Latent Structure Estimation for High-Dimensional Dependent Data via Eigenanalysis

Sixing Hao

Department of Statistics

The London School of Economics and Political Science

This dissertation is submitted for the degree of

Doctor of Philosophy

April 2025

To my loving family and friends.

Declaration

I certify that the thesis I have presented for examination for the PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work jointly with others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it). I confirm that **Chapters 2** is co-authored with Professor Bo Zhang (from University of Science and Technology of China) and my supervisor Professor Qiwei Yao, and I contributed the following parts of the chapter (Methodology Development, Algorithm, Simulation, Real Data Analysis, R Package Construction and maintenance). A version of this chapter was published on *Statistica Sinica*.

I warrant that this authorisation does not, to the best of my belief, infringe the rights of any third party.

The copyright of this thesis rests with the author. Quotation from it is permitted, provided that full acknowledgment is made. In accordance with the Regulations, I have deposited an electronic copy of it in LSE Theses Online held by the British Library of Political and Economic Science and have granted permission for my thesis to be made available for public reference. Otherwise, this thesis may not be reproduced without my prior written consent.

I declare that my thesis consists of 25325 words.

Sixing Hao

April 2025

Acknowledgements

Completing a PhD thesis is a difficult yet rewarding experience, and it would not have been possible without the help and support of many remarkable individuals. First and foremost, I would like to express my heartfelt gratitude to my supervisor, **Professor Qiwei Yao**, for his exceptional knowledge, insightful guidance, and generous support throughout my PhD journey. His reliable advice and encouragement have been a source of strength, enabling me to navigate the challenges of research with confidence. Being his student has been an honor, and I am grateful for the opportunities to learn from his expertise and mentorship, which have profoundly shaped my growth and development.

I am also thankful to the Department of Statistics at the London School of Economics and Political Science for providing the scholarship and academic resources, enabling me to pursue this research. Further, I wish to extend my gratitude to my colleagues and friends at COL 5.02. They have created a warm and welcoming atmosphere that made this challenging journey not only manageable but also enjoyable. Their support have been invaluable, and I feel privileged to have been part of such a wonderful group.

Finally, my deepest appreciation goes to my family **Jing Liu, Zhengming Hao, and Weiru Hao**, and to my lifetime partner **Fanyi Yang** for their unwavering love and support. Whether listening to my worries, cheering me up during difficult times, or simply being a source of reassurance, their presence has been my anchor throughout the hardest parts of this journey.

I am truly thankful for all the incredible people I have encountered in my life. Each of them has played a part in helping me complete this milestone, and for that, I am profoundly grateful.

Abstract

With advancements in technology, the collection and storage of high-dimensional data have become increasingly common, necessitating tools to analyze such data effectively. Recovering the latent structure has gained popularity as an approach to dimensionality reduction, as the latent processes typically have lower dimensionality, making them easier to analyze and interpret. This thesis explores methods for uncovering the latent structure in high-dimensional data across three domains.

Chapter 2 proposes a novel estimation method for the blind source separation model, as introduced in Bachoc et al. (2020). The new method leverages eigenanalysis of a positive definite matrix constructed from multiple normalized spatial local covariance matrices, enabling the handling of moderately high-dimensional random fields. The consistency of the estimated mixing matrix is established with explicit error rates, even under slowly decaying eigen-gaps.

Chapter 3 examines the factor model framework for time series Lam and Yao (2012a), with a focus on estimating the number of latent factors. Traditional methods struggle with varying factor strengths, limiting their applicability. To address this, a non-parametric hypothesis testing procedure is proposed, capable of identifying the correct number of factors even when factor strengths differ. The proof on significance level of the test is provided, and its effectiveness is demonstrated through comparisons with existing methods on both simulated and real-world datasets.

Chapter 4 addresses the challenge of electricity load forecasting, starting with Generalized Additive Models (GAM) provided by Électricité de France. The residuals from GAM forecasts is analyzed and modeled to uncover its latent structure, which not only simplify

the modeling process but also enhance GAM estimations. Two approaches are explored: latent segmentation using TS-PCA Chang, Guo, and Yao (2018a) and Matrix Time Series Decorrelation Han et al. (2023), and dimensionality reduction with factor models using the procedure developed in Chapter 3. Applied to national and regional electricity load data in France, both methods enhance forecast accuracy, as measured by Root Mean Squared Error (RMSE).

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Background

In the era of rapid technological advancement, data collection and storage have become more convenient and widespread, leading to the prevalence of high-dimensional datasets across various industries. High-dimensional data, defined as having a large number of features compared to the number of observations, has brought new challenges for analysis, including issues of over-parameterization, multicollinearity, and heavy computational burden. Traditional statistical methods, which rely on the assumption of a small number of variables relative to observations, often struggle in tasks such as model fitting and forecasting under these conditions. To address these challenges, many dimension reduction techniques have emerged as powerful tools for finding the lower-dimensional subspace that contains all useful information of the original dataset. Among these, recovering latent structures has gained growing attention as an effective approach, as latent structures typically encapsulate the underlying dynamics of the data in a lower-dimensional representation, enhancing both interpretability and computational efficiency. Latent structure analysis can be viewed as a specific form of finding a low-dimensional subspace, but with a strong focus on interpretability and uncovering hidden relationships.

Latent structures represent the hidden processes or variables that drives the observed data. These structures are analyzed based the assumption that high-dimensional observations are generated by a small number of underlying factors or components. Initially introduced in the context of psychology and social sciences for understanding the potentially existing unobservable traits, latent structure models have since evolved to address challenges in areas such as biology, meteorology, and finance. Their development has been driven by the need to uncover meaningful patterns in noisy or redundant data, providing insights that are otherwise obscured in high-dimensional spaces.

The use of latent structures offers several advantages. By reducing the dimensionality of the observed data, these models alleviate computational constraints and improve the efficiency of further analysis. However, these methods are not without limitations. The assumption of a well-defined latent structure may not hold in all scenarios, particularly in datasets with nonlinearity or highly complex relationships. Furthermore, the estimation of latent components can be sensitive to model specifications and parameter choices, such as factor strength in factor models, or may require prior knowledge of the dataset. For instance, in finance, the widely used three-factor model proposed by Fama and French (1993) assumes prior knowledge of market, size, and value factors as latent variables driving stock returns, which might not generalize to other domains. Balancing the trade-offs between interpretability, model complexity, and computational feasibility remains a critical challenge in the application of latent structure models.

1.2 Structure of Thesis

This thesis is structured as follows: Chapter 2 introduces a method for estimating latent structures in high-dimensional spatial data. Chapter 3 focuses on latent structure estimation in high-dimensional time series, with an emphasis on developing an estimator for the number of latent variables. Chapter 4 demonstrates the application of latent structure analysis to the real-world problem of electricity load forecasting.

1.3 Contribution of Each Chapter

The contribution of each chapter in this thesis can be summarized as follows:

Chapter 2 (Blind Source Separation Over Space) proposed a new estimation method for the blind source separation model of Bachoc et al. (2020). The new estimation is based on an eigenanalysis of a positive definite matrix defined in terms of multiple normalized spatial local covariance matrices, and, therefore, can handle moderately high-dimensional random fields. The consistency of the estimated mixing matrix is established with explicit error rates even when the eigen-gap decays to zero slowly. The proposed method is illustrated via both simulation and a real data example.

Chapter 3 (Permutation Tests for Identifying Number of Factors for High-Dimensional Time Series) presents a non-parametric testing method for determining the number of factors in high-dimensional time series, developed from the factor model proposed by Lam and Yao (2012a). Our introduced estimator utilizes permutation testing to identify the number of factors without relying on assumptions about the underlying factor structure. Unlike traditional ratio-based estimators, our method demonstrates robustness across various factor strength levels and is consistent in its estimation, even when the number of variables p exceeds the number of observations n. Our method is shown to effectively control Type I error and provides reliable estimates in both low and high-dimensional settings. Through theoretical analysis and empirical validation, we demonstrate the advantages of our proposed estimator over existing methods, particularly in scenarios with varying factor strengths. This work contributes to the literature by offering a more flexible, robust, and interpretable approach to factor identification in high-dimensional time series data.

Chapter 4 (Electricity Load Forecasting by Factor Models, TS-PCA and Matrix TS Models) demonstrates the integration of latent structure analysis tools, such as factor models, into a real-world application of electricity load forecasting. By analyzing residual series from Generalized Additive Models (GAM), we show how latent structure methodologies can improve predictive accuracy through dimensionality reduction. Additionally, we propose model-stacking procedures that combine factor models, time-series principal component analysis (TS-PCA), and dynamic Kalman filters, capturing both global latent structures and local temporal dependencies. Applied to national and regional electricity load datasets in France, these methods enhance forecast accuracy, providing a practical framework for utilizing latent structure analysis in high-dimensional data projects.

