matrix autoregressive model)} \qquad 21.$$

where $\boldsymbol{X}_t$ is the observed $d_1 \times d_2$-matrix time series, $\boldsymbol{F}_t$ is the latent factor process, and $\boldsymbol{U}_1 \in \mathbb{R}^{d_1 \times r_1}$ and $\boldsymbol{U}_2 \in \mathbb{R}^{d_2 \times r_2}$ are orthonormal loading matrices. The parameter λ represents the signal strength, while $\boldsymbol{E}_t$ is a white noise process. The latent factor process $\boldsymbol{F}_t$ follows the AR model in Equation 21, where $\boldsymbol{A}_1 \in \mathbb{R}^{r_1 \times r_1}$ and $\boldsymbol{A}_2 \in \mathbb{R}^{r_1 \times r_2}$ are the AR coefficient matrices, and $\boldsymbol{\xi}_t$ is the innovation process. The noise processes $\boldsymbol{E}_t$ and $\boldsymbol{\xi}_t$ are assumed to be uncorrelated at all lags.

The proposed dynamic structure offers two primary advantages: (*a*) It enables efficient factor model prediction through a two-step computationally efficient procedure, and (*b*) the AR formulation provides a transparent and interpretable characterization of the latent factor dynamics. Although the model suffers severe ambiguity, any equivalent estimators produce the same predictions.

Yu et al. (2024) proposed a two-stage estimation procedure, where the first stage estimates the factor model structure in Equation 20 using the iTOPUP or iTIPUP method (Han et al. 2024a), and the second stage estimates the parameters in Equation 21 using the estimated $\boldsymbol{F}_t$ obtained in the first stage. When the SNR in Equation 20 is high, the estimated $\widehat{\boldsymbol{F}}_t$ is relatively accurate and can be treated as observed in the second stage estimation using iterative least squares estimation (LSE). However, when the SNR in Equation 20 is low, $\widehat{\boldsymbol{F}}_t$ may contain relatively large errors. In this case, $\widehat{\boldsymbol{F}}_t$ can be viewed as $\boldsymbol{F}_t$ with an unignorable measurement errors. In this case, Yu et al. (2024) proposed a lagged estimator (L2E) for the AR coefficients $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ in Equation 21, and the factor process $\boldsymbol{F}_t$ is estimated using a Kalman filter within a state-space formulation,

$$\widehat{\boldsymbol{F}}_t = \boldsymbol{F}_t + \boldsymbol{\zeta}_t, \text{ and } \boldsymbol{F}_t = \boldsymbol{A}_1 \boldsymbol{F}_{t-1} \boldsymbol{A}_2^\top + \boldsymbol{\xi}_t.$$

Define the estimation error as $\boldsymbol{\psi} := \left(\mathrm{vec}(\widehat{\boldsymbol{A}}_1 - \boldsymbol{R}_1 \boldsymbol{A}_1 \boldsymbol{R}_1^\top)^\top, \mathrm{vec}(\widehat{\boldsymbol{A}}_2 - \boldsymbol{R}_2 \boldsymbol{A}_2 \boldsymbol{R}_2^\top)^\top\right)^\top$, where $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are appropriate rotation matrices (due to rotational ambiguity of the model). Yu et al.

(2024) show that the asymptotic order of the estimation errors is given by

$$\boldsymbol{\Psi}^{\mathrm{LSE}} = O_{\mathbb{P}}\left(\frac{\sigma\sqrt{d_{\max}}}{\lambda\sqrt{T}} + \frac{\sigma^2}{\lambda^2} + \frac{1}{\sqrt{T}}\right) \quad \text{and} \quad \boldsymbol{\Psi}^{\mathrm{L2E}} = O_{\mathbb{P}}\left(\frac{\sigma\sqrt{d_{\max}}}{\lambda\sqrt{T}} + \frac{\sigma^2}{\lambda^2\sqrt{T}} + \frac{1}{\sqrt{T}}\right),$$

where σ^2 represents the noise level of $\boldsymbol{E}_t$. It follows that the L2E estimator requires a lower SNR for effective performance. When the SNR is sufficiently large, Yu et al. (2024) establish the asymptotic normality of both estimators as $\sqrt{T}\boldsymbol{\Psi}^{\mathrm{LSE}} \to \mathrm{N}(\mathbf{0}, \Xi_1)$ and $\sqrt{T}\boldsymbol{\Psi}^{\mathrm{L2E}} \to \mathrm{N}(\mathbf{0}, \Xi_2)$. Furthermore, when $\mathrm{Cov}(\mathrm{vec}(\boldsymbol{\xi}_t)) = \sigma_\xi^2\mathbf{I}$, it holds that $\Xi_1 \preceq \Xi_2$, implying that $\Xi_2 - \Xi_1$ is positive semidefinite. This result indicates that the LSE estimator outperforms the L2E estimator when the SNR is high.