Chapter 2

Blind Source Separation Over Space

2.1 Introduction

Blind source separation is an effective way to reduce the complexity in modeling p-variant spatial data (Nordhausen et al. (2015), Bachoc et al. (2020)). Spatial data, denoted as $X(s)$, refers to multivariate measurements collected at specific spatial locations $s_1, \ldots, s_n$, where each location s lies in a d-dimensional space. For example, consider soil sampling in a mining field: if we randomly select points on a two-dimensional map and measure the concentrations of 10 elements (such as iron and aluminum) at each point, then $X(s)$ constitutes 10-dimensional spatial data, with the spatial locations $s \in \mathcal{R}^2$.

The approach of blind source separation can be viewed as a version of independent component analysis (Hyvarinen, Karhunen, and Oja (2001)) for multivariate spatial random fields. It is used to learn the underlying independent spatial signals or processes that, when mixed, produce the complex observed data. As spatial data typically exhibit both spatial dependence and inter-variable correlations, recovering these latent components would reduce the complexity and dimensionality of the data, which could enhance computational efficiency in modeling and prediction. Besides, it has the potential of isolating distinct spatial phenomena, which could be interpreted separately. Further, it also helps filtering out the noise or redundant information. In general, blind source separation on spatial data makes it easier to uncover, interpret, and utilize the inherent spatial structures within the data.

Though only the second moment properties are concerned, the challenge is to un-correlate p spatial random fields at the same location as well as across different locations. Note that the standard principal component analysis does not capture spatial correlations, as it only diagonalizes the covariance matrix (at the same location). Nordhausen et al. (2015) introduced a so-called local covariance matrix, defined as

$$M(f) = \frac{1}{n} \sum_{i,j=1}^{n} f(s_i, s_j) X(s_i) X(s_j)^\top,$$

to represent the dependence across different locations. Unlike conventional covariance matrices, which compute covariances using all available observations, a local covariance matrix is constructed by first selecting a subset of observations based on spatial proximity. This approach leverages the natural tendency of spatial data—where nearby measurements are typically more similar—to capture the most relevant covariance structure. By focusing on local neighborhoods, we can reduce computational costs and minimize the impact of noise from distant, less correlated data.

Kernel functions $f(s_i, s_j)$ serve as a mechanism to select observations based on spatial distance among observations. Kernel choice is central in spatial blind source separation because it determines how local covariance matrices are computed, which in turn influences the separation of latent components. In Nordhausen et al. (2015), they employed the ball kernel defined as $f_h(s_i - s_j) = \mathbb{1}(||s_i - s_j|| \leq h)$, which iwhich includes only those observation pairs that are within a specified distance h. This selective filtering includes only pairs of observations within a fixed distance. The only parameter h should be chosen to match the scale at which significant spatial correlation exists. Another common type of kernel is called ring kernel, given by $f(s) = \mathbb{1}(h_1 \leq ||s|| \leq h_2)$, with inner radius h_1 and outer radius h_2. It selectively weights pairs whose separation falls within a specific range, potentially highlighting intermediate-range interactions. Apart from these 2 basic kernels, there are other more complicated kernels, such as the squared exponential kernel, which provides a smooth decay in weight with distance, allowing every pair of points to contribute. The choice and tuning of the kernel function directly affect how local spatial correlations are quantified.

Furthermore, Nordhausen et al. (2015) proposed to estimate the mixing matrix, defined in (2.1) in Section 2.2.1 below, in the blind source separation decomposition based on a generalized eigenanalysis, which can be viewed as an extension of the principal component analysis as it diagonalizes a local covariance matrix in addition to the standard covariance matrix. To overcome the drawback of using the information from only one local covariance matrix, Bachoc et al. (2020) proposed to use multiple local covariance matrices in the estimation (see (2.5) in Section 2.2.2). The method of Bachoc et al. (2020) has a clear advantage in incorporating the spatial dependence information over different ranges. It is in the spirit of JADE (joint approximate diagonalization of eigenmatrices) in non-spatial contexts. See Chapter 11 of Hyvarinen, Karhunen, and Oja (2001) and the references within. Its estimation is based on a nonlinear optimization with p^2 parameters. Hence it is compute-intensive and cannot cope with very large p.

Inspired by Bachoc et al. (2020), we propose a new method also based on multiple (normalized) local covariance matrices for estimating the mixing matrix. Different from Bachoc et al. (2020), the new method is computationally efficient as it boils down to an eigenanalysis of a positive definite matrix which is a matrix function of multiple normalized spatial local covariance matrices. Therefore it can handle the cases with the dimension of random fields in the order of a few thousands on an ordinary personal computer. While the basic idea resembles that of Chang, Guo, and Yao (2018b) which dealt with multiple time series, the spatial random fields concerned are sampled irregularly and non-unilaterally, and the spatial correlations spread in all directions. Furthermore, we incorporate the pre-whitening in our search for the mixing matrix. This implies estimating the covariance matrix of the process, which is assumed to be an identity matrix in Chang, Guo, and Yao (2018b). The normalized spatial local covariance matrix, defined in (2.10) below, is a modified version of the spatial local covariance matrix in Nordhausen et al. (2015), and is introduced to facilitate the effect of the pre-whitening. All these entail completely different theoretical exploration; leading to the asymptotic results under the similar setting of Bachoc et al. (2020) but allowing the dimension of the random field to diverge together with the number of the observed locations, which is assumed to be fixed in Bachoc et al. (2020).

The efficiency gain in computing of the proposed method is due to adding together the information from different normalized local covariance matrices. However, rather than adding covariance matrices directly such as in TDSEP (Ziehe and Müller (1998)), each term in the sum of (2.9) in Section 2.2.3 below is the product of a normalized local covariance matrix and its transpose, which, therefore, is non-negative definite matrix. This avoids the possible cancellation of the information from different normalized local covariance matrices. Note covariance matrices are not non-negative definite, and adding them together directly may leads to volatile performance due to information cancellation; see Table 1 of Ziehe and Müller (1998). Although the sample fourth moments occur in (2.9) in order to avoid the information cancellation, our goal is decorrelaton across space via diagonalizing multiple normalized local covariance matrices. Indeed the way to use the fourth moments and the purpose of using them are radically different from those of FOBI (forth-order blind identification) algorithms. See Chapter 11 of Hyvarinen, Karhunen, and Oja (2001) and the references within.

Another new contribution of this chapter concerns the eigen-gap in the eigenanalysis for estimating the mixing matrix. In order to identify a consistent estimator for the mixing matrix, the standard condition is to assume that the minimum pairwise absolute difference among the eigenvalues remains positive. See Assumptions 8 and 9 of Bachoc et al. (2020). The similar conditions have been imposed in the literature in order to identify factor loading spaces in factor models in Lam and Yao (2012b). However this condition is invalid under the setting concerned in this chapter when the dimension of random field p diverges to infinity, as the maximum order of the eigen-gap is p^{-1}. We show that the identification of the mixing matrix is still possible when $p \to \infty$ at the rate $p = o(n^{1/3})$. See Theorem 2.3.2 and Remark 2 in Section 2.3.

The rest of the chapter is organized as follows. We present the spatial blind source separation model and the new estimation method in Section 2.2. The asymptotic properties are developed in Section 2.3. Numerical illustration with both simulated data and a real data set is presented in Section 2.4. All the technical proofs are given in the Appendix A.1-A.2.

The R-package `BSSoverSpace`, available in the CRAN project, implements the methods proposed in this chapter.

2.2 Setting and Methodology

2.2.1 Model

We adopt the spatial blind source separation model of Bachoc et al. (2020). More precisely, let $X(s) = \{X_1(s), \cdots, X_p(s)\}^\top$ be a p-variate random field defined on $s \in \mathscr{S} \subset \mathcal{R}^d$, and $X(s)$ admits the representation

$$X(s) = \Omega Z(s) \equiv \Omega\{Z_1(s), \cdots, Z_p(s)\}^\top, \tag{2.1}$$

where $Z_1(s), \cdots, Z_p(s)$ are p independent latent random fields, and Ω is a $p \times p$ invertible constant matrix and is called the mixing matrix. Furthermore, Bachoc et al. (2020) assumes that for any $s, u \in \mathscr{S}$,

$$EZ(s) = \mu_0, \quad \text{Var}\{Z(s)\} = I_p, \quad \text{Cov}\{Z(s), Z(u)\} = H(s-u), \tag{2.2}$$

where μ_0 is an unknown constant vector, I_p denotes the $p \times p$ identity matrix, $H(\cdot)$ is a $p \times p$ diagonal matrix

$$H(s-u) = \text{diag}\{K_1(s-u), \cdots, K_p(s-u)\},$$

i.e. $\text{Cov}\{Z_i(s), Z_j(u)\} = K_i(s-u)$ if $i = j$, and 0 otherwise. Let $\mu = \Omega\mu_0$. Under (2.1) and (2.2), $X(\cdot)$ is a weakly stationary process as

$$EX(s) = \mu, \quad \text{Var}\{X(s)\} = \Omega\Omega^\top, \quad \text{Cov}\{X(s), X(u)\} = \Omega H(s-u)\Omega^\top. \tag{2.3}$$

2.2.2 The Existing Methods

Let $X(s_1), \cdots, X(s_n)$ be available observations. Put

$$\tilde{X}(s_i) = X(s_i) - \frac{1}{n}\sum_{j=1}^{n} X(s_j), \quad \tilde{Z}(s_i) = Z(s_i) - \frac{1}{n}\sum_{j=1}^{n} Z(s_j), \quad i = 1, \cdots, n.$$

Then the spatial local covariance matrix of Nordhausen et al. (2015) is defined as

$$\tilde{M}(f) = \frac{1}{n}\sum_{i,j=1}^{n} f(s_i - s_j)\tilde{X}(s_i)\tilde{X}(s_j)^\top, \tag{2.4}$$

where $f(\cdot)$ is a kernel function such as the ring kernel $f(s) = \mathbb{1}(h_1 \leq \|s\| \leq h_2)$ for some constants $0 \leq h_1 < h_2 < \infty$, and $\mathbb{1}(\cdot)$ denotes the indicator function. To recover the mixing matrix Ω, Bachoc et al. (2020) proposed to estimate the unmixing matrix (i.e. the inverse of the mixing matrix) $\Gamma = \Omega^{-1} \equiv (\gamma_1, \cdots, \gamma_p)^\top$ by

$$\hat{\Gamma} \in \arg \max_{\Gamma \tilde{M}(f_0)\Gamma^\top = I_p} \sum_{i=1}^{k}\sum_{j=1}^{p}\{\gamma_j^\top \tilde{M}(f_i)\gamma_j\}^2, \tag{2.5}$$

where $f_0(s) = I(s = 0)$, and $f_1, \cdots, f_k$ are appropriately specified kernels. This is a nonlinear optimization problems with p^2 variables, which Bachoc et al. (2020) adopted the algorithm of Clarkson (1988) to solve. When $k = 1$, the objective function contains only one kernel function. Then the above optimization can be solved based on a generalized eigenanalysis; see Nordhausen et al. (2015) and Bachoc et al. (2020), though the estimation based on a single kernel requires the prior knowledge on which kernel to use for a given problem.

2.2.3 The New Method

We now propose a new method to estimate the mixing matrix using multiple kernels but based on a single eigenanalysis. To this end, we define, for any given k kernel function $f_1(\cdot), \cdots, f_k(\cdot)$,

$$N = E\Big[\frac{1}{k}\sum_{h=1}^{k}\{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}(s_i)\tilde{Z}(s_j)^\top\}\{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}(s_i)\tilde{Z}(s_j)^\top\}^\top\Big], \tag{2.6}$$

$$\begin{aligned} W = E\Big[&\frac{1}{k}\sum_{h=1}^{k}\{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\Sigma^{-1/2}\tilde{X}(s_i)\tilde{X}(s_j)^\top\}\Sigma^{-1} \\ &\times\{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{X}(s_i)\tilde{X}(s_j)^\top\Sigma^{-1/2}\}^\top\Big], \end{aligned}$$

where $\Sigma = \text{Var}\{X(s)\} = \Omega\Omega^\top$. Then N and W are $p \times p$ non-negative definite matrices. Furthermore, N is a diagonal matrix, as its (i, j)-th element, for $i \neq j$, is

$$\frac{1}{n^2k}\sum_{h=1}^{k}\sum_{\ell=1}^{p}\sum_{i_1,i_2,j_1,j_2=1}^{n} f_h(s_{i_1}-s_{j_1})f_h(s_{i_2}-s_{j_2})E\{\tilde{Z}_i(s_{i_1})\tilde{Z}_\ell(s_{j_1})\tilde{Z}_j(s_{i_2})\tilde{Z}_\ell(s_{j_2})\} = 0,$$

which is guaranteed by the fact that the components of $Z(\cdot)$ are the p independent random fields. Since Ω is a $p \times p$ full rank matrix, we can rewrite $\Omega = V_\Omega \Lambda_\Omega U_\Omega$, where V_Ω and U_Ω are two $p \times p$ orthogonal matrices, and Λ_Ω is a diagonal matrix. Then $\Sigma^{-1/2} = V_\Omega \Lambda_\Omega^{-1} V_\Omega^\top$. Combining this and (2.1), we have

$$W = V_\Omega U_\Omega N U_\Omega^\top V_\Omega^\top, \tag{2.7}$$

i.e. the columns of $U_W \equiv V_\Omega U_\Omega$ are the p orthonormal eigenvectors of matrix W with the diagonal elements of N as the corresponding eigenvalues. As $\Sigma^{1/2}U_W = V_\Omega \Lambda_\Omega V_\Omega^\top V_\Omega U_\Omega = \Omega$, this paves the way to identifying mixing matrix Ω. We summarize the finding in the proposition below.

Proposition 2.2.1. *Under the condition (2.2), the mixing matrix Ω defined in (2.1) is of the form $\Sigma^{1/2}U_W$, where the columns of U_W are the p orthonormal eigenvectors of matrix W. Moreover, those p eigenvectors are identifiable, upto the sign changes, if the p diagonal elements of N are distinct from each other.*

Note that the sign changes of any columns of U_W will not change the independence of the components of $Z(\cdot)$ in (2.1), as $Z(s) = U_W^\top \Sigma^{-1/2} X(s)$. By Proposition 2.2.1, we define an estimator for the mixing matrix as

$$\hat{\Omega} = \hat{\Sigma}^{1/2}\hat{U}_W, \tag{2.8}$$

where $\hat{\Sigma} = n^{-1}\sum_{1\leq j\leq n}\tilde{X}(s_j)\tilde{X}(s_j)^\top$, and the columns of $\hat{U}_W$ are the p orthonormal eigenvectors of matrix

$$\hat{W} = \frac{1}{k}\sum_{h=1}^{k}\hat{M}(f_h)\hat{M}(f_h)^\top. \tag{2.9}$$

In the above expression, $\hat{M}(f_h)$ is a normalized local covariance matrix defined as

$$\hat{M}(f) = \frac{1}{n} \sum_{i,j=1}^{n} f(s_i - s_j)\hat{\Sigma}^{-1/2}\tilde{X}(s_i)\tilde{X}(s_j)^\top \hat{\Sigma}^{-1/2}. \tag{2.10}$$

This estimation procedure is implemented in Algorithm 1 below. In comparison to the local covariance matrix (2.4), we replace $X(\cdot)$ by its standardized version $\hat{\Sigma}^{-1/2}\tilde{X}(\cdot)$. This effectively pre-whitens the data in our search for the mixing matrix.

```
Algorithm 1: Eigenanalysis approach for BSS over space
  Input: $X(s_1), \cdots, X(s_n)$ and $f_1(\cdot), \cdots, f_k(\cdot)$.
  (i) Compute $\tilde{X}(s_i) = X(s_i) - \frac{1}{n}\sum_{j=1}^{n} X(s_j)$ and $\hat{\Sigma} = n^{-1}\sum_{1\le j\le n}\tilde{X}(s_j)\tilde{X}(s_j)^\top$.
  (ii) Compute $\hat{W}$ in (2.9).
  (iii) Compute eigenvalues $\hat{\Lambda}_W$ and eigenvectors $\hat{U}_W$ of matrix $\hat{W}$.
  (iv) Compute $\hat{\Omega}^{-1} = \hat{U}_W^\top \hat{\Sigma}^{-1/2}$.
  Output: $\hat{Z}(s_i) = \hat{\Omega}^{-1} X(s_i)$, $i = 1, \cdots, n$.
```

Remark 1. The proposed new method makes use of the normalized 4th moments of the observations while the methods of Bachoc et al. (2020) and Nordhausen et al. (2015) only depend on the 2nd moments. However the 4th moments occur only in the matrix products $\hat{M}(f_h)\hat{M}(f_h)^\top$ in defining $\hat{W}$ in (2.9), and each of those products is a non-negative definite matrix. We add together those non-negative definite matrices, instead of $\hat{M}(f_h)$ (as suggested in Ziehe and Müller (1998)), to avoid the information cancellation from different $\hat{M}(f_h)$. See also Chang, Guo, and Yao (2018b). Note that both our way of using the fourth moments and our purpose of using them are radically different from those of FOBI (Hyvarinen, Karhunen, and Oja (2001, Chapter 11)).