The choice between the two estimation methods depends on the SNR in Equation 20, which controls the size of measurement errors. The SNR can be assessed via the smallest eigenvalue of the estimated covariance matrix of $\mathrm{vec}(\boldsymbol{\zeta}_t)$, which quantifies the measurement error in $\widetilde{\boldsymbol{F}}_t$. Alternatively, out-of-sample prediction performance provides an empirical measure of estimator accuracy.

Barigozzi & Trapin (2025) proposed a QMLE approach using the expectation–maximization algorithm in conjunction with the Kalman smoother, treating the DMFM as a state-space system. This method jointly estimates the latent factors and model parameters while accommodating serial correlation in the idiosyncratic components.

Yu et al. (2024) propose two prediction approaches based on the SNR in Equation 20. When the SNR is high, the plug-in prediction method directly substitutes the estimated $\boldsymbol{F}_t$ into the MAR model in Equation 21 to produce predictions. In low-SNR scenarios, the Kalman filter–based prediction is employed to account for the increased measurement error.

The model in Equation 21 can be extended to the more general MAR model in Equation 7. Specifically, when r_1 and r_2 in Equation 20 are small, a variety of low-dimensional time series models can be employed, including the VAR model as well as other linear and nonlinear models. The DMFM can be easily extended to tensor time series.

5.2. Two-Way Dynamic Factor Model

Yuan et al. (2023) proposed a two-way dynamic factor model for matrix time series. The model incorporates two sets of additive factors: row factors, $\mathbf{F}_t \in \mathbb{R}^{r_1\times d_2}$, and column factors, $\mathbf{G}_t \in \mathbb{R}^{r_2\times d_1}$, which help to capture both temporal and structural dependencies within the observed matrix time series, $\mathbf{X}_t \in \mathbb{R}^{d_1\times d_2}$.

The model is formulated as follows:

$$\boldsymbol{X}_t = \boldsymbol{F}_t\boldsymbol{L}' + \boldsymbol{\Lambda}\boldsymbol{G}_t^\top + \boldsymbol{E}_t,$$

$$\boldsymbol{F}_t = \boldsymbol{F}_{t-1}\boldsymbol{\Phi}_1 + \cdots + \boldsymbol{F}_{t-p}\boldsymbol{\Phi}_p + \boldsymbol{\varepsilon}_t,$$

$$\boldsymbol{G}_t = \boldsymbol{G}_{t-1}\boldsymbol{\Gamma}_1 + \cdots + \boldsymbol{G}_{t-q}\boldsymbol{\Gamma}_q + \boldsymbol{\eta}_t,$$

where $\boldsymbol{L}$ and $\boldsymbol{\Lambda}$ represent the column and row loading matrices, respectively. The matrices $\boldsymbol{\Phi}_j = \mathrm{diag}(\phi_{j1}, \ldots, \phi_{jr_1})$ for $j = 1, \ldots, p$ and $\boldsymbol{\Gamma}_i = \mathrm{diag}(\gamma_{i1}, \ldots, \gamma_{ir_2})$ for $i = 1, \ldots, q$ are the AR coefficient matrices.

Since $\boldsymbol{\Phi}_j$ and $\boldsymbol{\Gamma}_i$ are diagonal, each time series element in $\boldsymbol{F}_t$ and $\boldsymbol{G}_t$ follows an individual AR model. The terms $\mathbf{E}_t$, $\boldsymbol{\varepsilon}_t$, and $\boldsymbol{\eta}_t$ represent white noise processes that are uncorrelated with each other at all lags. For parameter estimation, Yuan et al. (2023) propose a two-step quasi-likelihood estimation procedure. The convergence rates for the estimated $\boldsymbol{L}$ and $\boldsymbol{G}$ are established, and the estimates for $\boldsymbol{\Phi}$ and $\boldsymbol{\Gamma}$ are shown to be asymptotically normal. The asymptotic variance of the estimates is the same as if $\boldsymbol{F}_t$ and $\boldsymbol{G}_t$ were directly observed.

6. SEGMENTATION APPROACHES

The TenAR models and TFM are complex models with a large number of parameters, aiming to jointly model all cross-serial correlations among individual series. However, these complex models often fail to deliver more accurate predictions than the simpler models that focus on individual series or small subsets (e.g., applying VAR to each fiber separately). One of the main reasons is that while the (weak) cross-serial correlations can, in principle, enhance predictive performance, the large number of parameters in these models leads to substantial estimation errors, which may ultimately compromise prediction accuracy.