For example, W in (2.6) is a $p \times p$ matrix with the (l,m)-th element

$$E\Big[\frac{1}{n^2 k}\sum_{h=1}^{k}\sum_{v=1}^{p}\sum_{i,j,c,d=1}^{n} f_h(s_i - s_j) f_h(s_c - s_d)\tilde{Z}_l(s_i)\tilde{Z}_m(s_c)\tilde{Z}_v(s_j)\tilde{Z}_v(s_d)\Big],$$

while a FOBI algorithm would use instead a $p^2 \times p^2$ quadricovariance matrix with the elements being the fourth order cumulants (Ferréol, Albera, and Chevalier (2005)). Our goal is to avoid information cancellation while diagonalizing different local covariance matrices. FOBI is to diagonalize a quadricovariance matrix.

2.3 Asymptotic Properties

We consider the asymptotic behavior of the estimator $\widehat{\Omega}$ when $n \to \infty$ and p either remaining fixed or $p = o(n)$. Since $\widehat{\Omega}^{-1}X(s) = \widehat{\Omega}^{-1}\Omega Z(s)$, we will focus on $\widehat{\Gamma}_\Omega = \widehat{\Omega}^{-1}\Omega$. We introduce some regularity assumptions first.

Assumption 2.3.1. *In model (2.1), $Z_1(\cdot), \cdots, Z_p(\cdot)$ are p independent and strictly stationary random fields on R^d, and assumption (2.2) holds. Furthermore, $Z(\cdot)$ is sub-Gaussian in the sense that there exists a constant $C_0 > 0$ independent of p for which*

$$\sup_{\beta \geq 1, 1 \leq i \leq p} \beta^{-1/2}\{E|Z_i(s)|^\beta\}^{1/\beta} \leq C_0. \tag{2.11}$$

Moreover, for any unit vector $(a_1, \cdots, a_n)^\top \in R^n$ and $1 \leq \ell \leq p$, $\sum_{i=1}^n a_i Z_\ell(s_i)$ is sub-Gaussian.

Assumption 2.3.2. *There exist positive constants Δ, α and A (independent of n and p) such that for any $1 \leq i \neq j \leq n$ and $n \geq 2$, $\|s_i - s_j\| \geq \Delta$, and for $s, u \in R^d$, $1 \leq \ell \leq p$ and $1 \leq h \leq k$ (k is fixed),*

$$|\mathrm{Cov}\{Z_\ell(s+u), Z_\ell(s)\}| \leq A/(1 + \|u\|^{d+\alpha}), \tag{2.12}$$

$$|f_h(s)| \leq A/(1 + \|s\|^{d+\alpha}). \tag{2.13}$$

Assumption 2.3.3. *Let $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$ be the diagonal elements of matrix N defined in (2.6), arranged in the descending order. There exist integers $0 = p_0 < p_1 < \cdots < p_m = p$ for which*

$$\limsup_{n\to\infty} \max_{1 \leq i \leq m} |\lambda_{p_{i-1}+1} - \lambda_{p_i}| = 0, \qquad \text{and} \tag{2.14}$$

$$\liminf_{n\to\infty} \min_{1 \leq i < m} |\lambda_{p_i} - \lambda_{p_i+1}| = C_1 > 0, \tag{2.15}$$

where $m \geq 2$ is a fixed integer, and C_1 is a constant independent of p.

Assumptions 2.3.1 and 2.3.2 are essentially the same as Assumptions 1-7 of Bachoc et al. (2020), though we impose only the sub-Gaussianality instead of requiring $Z(\cdot)$ to be normally distributed. In addition, our setting allows p to diverge together with n. Assumption 2.3.3 is required for distinguishing the columns of the mixing matrix Ω from each other. Those p columns are completely identifiable when p is fixed and $m = p$. Then condition (2.14) vanishes, and (2.15) ensures that the p diagonal elements of matrix N are distinct from each other (see Proposition 2.2.1). The similar conditions (i.e. with p fixed) were imposed in Bachoc et al. (2020): see Assumptions 8 and 9 therein. Note that condition (2.15) cannot hold when $m = p \to \infty$. When $p \to \infty$ together with n, (2.14) and (2.15) ensure that the estimated mixing matrix $\widehat{\Omega}$ transforms $X(\cdot)$ into m independent subvectors; see Theorem 2.3.1 below. Recalling the definition of N in (2.6), we can see that the choice of kernels should satisfy Assumption 2.3.3. This is the same for Bachoc et al. (2020).

Without the loss of generality, we assume that the p components of $Z(\cdot)$ are arranged in the order such that the diagonal elements of matrix N in (2.6) are in the descending order. This simplifies the presentation of Theorem 2.3.1 substantially.

Write $\widehat{W} = \widehat{U_W}\widehat{\Lambda}_W\widehat{U}_W^\top$ as its spectral decomposition, i.e.

$$\widehat{\Lambda}_W = \operatorname{diag}(\widehat{\lambda}_{W,1}, \cdots, \widehat{\lambda}_{W,p}),$$

where $\widehat{\lambda}_{W,1} \geq \cdots \geq \widehat{\lambda}_{W,p} \geq 0$ are the eigenvalues of $\widehat{W}$, and the columns of the orthogonal matrix $\widehat{U}_W$ are the corresponding eigenvectors. Consequently,

$$\widehat{\Gamma}_\Omega = \widehat{\Omega}^{-1}\Omega = \widehat{U}_W^\top\widehat{\Sigma}^{-1/2}\Omega. \tag{2.16}$$

Corollary 2.3.1 below shows that $\widehat{\Omega}^{-1}\Omega = \widehat{\Gamma}_\Omega \xrightarrow{P} I_p$ when p is finite and $m = p$ in Assumption 2.3.3. To state a more general result first, put $q_i = p_i - p_{i-1}$ for $i = 1, \cdots, m$ (see

Assumption 2.3.1), and

$$\widehat{\Omega}^{-1}\Omega = \widehat{\Gamma}_{\Omega} = \begin{pmatrix} \widehat{\Gamma}_{\Omega,11} & \cdots & \widehat{\Gamma}_{\Omega,1m} \\ \cdots & \cdots & \cdots \\ \widehat{\Gamma}_{\Omega,m1} & \cdots & \widehat{\Gamma}_{\Omega,mm} \end{pmatrix}, \tag{2.17}$$

where submatrix $\widehat{\Gamma}_{\Omega,ij}$ is of the size $q_i \times q_j$.

Theorem 2.3.1. *Let Assumptions 2.3.1-2.3.3 hold. As $n \to \infty$ and $p = o(n)$, it holds that*

$$||\hat{\Gamma}_{\Omega,ii}|| = 1 + O_p\{n^{-1/2}p^{1/2}\}, \ ||\hat{\Gamma}_{\Omega,ii}||_{min} = 1 + O_p\{n^{-1/2}p^{1/2}\} \quad 1 \le i \le m, \tag{2.18}$$

$$||\hat{\Gamma}_{\Omega,ij}|| = O_p\{n^{-1/2}p^{1/2}\}, \quad 1 \le i \ne j \le m, \quad \text{and} \tag{2.19}$$

$$||\hat{\Lambda}_W - \Lambda|| = O_p(n^{-1/2}p^{1/2}), \tag{2.20}$$

where $\Lambda = \operatorname{diag}(\lambda_1, \cdots, \lambda_p)$, and λ_i are specified in Assumption 2.3.3.

Theorem 2.3.1 implies that $\hat{\Gamma}_{\Omega,ij} \xrightarrow{P} 0$ for any $i \ne j$. Hence the transformed process $\widehat{\Omega}^{-1}X(\cdot) = \widehat{\Gamma}_{\Omega}Z(\cdot)$ can only be divided into the m asymptotically independent random fields of dimensions $q_1, \cdots, q_m$ respectively. This is due to the lack of separation of the corresponding eigenvalues within each of those m groups; see (2.14). On the other hand, Theorem 2.3.1 still holds, under some additional conditions, if the components of $Z(\cdot)$ within each of those m groups are not independent with each other. Then this is in the spirit of the so-called multidimensional independent component analysis of Cardoso (1998). In practice, one needs to identify the m latent groups among the p components of $\widehat{\Omega}^{-1}X(\cdot)$, which can be carried out by adapting the procedures in Section 2.2 of Chang, Guo, and Yao (2018b). By (2.20), $\hat{\Lambda}_W$ will indicate how those eigenvalues are different from each other; see Assumption 2.3.3.

Note that Theorem 2.3.1 holds when either p is fixed and finite, or $p/n \to 0$ as $n \to \infty$. When p is fixed and $m = p$ in Assumption 2.3.3, all $\hat{\Gamma}_{\Omega,ij}$ reduces to a scale and $q_i = 1$. Then Corollary 2.3.1 below follows from Theorem 2.3.1 immediately.

Corollary 2.3.1. *Let Assumptions 2.3.1-2.3.3 hold with $m = p$, and p be a fixed integer. Then as $n \to \infty$, $\|I_p - \widehat{\Omega}^{-1}\Omega\| = O_p(n^{-1/2})$.*

A key condition in Corollary 2.3.1 for identifying all the columns of the mixing matrix is that the eigengap defined as

$$v_{\text{gap}} = \min_{1 \le i \ne j \le p} |\lambda_i - \lambda_j| \tag{2.21}$$

remains bounded away from 0, which is implied by (2.15) when $p = m$ is fixed. This condition cannot be fulfilled when p diverges (together with n). To appreciate the performance of the proposed procedure when p is large in relation to n, we present Theorem 2.3.2 below which indicates that the mixing matrix can still be estimated consistently but at much slower rates when the eigengap v_{gap} decays to 0 provided p diverges to ∞ not too fast; see Remark 2 below.

Assumption 2.3.4. $\limsup_{n\to\infty} v_{\text{gap}}^{-1} n^{-1/2} p^{1/2} = 0$.

Theorem 2.3.2. *Let Assumptions 2.3.1, 2.3.2 and 2.3.4 hold. Denote by $\hat{\gamma}_{\Omega,ij}$ the (i,j)-th entry of matrix $\widehat{\Gamma}_\Omega$. Then as $n, p \to \infty$, it holds that*

$$\widehat{\gamma}_{\Omega,ij} = O_p(n^{-1/2} p^{1/2} v_{\text{gap}}^{-1} |j-i|^{-1}) \quad \text{for } 1 \le i \ne j \le p, \quad \text{and} \tag{2.22}$$

$$\widehat{\gamma}_{\Omega,ii} = 1 + O_p(n^{-1} p v_{\text{gap}}^{-2} + n^{-1/2} p^{1/2}) \quad \text{for } i = 1, \cdots, p. \tag{2.23}$$

Moreover, (2.20) still holds.

Remark 2. Note that $\lambda_1 - \lambda_p \ge (p-1)v_{\text{gap}}$, and, therefore, $v_{\text{gap}} = O(p^{-1})$. Thus it follows from Assumption 2.3.4 that $p = o(n^{1/3})$, i.e. in order to fully identify the mixing matrix, p cannot be too large in the sense that $p/n^{1/3} \to 0$.

2.4 Numerical Illustration

2.4.1 Simulation

We illustrate the finite sample properties of the proposed method by simulation. We set the dimension of random fields at p =3 and 50, and the sample size n (i.e. the number of locations) between 100 to 2000. The coordinates of those n locations are drawn independently from $U(0,50)^2$. Both Gaussian and non-Gaussian random fields are used. Also included in the simulation is the method of Bachoc et al. (2020). For each setting, we replicate the simulation 1000 times.

The p-variate random fields $X(\cdot)$ are generated according to (2.1) in which $Z_1(\cdot),\cdots,Z_p(\cdot)$ are p independent random fields with either $N(0,1)$ or t_5 marginal distributions, and the Matern correlation function

$$\rho(s) = 2^{1-\kappa}\Gamma(\kappa)^{-1}(s/\phi)^{\kappa}B_{\kappa}(s/\phi),$$

where $\kappa > 0$ is the shape parameter, $\phi > 0$ is the range parameter, $\Gamma(\cdot)$ is the Gamma function, and B_κ is the modified Bessel function of the second kind of order κ. We set different values of (κ,ϕ) for different Z_j. More precisely κ's are drawn independently from $U(0,6)$, and ϕ's are drawn independently from $U(0,2)$. The mixing matrix Ω in (2.1) is set to be the $p \times p$ identity matrix.

To measure the accuracy of the estimation for Ω, we define

$$D(\Omega,\hat{\Omega}) = \frac{1}{2p(\sqrt{p}-1)}\sum_{j=1}^{p}\Big\{\frac{(\sum_{1\le i\le p} d_{ij}^2)^{1/2}}{\max_{1\le i\le p}|d_{ij}|} + \frac{(\sum_{1\le i\le p} d_{ji}^2)^{1/2}}{\max_{1\le i\le p}|d_{ji}|} - 2\Big\},$$

where d_{ij} is the (i,j)-th element of matrix $\Omega^{-1}\hat{\Omega}$. As

$$p^{-1/2} \le \max_{1\le i\le p}|d_{ij}| \Big/ \Big(\sum_{1\le i\le p} d_{ij}^2\Big)^{1/2} \le 1.$$

it holds that $D(\Omega,\hat{\Omega}) \in [0,1]$, and $D(\Omega,\hat{\Omega}) = 0$ if $\hat{\Omega}$ is a column permutation and/or column sign changes of Ω.

We set $k = 10$ in (2.9), and

$$f_h(s) = \mathbb{1}(c_{h-1} < \|s\| \leq c_h), \qquad h = 1,\cdots,10, \tag{2.24}$$

where $0 = c_0 < c_1 < \cdots < c_{10} = \infty$ are specified such that for each $h = 1,\cdots,10$, $\{(s_i,s_j) : 1 \leq i < j \leq n,\ c_{h-1} < \|s_i - s_j\| \leq c_h\}$ contains the 10% of the total pairs (s_i,s_j), $1 \leq i < j \leq n$.

The boxplots of $D(\Omega,\hat{\Omega})$ obtained in the 1000 replications are presented in Figures 2.1–2.4. Estimations by the method of Bachoc et al. (2020) are computed using the R-function `sbss`, provided in R-package `SpatialBSS`. In addition to the multiple kernel estimation, we also compute the estimates with a single kernel, using each of the 10 kernels in (2.24),

For computing the multiple kernel method of Bachoc et al. (2020), we set the maximum number of iterations at 2000. By using a single kernel, the method of Bachoc et al. (2020) leads to almost identical estimates as those obtained by the proposed method (with the same single kernel). Therefore we omit the detailed results.

Figures 2.1 – 2.4 and Tables 2.1 – 2.4 indicate clearly that both the methods with multiple kernels outperform most of those with a single kernel, and the proposed method outperforms the multiple kernel method of Bachoc et al. (2020) especially when p is large (i.e. $p = 50$).

The proposed method with multiple kernels performs about the same as that with the best single kernel (i.e. Kernel 1 $f_1(\cdot)$). The accuracy of estimation improves with the increase in the number of observations n, which can be seen as a decrease in $D(\Omega,\hat{\Omega})$ in Figures 2.1–2.4. Among all single kernel methods, those using kernel f_1 perform the best, as those estimations include the 10% nearest locations. Indeed the Matern correlation is the strongest at the smallest distance. On the other hand, the performances for the Gaussian and the non-Gaussian random fields are about the same. See Figures 2.1 & 2.2, and Figures 2.3 & 2.4.

The iterative algorithm for implementing the multiple kernel method of Bachoc et al. (2020) is to solve a nonlinear optimization problem with p^2 parameters. When $p = 50$, it failed to converge within the 2000 iterations in some of the 1000 simulation replications. The numbers of failures with n =100, 500, 1000 and 2000 are, respectively, $3,1,2$ and 1 for the Gaussian random fields, and $6,3,3$ and 1 for the non-Gaussian random fields. We only include the results from the converged replications in the figures.