Following the time series PCA method of Chang et al. (2018) for vector time series, Han et al. (2024b) proposed a segmentation approach for matrix time series. Their method applies a linear transformation to consolidate weak cross-serial correlations among individual series into a few matrix blocks, ensuring that (*a*) within each block, cross-serial correlations are strong, making joint modeling beneficial, while (*b*) between blocks, cross-serial correlations are negligibly weak, eliminating the need for joint modeling.

More specifically, suppose $\boldsymbol{X}_t \in \mathbb{R}^{p\times q}$ is a matrix time series; the segmentation approach seeks a bilinear transformation so that

$$\boldsymbol{X}_t = \boldsymbol{B}\boldsymbol{U}_t\boldsymbol{A}^\top, \quad \boldsymbol{U}_t = \begin{pmatrix} \boldsymbol{U}_{t,1,1} & \boldsymbol{U}_{t,1,2} & \dots & \boldsymbol{U}_{t,1,n_c} \\ \boldsymbol{U}_{t,2,1} & \boldsymbol{U}_{t,2,2} & \dots & \boldsymbol{U}_{t,2,n_c} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{U}_{t,n_r,1} & \boldsymbol{U}_{t,n_r,2} & \dots & \boldsymbol{U}_{t,n_r,n_c} \end{pmatrix}, \qquad 22.$$

where $\boldsymbol{A} \in \mathbb{R}^{q\times q}$ and $\boldsymbol{B} \in \mathbb{R}^{p\times p}$ are unknown invertible matrices, and $\boldsymbol{U}_t$ is the latent matrix with $\mathrm{Cov}(\mathrm{vec}(\boldsymbol{U}_{t+h,i_1,j_1}), \mathrm{vec}(\boldsymbol{U}_{t,i_2,j_2})) = 0$ for any $(i_1, j_1) \neq (i_2, j_2)$ and any integer h. In other words, $\boldsymbol{X}_t$ is driven by a latent process $\boldsymbol{U}_t$ consisting of contemporaneously and temporally uncorrelated blocks. Once such a structure is identified, forecasting can be done with more parsimonious models for each block separately, often leading to improved prediction accuracy. This is because within-block cross-serial correlations are strong, while the reduced model complexity, due to smaller block sizes, mitigates estimation error. The transformation from $\boldsymbol{U}_t$ to $\boldsymbol{X}_t$ is one-to-one, ensuring that a good forecast performance for $\boldsymbol{U}_t$ directly translates to a good forecast performance for the original series.

Although Equation 22 does not uniquely determine $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{U}_t)$, the maximal segmentation structure (the sizes of each block) is unique up to permutations. Thus, it suffices to identify a single representative of $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{U}_t)$ under a given structure, as all equivalent representations yield the same forecasts for the original series when linear prediction is applied to each $\boldsymbol{U}_t$.

Han et al. (2024b) propose an estimation procedure based on spectral analysis combined with a multiple-testing procedure to infer both the block structure and the transformation. They establish the consistency of the estimated segmentation and transformation matrices under the assumption that the true data-generating process follows Equation 22. Empirically, they demonstrate that even when the true process deviates slightly from this idealized form, their method still leads to improved forecasting performance.

7. SOFTWARE FOR MATRIX AND TENSOR TIME SERIES ANALYSIS

In this section, we review some R packages for tensor time series analysis.

The `tensorTS` package (Li et al. 2024) provides TenAR model estimation with the function `tenAR.est`, with three estimation methods, projection, alternating least squares, and maximum likelihood estimation under a special error covariance structure; reduced-rank MAR model estimation with the function `matAR.RR`; TFM-Tucker estimation with the function `tenFM.est`, allowing TOPUP, TIPUP, iTOPUP and iTIPUP methods; and TFM-Tucker rank implemented via the `tenFM.rank` function, which estimates the factor ranks in the TFM-Tucker. It supports multiple penalty functions and allows both TOPUP and TIPUP approaches.

The `HDTSA` package (Chang et al. 2025b) provides the function `CP_MTS` to estimate a CP decomposition–based factor model for matrix time series, represented as a three-mode tensor, where the first mode corresponds to time. The `method` parameter supports three estimation alternatives: CP.Direct, CP.Refined, and CP.Unified (see Chang et al. 2025a).

The `HDRFA` package (He et al. 2024b) provides a robust estimation for MFM-Tucker models with heavy-tailed idiosyncratic errors by minimizing the Huber loss function (He et al. 2023a). This package implements two algorithms: robust matrix factor analysis, which employs a weighted iterative projection method for parameter estimation, and iterative Huber regression, which is focused on minimizing the element-wise Huber loss. The function `MHFA` implements these two methods. For further details about these approaches and the derivative algorithms implemented in the function described below, readers are referred to He et al. (2023a) and He et al. (2024b). The function `KMHFA` determines the number of factors under either robust matrix factor analysis or iterative Huber regression.

The `RTFA` package provides an alternative method for estimating the TFM-Tucker, as introduced by Barigozzi et al. (2023b). This method employs a one-step projection estimator that minimizes the least squares loss function. Additionally, `RTFA` offers robust estimation capabilities to effectively handle heavy-tailed noise by implementing the methodology proposed by Barigozzi et al. (2023a). This approach builds on the work of Barigozzi et al. (2023b) by utilizing an iterative weighted projection technique based on the Huber loss function. The estimation methods are implemented in the function `TFM_est`, including four methods: initial estimation (without projection), one-step projection estimation, iterative projection estimation, and iterative weighted projection estimation based on the Huber loss function. The function `TFM_FN` determines the number of factors in a $(K + 1)$-mode tensor. Further details on the implemented algorithms are provided by Barigozzi et al. (2023b) and Barigozzi et al. (2023a).

The `TensorPreAve` package (Chen & Lam 2024b) implements the preaveraging technique proposed by Chen (2023). This technique is particularly useful when dealing with serial and cross-correlations in the idiosyncratic components. The main function, `rank_factors_est`, estimates the TFM-Tucker, including identifying the rank of the core tensor and the factor loading matrices unless a specific rank is provided in the argument. The package also offers specialized functions for each step of the estimation process. Users can refer to Chen & Lam (2024b) and Chen (2023) for further details.

The `tensorMiss` package (Cen 2024) implements the method described by Cen & Lam (2025b) for handling missing data in the TFM-Tucker. Its main function is `miss_factor_est`. For more detailed information, please refer to Cen (2024) and Cen & Lam (2025b).

8. SUMMARY AND SOME POTENTIAL FUTURE DEVELOPMENTS

This article reviews recent advancements in modeling and analyzing tensor time series, which have become increasingly prevalent across various applications. As highlighted in the literature, preserving the tensor structure during analysis, rather than vectorizing it, has demonstrated numerous advantages, including improved model parsimony, enhanced interpretability, greater predictive accuracy, more efficient estimation, and reduced computational complexity, among others.

The development of tensor time series analysis remains in its early stages, with significant potential for advancement through developments in tensor data analysis, factor models, high-dimensional statistics, state-space models, Monte Carlo methods, optimization techniques, and beyond. On the modeling front, incorporating trend, seasonality, and ARIMA (autoregressive integrated moving average) structures into tensor time series, as discussed by Tsay (2024), presents a promising direction. Cointegration modeling warrants further exploration, as tensor time series exhibit complex dependencies and unique challenges. Dynamic tensor factor models can be further extended to incorporate additional structures for more effective dimension reduction and improved interpretability. The high-dimensional nature of tensors also calls for the development of specialized nonlinear models and non-Gaussian tensor time series frameworks. Furthermore, more effective tensor time series models are needed to address network and spatiotemporal dependencies. Lastly, developing efficient methods for incorporating covariates into tensor time series models remains an open and important research avenue.

A dedicated resource webpage on tensor time series analysis is available at **https://statistics.rutgers.edu/resources-of-analysis-of-tensor-time-series** and will be regularly updated with the latest research developments. If you have relevant papers (working papers, arXiv publications, or published articles) that you would like to be included, please send them to rongchen@stat.rutgers.edu, in case we overlook them.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

The authors thank the Editorial Committee and an anonymous reviewer for their detailed and constructive comments, which improved the paper significantly. The research is partially supported by National Science Foundation grants DMS-2413858 and DMS-2319260.

LITERATURE CITED

Ahn SC, Horenstein AR. 2013. Eigenvalue ratio test for the number of factors. *Econometrica* 81(3):1203–27

Anandkumar A, Ge R, Janzamin M. 2015. Guaranteed non-orthogonal tensor decomposition via alternating rank-1 updates. Preprint, arxiv:1402.5180 [cs.LG]

Auddy A, Xia D, Yuan M. 2025. Tensors in high-dimensional data analysis: methodological opportunities and theoretical challenges. *Annu. Rev. Stat. Appl.* 12:527–51

Babii A, Ghysels E, Pan J. 2023. Tensor principal component analysis. Preprint, arXiv:2212.12981v2 [econ.EM]

Babii A, Ghysels E, Pan J. 2025. Tensor PCA for factor models. Preprint, arXiv:2212.12981v3 [econ.EM]

Bai J. 2003. Inferential theory for factor models of large dimensions. *Econometrica* 71(1):135–71

Bai J, Ng S. 2002. Determining the number of factors in approximate factor models. *Econometrica* 70(1):191–221