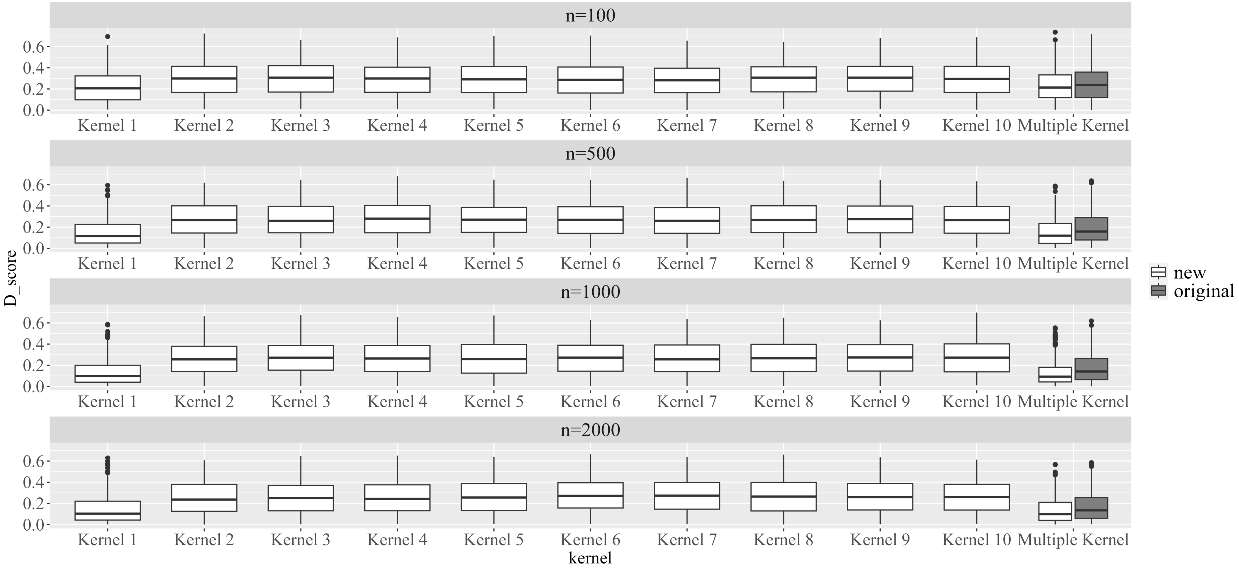

Fig. 2.1 Boxplots of $D(\Omega,\hat{\Omega})$ for the proposed method using the 10 kernels (new) in (2.24), or each of those 10 kernels (Kernel 1, $\cdots$, Kernel 10), and the method of Bachoc et al. (2020) using the 10 kernels (original) in a simulation with 1000 replications for the Gaussian random fields. The number of observations n is 100, 500, 1000 or 2000 (from top to bottom), and the dimension of random fields is $p=3$.

Kernel	1	2	3	4	5	6	7	8	9	10	Multiple(new)	Multiple(original)
n=100	0.0814	0.2284	0.2707	0.2584	0.2594	0.2617	0.2542	0.2688	0.2619	0.2517	0.0933	0.1298
n=500	0.0248	0.1437	0.2019	0.2051	0.2042	0.1873	0.1830	0.1926	0.2076	0.2071	0.0327	0.0444
n=1000	0.0189	0.1124	0.1992	0.1782	0.1800	0.1746	0.1862	0.1803	0.1823	0.1887	0.0233	0.0324
n=2000	0.0164	0.1194	0.1870	0.1631	0.1686	0.1746	0.1533	0.1761	0.1701	0.1845	0.0204	0.0260

Table 2.1 Median of $D(\Omega,\hat{\Omega})$ from the proposed method using the 10 single kernels, or multiple kernel(including all 10 ring kernels), and the method of Bachoc et al. (2020). using the multiple kernel (multiple original) in a simulation with 1000 replications for the Gaussian random fields. The number of observations n is 100, 500, 1000 or 2000 , and the dimension of random fields is $p=3$.

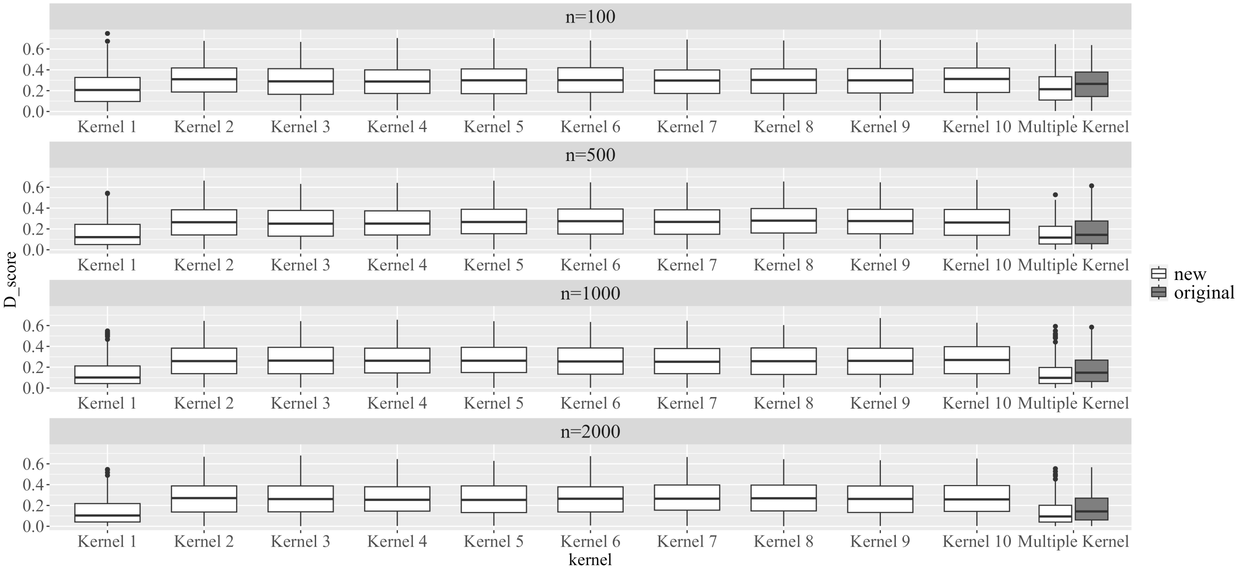

Fig. 2.2 Boxplots of $D(\Omega, \hat{\Omega})$ for the proposed method using the 10 kernels (new) in (2.24), or each of those 10 kernels (Kernel 1, $\cdots$, Kernel 10), and the method of Bachoc et al. (2020) using the 10 kernels (original) in a simulation with 1000 replications for the non-Gaussian random fields. The number of observations n is 100, 500, 1000 or 2000 (from top to bottom), and the dimension of random fields is $p = \ 3$.

Kernel	1	2	3	4	5	6	7	8	9	10	Multiple(new)	Multiple(original)
n=100	0.0837	0.2377	0.2656	0.2579	0.2676	0.2664	0.2478	0.2503	0.2440	0.2531	0.0915	0.1194
n=500	0.0244	0.1526	0.2028	0.2001	0.2052	0.1998	0.1923	0.1964	0.2028	0.2107	0.0284	0.0424
n=1000	0.0178	0.1096	0.1767	0.1943	0.1868	0.1812	0.1741	0.1627	0.1885	0.1911	0.0215	0.0326
n=2000	0.0165	0.1230	0.1907	0.1765	0.1676	0.1663	0.1652	0.1625	0.1742	0.1818	0.0194	0.0293

Table 2.2 Median of $D(\Omega, \hat{\Omega})$ from the proposed method using the 10 single kernels, or multiple kernel(including all 10 ring kernels), and the method of Bachoc et al. (2020). using the multiple kernel (multiple original) in a simulation with 1000 replications for the non-Gaussian random fields. The number of observations n is 100, 500, 1000 or 2000 , and the dimension of random fields is $p = 3$.

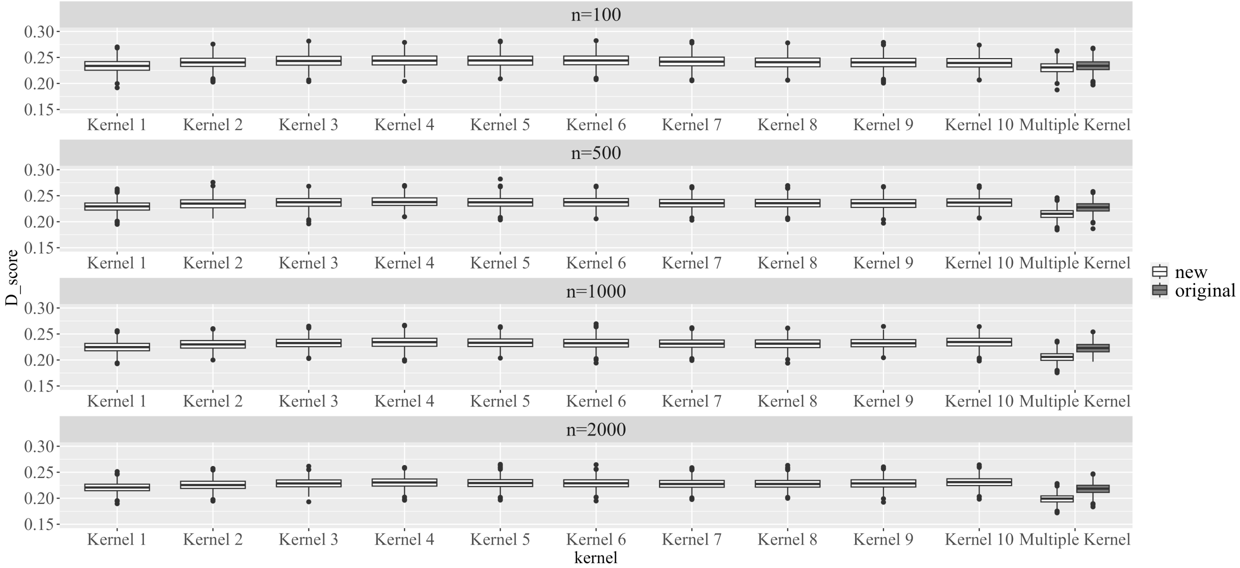

Fig. 2.3 Boxplots of $D(\Omega, \hat{\Omega})$ for the proposed method using the 10 kernels (new) in (2.24), or each of those 10 kernels (Kernel 1, $\cdots$, Kernel 10), and the method of Bachoc et al. (2020) using the 10 kernels (original) in a simulation with 1000 replications for the Gaussian random fields. The number of observations n is 100, 500, 1000 or 2000 (from top to bottom), and the dimension of random fields is $p = 50$.