Bai J, Ng S. 2013. Principal components estimation and identification of static factors. *J. Econom.* 176(1):18–29

Bai J, Ng S. 2023. Approximate factor models with weaker loadings. *J. Econom.* 235(2):1893–916

Banin M, Barigozzi M, Trapin L. 2025. Predicting energy demand with tensor factor models. Preprint, arxiv:2502.06213 [stat.AP]

Barigozzi M, Cavaliere G, and GM. 2025a. Factor network autoregressions. *J. Bus. Econ. Stat.* 43(4):1105–18

Barigozzi M, Cho H, Maeng H. 2025b. Tail-robust factor modelling of vector and tensor time series in high dimensions. Preprint, arxiv:2407.09390 [stat.ME]

Barigozzi M, He Y, Li L, Trapani L. 2023a. Robust tensor factor analysis. Preprint, arxiv:2303.18163 [stat.ME]

Barigozzi M, He Y, Li L, Trapani L. 2023b. Statistical inference for large-dimensional tensor factor model by iterative projections. Preprint, arxiv:2206.09800 [stat.ME]

Barigozzi M, Trapin L. 2025. Estimation of large approximate dynamic matrix factor models based on the EM algorithm and Kalman filtering. Preprint, arxiv:2502.04112 [stat.ME]

Barigozzi M, Vecchia DL, Liu H. 2025c. General spatio-temporal factor models for high-dimensional random fields on a lattice. *Ann. Stat.* 53(1):268–94

Basu S, Michailidis G. 2015. Regularized estimation in sparse high-dimensional time series models. *Ann. Stat.* 43(4):1535–67

Bolivar S, Nieto FH, Peña D. 2021. On a new procedure for identifying a dynamic common factor model. *Rev. Colomb. Estadíst.* 44(1):1–21

Bollerslev T. 1987. A conditionally heteroskedastic time series model for speculative prices and rates of return. *Rev. Econ. Stat.* 69(3):542–47

Caro A, Peña D. 2025. Selecting the number of factors in multi-variate time series. *J. Time Ser. Anal.* 46(1):113–36

Carriero A, Clark TE, Marcellino M. 2016. Common drifting volatility in large Bayesian VARs. *J. Bus. Econ. Stat.* 34(3):375–90

Cen Z. 2024. `tensorMiss`: Handle missing tensor data with C++ integration. *R package*, Version 1.1.1

Cen Z, Lam C. 2025a. On testing Kronecker product structure in tensor factor models. Preprint, arxiv:2501.11208 [math.ST]

Cen Z, Lam C. 2025b. Tensor time series imputation through tensor factor modelling. *J. Econom.* 249:105974

Chan JC, Qi Y. 2025. Large Bayesian matrix autoregressions. *J. Econom.* In press

Chang J, Du Y, Huang G, Yao Q. 2025a. Identification and estimation for matrix time series CP-factor models. Preprint, arxiv:2410.05634 [stat.ME]

Chang J, Guo B, Yao Q. 2018. Principal component analysis for second-order stationary vector time series. *Ann. Stat.* 46(5):2094–124

Chang J, He J, Lin C, Yao Q. 2025b. `HDTSA`: high dimensional time series analysis tools. *R package*, Version 1.0.5-1

Chang J, He J, Yang L, Yao Q. 2023. Modelling matrix time series via a tensor CP-decomposition. *J. R. Stat. Soc. Ser. B* 85(1):127–48

Chen B, Chen EY, Bolivar S, Chen R. 2024a. Time-varying matrix factor models. Preprint, arxiv:2404.01546 [stat.ME]

Chen B, Han Y, Yu Q. 2024b. Estimation and inference for CP tensor factor models. Preprint, arxiv:2406.17278 [stat.ME]

Chen EY, Chen R. 2022. Modeling dynamic transport network with matrix factor models: an application to international trade flow. *J. Data Sci.* 21(3):490–507

Chen EY, Fan J. 2021. Statistical inference for high-dimensional matrix-variate factor models. *J. Am. Stat. Assoc.* 118(542):1038–55

Chen EY, Tsay RS, Chen R. 2020. Constrained factor models for high-dimensional matrix-variate time series. *J. Am. Stat. Assoc.* 115(530):775–93

Chen EY, Xia D, Cai C, Fan J. 2024. Semi-parametric tensor factor analysis by iteratively projected singular value decomposition. *J. R. Stat. Soc. Ser. B* 86(3):793–823

Chen H, Raskutti G, Yuan M. 2019. Non-convex projected gradient descent for generalized low-rank tensor regression. *J. Mach. Learn. Res.* 20(1):172–208

Chen R, Giannerini S, Goracci G, Trapani L. 2025. Inference in matrix-valued time series with common stochastic trends and multifactor error structure. Preprint, arxiv:2501.01925 [stat.ME]