Kernel	1	2	3	4	5	6	7	8	9	10	Multiple(new)	Multiple(original)
n=100	0.2337	0.2404	0.2433	0.2438	0.2442	0.2442	0.2418	0.2408	0.2405	0.2394	0.2308	0.2339
n=500	0.2295	0.2348	0.2375	0.2378	0.2373	0.2377	0.2356	0.2356	0.2355	0.2369	0.2153	0.2276
n=1000	0.2247	0.2300	0.2326	0.2343	0.2331	0.2323	0.2313	0.2313	0.2321	0.2346	0.2059	0.2228
n=2000	0.2207	0.2254	0.2285	0.2303	0.2293	0.2288	0.2275	0.2275	0.2286	0.2310	0.1993	0.2184

Table 2.3 Median of $D(\Omega, \hat{\Omega})$ from the proposed method using the 10 single kernels, or multiple kernel(including all 10 ring kernels), and the method of Bachoc et al. (2020). using the multiple kernel (multiple original) in a simulation with 1000 replications for the Gaussian random fields. The number of observations n is 100, 500, 1000 or 2000 , and the dimension of random fields is $p = 50$.

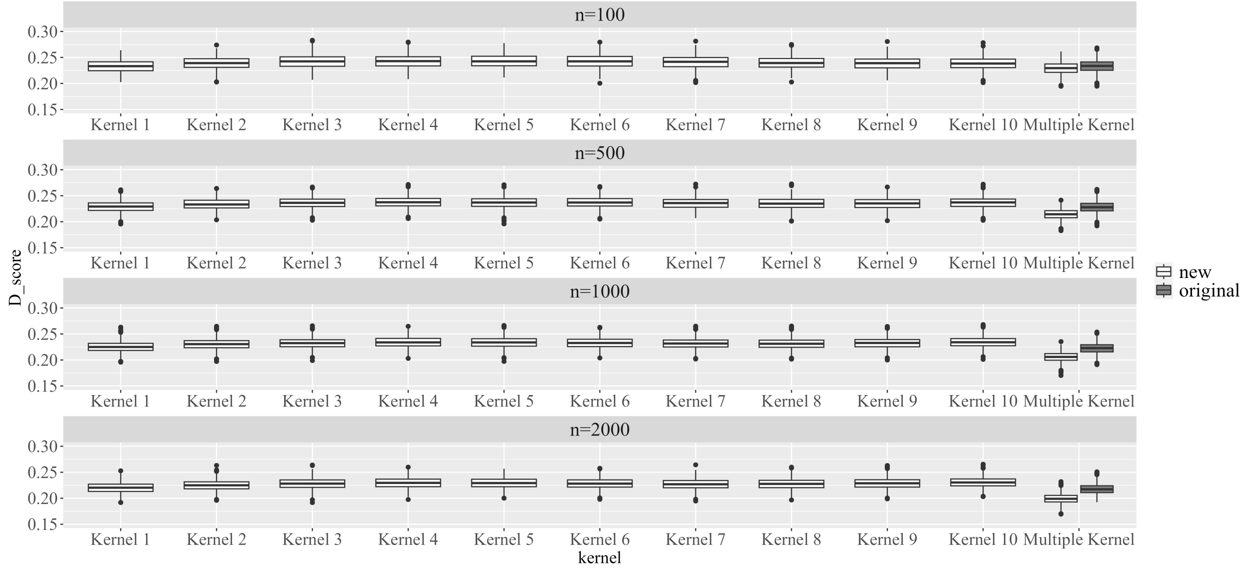

Fig. 2.4 Boxplots of $D(\Omega, \hat{\Omega})$ for the proposed method using the 10 kernels (new) in (2.24), or each of those 10 kernels (Kernel 1, $\cdots$, Kernel 10), and the method of Bachoc et al. (2020) using the 10 kernels (original) in a simulation with 1000 replications for the non-Gaussian random fields. The number of observations n is 100, 500, 1000 or 2000 (from top to bottom), and the dimension of random fields is $p = 50$.

Kernel	1	2	3	4	5	6	7	8	9	10	Multiple(new)	Multiple(original)
n=100	0.2332	0.2391	0.2425	0.2429	0.2425	0.2425	0.2417	0.2393	0.2390	0.2383	0.2295	0.2336
n=500	0.2292	0.2331	0.2363	0.2374	0.2369	0.2369	0.2359	0.2348	0.2352	0.2372	0.2143	0.2278
n=1000	0.2250	0.2305	0.2324	0.2338	0.2338	0.2327	0.2317	0.2312	0.2328	0.2341	0.2059	0.2228
n=2000	0.2203	0.2249	0.2281	0.2296	0.2292	0.2281	0.2269	0.2277	0.2288	0.2303	0.1990	0.2172

Table 2.4 Median of $D(\Omega, \hat{\Omega})$ from the proposed method using the 10 single kernels, or multiple kernel(including all 10 ring kernels), and the method of Bachoc et al. (2020). using the multiple kernel (multiple original) in a simulation with 1000 replications for the non-Gaussian random fields. The number of observations n is 100, 500, 1000 or 2000 , and the dimension of random fields is $p = 50$.

The estimated eigengaps for the proposed method for the Gaussian random fields are presented in Figures 2.5 and 2.6. As n increases, the eigengap also increases. Under low-dimensional setting $p = 3$, the estimates based on single kernel f_1 entail the largest eigengaps and the smallest estimation errors $D(\Omega, \hat{\Omega})$ (see also Theorem 2.3.2). However when $p = 50$, using the multiple kernels leads to the largest eigengaps and the smallest estimation errors.

The patterns with the non-Gaussian random fields are similar and not reported here to save space.

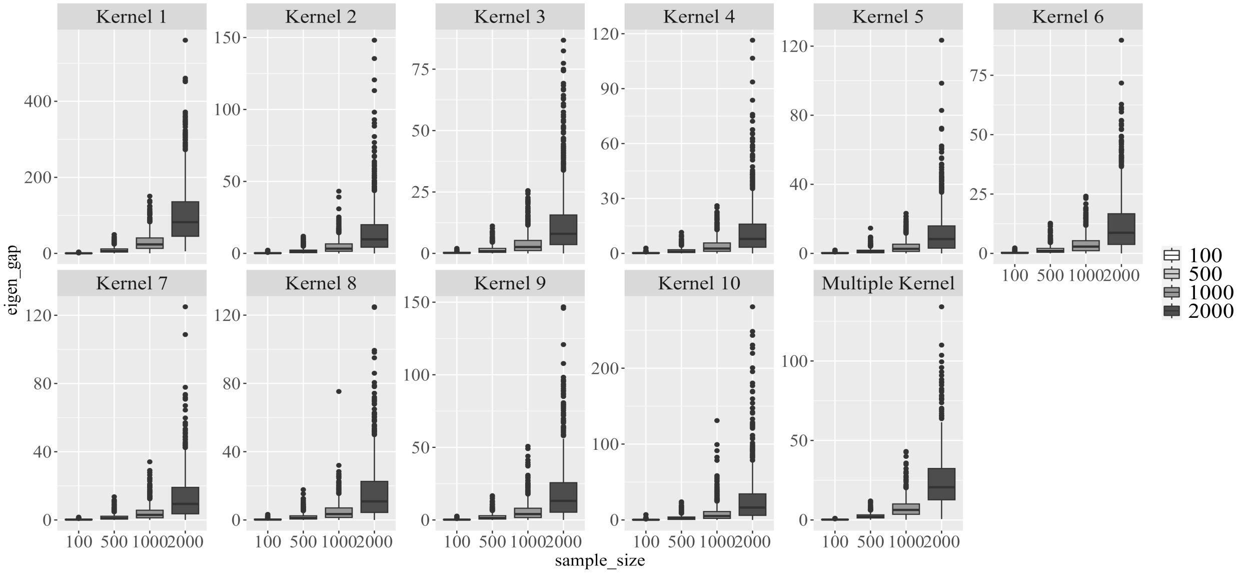

Fig. 2.5 Boxplots of the estimated eigengaps of the proposed method using the 10 kernels (Multiple kernels) in (2.24), or each of those 10 kernels (Kernel 1, $\cdots$, Kernel 10) for the Gaussian random fields. Number of observations n is set at 100, 500, 1000 and 2000, the dimension of random fields is $p = 3$.