Chen R, Xiao H, Yang D. 2021a. Autoregressive models for matrix-valued time series. *J. Econom.* 222(1):539–60

Chen R, Yang D, Zhang CH. 2021b. Factor models for high-dimensional tensor time series. *J. Am. Stat. Assoc.* 117(537):94–116

Chen W. 2023. `TensorPreAve`: rank and factor loadings estimation in time series tensor factor models. *R package*, Version 1.1.0

Chen W, Lam C. 2024a. Factor strength estimation in vector and matrix time series factor models. Preprint, arxiv:2405.07294 [stat.ME]

Chen W, Lam C. 2024b. Rank and factor loadings estimation in time series tensor factor model by pre-averaging. *Ann. Stat.* 52(1):364–91

Cressie N, Johannesson G. 2008. Fixed rank kriging for very large spatial data sets. *J. R. Stat. Soc. Ser. B* 70(1):209–26

Davis RA, Zang P, Zheng T. 2016. Sparse vector autoregressive modeling. *J. Comput. Graph. Stat.* 25(4):1077–96

Engle RF, Kroner KF. 1995. Multivariate simultaneous generalized ARCH. *Econom. Theory* 11(1):122–50

Fan J, Guo J, Zheng S. 2022. Estimating number of factors by adjusted eigenvalues thresholding. *J. Am. Stat. Assoc.* 117(538):852–61

Fan J, Liao Y, Mincheva M. 2013. Large covariance estimation by thresholding principal orthogonal complements. *J. R. Stat. Soc. Ser. B* 75(4):603–80

Forni M, Hallin M, Lippi M, Reichlin L. 2000. The generalized dynamic-factor model: identification and estimation. *Rev. Econ. Stat.* 82(4):540–54

Fosdick BK, Hoff PD. 2014. Separable factor analysis with applications to mortality data. *Ann. Appl. Stat.* 8(1):120–47

Gao Z, Tsay RS. 2023. A two-way transformed factor model for matrix-variate time series. *Econom. Stat.* 27:83–101

Goldenberg A, Zheng AX, Fienberg SE, Airoldi EM. 2010. A survey of statistical network models. *Found. Trends Mach. Learn.* 2(2):129–233

Golub GH, Van Loan CF. 2013. *Matrix Computations*. Johns Hopkins University Press

Han F, Lu H, Liu H. 2015. A direct estimation of high dimensional stationary vector autoregressions. *J. Mach. Learn. Res.* 16(97):3115–50

Han Y, Chen R, Yang D, Zhang CH. 2024a. Tensor factor model estimation by iterative projection. *Ann. Stat.* 52(6):2641–67

Han Y, Chen R, Zhang CH. 2022. Rank determination in tensor factor model. *Electron. J. Stat.* 16(1):1726–803

Han Y, Chen R, Zhang CH, Yao Q. 2024b. Simultaneous decorrelation of matrix time series. *J. Am. Stat. Assoc.* 119(546):957–69

Han Y, Yang D, Zhang CH, Chen R. 2024c. CP factor model for dynamic tensors. *J. R. Stat. Soc. Ser. B* 86(5):1383–413

Hannan EJ. 1970. *Multiple Time Series*. John Wiley and Sons

Hanneke S, Fu W, Xing EP. 2010. Discrete temporal models of social networks. *Electron. J. Stat.* 4:585–605

He Y, Kong X, Trapani L, Yu L. 2024a. Online change-point detection for matrix-valued time series with latent two-way factor structure. *Ann. Stat.* 52(4):1646–70

He Y, Kong X, Yu L, Zhang X, Zhao C. 2023a. Matrix factor analysis: from least squares to iterative projection. *J. Bus. Econ. Stat.* 42(1):322–34

He Y, Kong XB, Liu D, Zhao R. 2023b. Robust statistical inference for large-dimensional matrix-valued time series via iterative Huber regression. Preprint, arxiv:2306.03317 [stat.ME]

He Y, Li L, Liu D, Zhou W. 2024b. `HDRFA`: high-dimensional robust factor analysis. *R package*, Version 0.1.5

Hoff PD. 2011. Separable covariance arrays via the Tucker product, with applications to multivariate relational data. *Bayesian Anal.* 6(2):179–96

Hoff PD. 2015. Multilinear tensor regression for longitudinal relational data. *Ann. Appl. Stat.* 9(3):1169–93

Hsu NJ, Huang HC, Tsay RS. 2021. Matrix autoregressive spatio-temporal models. *J. Comput. Graph. Stat.* 30(4):1143–55

Ji P, Jin J. 2016. Coauthorship and citation networks for statisticians. *Ann. Appl. Stat.* 10(4):1779–812

Kock AB, Callot L. 2015. Oracle inequalities for high dimensional vector autoregressions. *J. Econom.* 186(2):325–44

Kolaczyk ED, Csárdi G. 2014. *Statistical Analysis of Network Data with R*. Springer

Kolda TG, Bader BW. 2009. Tensor decompositions and applications. *SIAM Rev.* 51(3):455–500

Lam C, Yao Q. 2012. Factor modeling for high-dimensional time series: inference for the number of factors. *Ann. Stat.* 40(2):694–726

Lee CE, Zhang X. 2024. Conditional mean dimension reduction for tensor time series. *Comput. Stat. Data Anal.* 199:107998

Li Z, Xiao H. 2021. Multi-linear tensor autoregressive models. Preprint, arxiv:2110.00928 [stat.ME]

Li Z, Xiao H. 2024. Cointegrated matrix autoregression models. Preprint, arxiv:2409.10860 [stat.ME]

Li Z, Yu R, Chen R, Han Y, Xiao H, Yang D. 2024. tensorTS: factor and autoregressive models for tensor time series. *R package*, Version 1.0.2

Lin J, Michailidis G. 2017. Regularized estimation and testing for high-dimensional multi-block vector-autoregressive models. *J. Mach. Learn. Res.* 18(1):4188–236

Linton OB, Tang H. 2022a. Comment on "Factor models for high-dimensional tensor time series" by Rong Chen, Dan Yang, and Cun-Hui Zhang. *J. Am. Stat. Assoc.* 117(537):117

Linton OB, Tang H. 2022b. Estimation of the Kronecker covariance model by quadratic form. *Econom. Theory* 38(5):1014–67

Liu J, Musialski P, Wonka P, Ye J. 2012. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* 35(1):208–20

Liu T, Yuan M, Zhao H. 2022. Characterizing spatiotemporal transcriptome of the human brain via low-rank tensor decomposition. *Stat. Biosci.* 14:485–513

Liu X, Chen EY. 2022. Identification and estimation of threshold matrix-variate factor models. *Scand. J. Stat.* 49(3):1383–417

Liu Y, Shang F, Fan W, Cheng J, Cheng H. 2014. Generalized higher-order orthogonal iteration for tensor decomposition and completion. In *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems*, ed. Z Ghahramani, M Welling, C Cortes, ND Lawrence, KQ Weinberger. MIT Press

Lock EF. 2018. Tensor-on-tensor regression. *J. Comput. Graph. Stat.* 27(3):638–47

Loh PL, Wainwright MJ. 2012. High-dimensional regression with noisy and missing data: provable guarantees with nonconvexity. *Ann. Stat.* 40(3):1637–64

Luo Y, Zhang AR. 2024. Tensor-on-tensor regression: Riemannian optimization, over-parameterization, statistical-computational gap and their interplay. *Ann. Stat.* 52(6):2583–612

Lütkepohl H. 2005. *New Introduction to Multiple Time Series Analysis*. Springer

Melnyk I, Banerjee A. 2016. Estimating structured vector autoregressive model. Preprint, arxiv:1602.06606 [math.ST]

Nicholson WB, Matteson DS, Bien J. 2017. VARX-L: structured regularization for large vector autoregressions with exogenous variables. *Int. J. Forecast.* 3(33):627–51

Onatski A. 2009. Testing hypotheses about the number of factors in large factor models. *Econometrica* 77(5):1447–79

Onatski A. 2010. Determining the number of factors from empirical distribution of eigenvalues. *Rev. Econ. Stat.* 92(4):1004–16

Peña D, Box GE. 1987. Identifying a simplifying structure in time series. *J. Am. Stat. Assoc.* 82(399):836–43

Peña D, Poncela P. 2006. Nonstationary dynamic factor analysis. *J. Stat. Plan. Inference* 136(4):1237–57

Phan TQ, Airoldi EM. 2015. A natural experiment of social network formation and dynamics. *PNAS* 112(21):6595–600

Raskutti G, Yuan M, Chen H. 2019. Convex regularization for high-dimensional multiresponse tensor regression. *Ann. Stat.* 47(3):1554–84

Reinsel GC. 1993. *Elements of Multivariate Time Series Analysis*. Springer

Shumway R, Stoffer D. 2017. *Time Series Analysis and Its Applications: With R Examples*. Springer. 4th ed.

Sidiropoulos ND, De Lathauwer L, Fu X, Huang K, Papalexakis EE, Faloutsos C. 2017. Tensor decomposition for signal processing and machine learning. *IEEE Trans. Signal Proc.* 65(13):3551–82

Snijders TA. 2001. The statistical evaluation of social network dynamics. *Sociol. Methodol.* 31(1):361–95

Stock JH, Watson MW. 2016. Dynamic factor models, factor-augmented vector autoregressions, and structural vector autoregressions in macroeconomics. In *Handbook of Macroeconomics*, Vol. 2, ed. JB Taylor, H Uhlig. Elsevier

Sun WW, Lu J, Liu H, Cheng G. 2017. Provable sparse tensor decomposition. *J. R. Stat. Soc. Ser. B* 79(3):899–916

Tiao GC, Box GEP. 1981. Modeling multiple time series with applications. *J. Am. Stat. Assoc.* 76(376):802–16

Tiao GC, Tsay RS. 1989. Model specification in multivariate time series. *J. R. Stat. Soc. Ser. B* 51(2):157–95

Tsay RS. 2013. *Multivariate Time Series Analysis: With R and Financial Applications*. John Wiley and Sons

Tsay RS. 2024. Matrix-variate time series analysis: a brief review and some new developments. *Int. Stat. Rev.* 92(2):246–62

Velu R, Reinsel GC. 2013. *Multivariate Reduced-Rank Regression: Theory and Applications*. Springer

Wang D, Liu X, Chen R. 2019. Factor models for matrix-valued high-dimensional time series. *J. Econom.* 208(1):231–48

Wang D, Zheng Y, Li G. 2024. High-dimensional low-rank tensor autoregressive time series modeling. *J. Econom.* 238(1):105544

Wang D, Zheng Y, Lian H, Li G. 2020. High-dimensional vector autoregressive time series modeling via tensor decomposition. Preprint, arxiv:1909.06624 [stat.ME]

Wang Y, Zhu Y, Sun Q, Qin L. 2024. Adaptively robust high-dimensional matrix factor analysis under Huber loss function. *J. Stat. Plan. Inference* 231:106137

Wilkinson JH. 1965. *The Algebraic Eigenvalue Problem*. Clarendon Press

Xia D, Yuan M. 2019. On polynomial time methods for exact low-rank tensor completion. *Found. Comput. Math.* 19(6):1265–313

Xia D, Yuan M, Zhang CH. 2021. Statistically optimal and computationally efficient low rank tensor completion from noisy entries. *Ann. Stat.* 49(1):76–99

Xia D, Zhou F. 2019. The sup-norm perturbation of HOSVD and low rank tensor denoising. *J. Mach. Learn. Res.* 20(61):1–42

Xiao H, Han Y, Chen R, Liu C. 2021. *Reduced rank autoregressive models for matrix time series*. Tech. Rep., Rutgers University

Yu C, Li D, Jiang F, Zhu K. 2025a. Matrix GARCH model: inference and application. *J. Am. Stat. Assoc.* 120(551):1747–62

Yu C, Li D, Zhang X, Tong H. 2025b. Two-way matrix autoregressive model with thresholds. Preprint, arxiv:2407.10272 [stat.ME]

Yu L, He Y, Kong X, Zhang X. 2022. Projected estimation for large-dimensional matrix factor models. *J. Econom.* 229(1):201–17

Yu R, Chen R, Xiao H, Han Y. 2024. Dynamic matrix factor models for high dimensional time series. Preprint, arxiv:2407.05624 [stat.ME]

Yuan C, Gao Z, He X, Huang W, Guo J. 2023. Two-way dynamic factor models for high-dimensional matrix-valued time series. *J. R. Stat. Soc. Ser. B* 85(5):1517–37

Yuan M, Zhang CH. 2016. On tensor completion via nuclear norm minimization. *Found. Comput. Math.* 16(4):1031–68

Yuan M, Zhang CH. 2017. Incoherent tensor norms and their applications in higher order tensor completion. *IEEE Trans. Inform. Theory* 63(10):6753–66

Zhang A. 2019. Cross: efficient low-rank tensor completion. *Ann. Stat.* 47(2):936–64

Zhang AR, Luo Y, Raskutti G, Yuan M. 2020. ISLET: fast and optimal low-rank tensor regression via importance sketching. *SIAM J. Math. Data Sci.* 2(2):444–79

Zhang W. 2024. Bayesian dynamic factor models for high-dimensional matrix-valued time series. Preprint, arxiv:2409.08354 [econ.EM]

Zhang X, Li G, Liu CC, Guo J. 2024. Tucker tensor factor models: matricization and mode-wise PCA estimation. Preprint, arxiv:2206.02508 [stat.ME]

Zhao Y, Levina E, Zhu J. 2012. Consistency of community detection in networks under degree-corrected stochastic block models. *Ann. Stat.* 40(4):2266–92

Zhou H, Li L, Zhu H. 2013. Tensor regression with applications in neuroimaging data analysis. *J. Am. Stat. Assoc.* 108(502):540–52