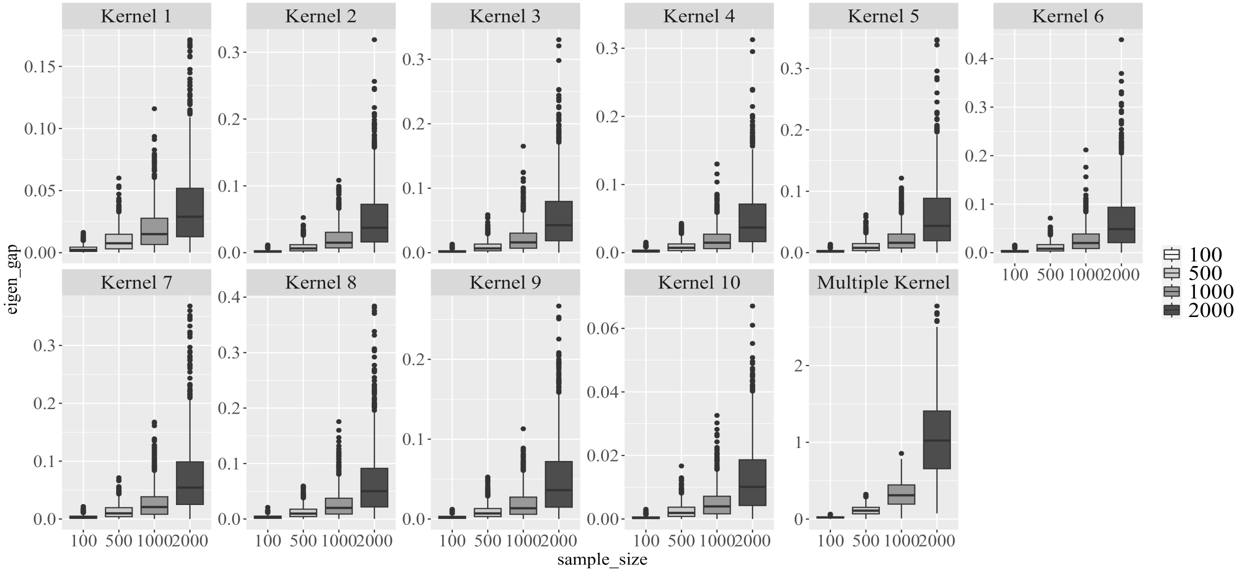

Fig. 2.6 Boxplots of the estimated eigengaps of the proposed method using the 10 kernels (Multiple kernels) in (2.24), or each of those 10 kernels (Kernel 1, $\cdots$, Kernel 10) for the Gaussian random fields. Number of observations n is set at 100, 500, 1000 and 2000, the dimension of random fields is $p = 50$.

2.4.2 A Real Data Example

We apply the proposed method to the moss data from the Kola project in the R package `StatDa` (See P. Filzmoser and M. P. Filzmoser (2015)). The data consists of chemical elements discovered in terrestrial moss at the 594 locations in northern Europe; see the map in Fig.D.1 of Bachoc et al. (2020). More information on the data is presented in Reimann et al. (2011). Following the lead of Nordhausen et al. (2015) and Bachoc et al. (2020), we apply the so-called isometric-log-ratio transformation to the 31 compositional chemical elements in the data. The transformed data are used in our analysis with $n = 594$ and $p = 30$. We standardize the data first such that the sample mean is 0 and the sample variance is I_{30}.

We apply the proposed estimation method with 10 kernels specified as in (2.24). The scores of the first six independent components (IC), corresponding to the six largest eigenvalues of $\hat{W}$ (see Table 2.5), are plotted in Figure 2.8; showing some interesting spatial patterns. For example, the 1st IC can be viewed as a contrast between the locations in the west and those in the east, and the 2nd IC is that between the north and the south. To check if the proposed estimation method has effectively removed the correlation between components, we visualized the correlation matrix of the original dataset and the processed dataset, presented in Figure 2.9. As the figure shows, the correlation matrix on the right is blank for non-diagonal elements, suggesting that there are no correlation between any two different components. On the left, the pairwise correlation is easily observable among almost any two different components. This figure clearly displayed the ability of our proposed method to remove dependence among components.

Figure 2.10 displays the absolute correlation coefficients between the first twelve ICs and those obtained in Nordhausen et al. (2015) which was referred as ‘gold standard’ by Bachoc et al. (2020). While the ICs derived from the two methods differ from each other, the two sets of ICs correlate with each other significantly. For example the correlation between the 1st IC derived from our new method and the 2nd IC obtained in Nordhausen et al. (2015) is 0.92. Note that the ‘gold standard’estimation was obtained using the kernel specified with the relevant subject knowledge. In contrast our estimation is based on the multiple kernels defined generically in (2.24).

i	1	2	3	4	5	6
$\hat{\lambda}_i$	1136.50	877.59	444.21	161.34	126.16	81.13

Table 2.5 The six largest eigenvalues of $\hat{W}$ (with $k = 10$) for the real data example.

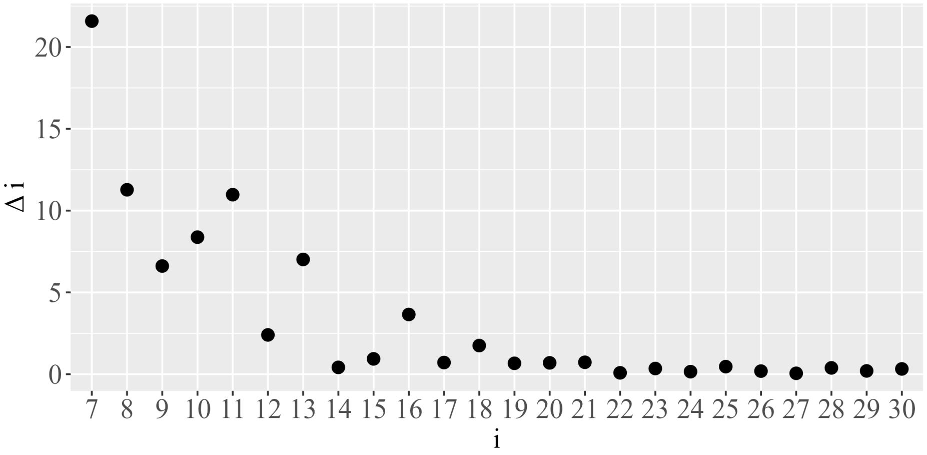

Fig. 2.7 The estimated eigengaps $\Delta_i = \hat{\lambda}_{i-1} - \hat{\lambda}_i$ for $i = 7, \cdots, 30$ on real data example from proposed method with multiple kernel.

The six largest eigenvalues of $\hat{W}$ are listed in Table 2.5. The eigengaps $\Delta_i = \hat{\lambda}_{i-1} - \hat{\lambda}_i$ for $i = 7, \cdots, 30$ are plotted in Figure 2.7. It is clear that the eigengaps among the 13 largest eigenvalues are large. Based on Theorem 2.3.1, we have

$$\hat{\Omega}^{-1}\Omega = \widehat{\Gamma}_{\Omega} = \begin{pmatrix} \hat{\Gamma}_{\Omega,aa} & \hat{\Gamma}_{\Omega,ab} \\ \hat{\Gamma}_{\Omega,ba} & \hat{\Gamma}_{\Omega,bb} \end{pmatrix}, \tag{2.25}$$

where $\hat{\Gamma}_{\Omega,aa}$ is a 12×12 matrix satisfying $\|\hat{\Gamma}_{\Omega,aa} - I_{12}\| = O_p(n^{-1/2}p^{1/2})$. Theorem 2.3.1 also shows that $\|\hat{\Gamma}_{\Omega,ab}\| = O_p(n^{-1/2}p^{1/2})$, $\|\hat{\Gamma}_{\Omega,ba}\| = O_p(n^{-1/2}p^{1/2})$ and $\|\hat{\Gamma}_{\Omega,bb}\| = 1 + O_p(n^{-1/2}p^{1/2})$. Thus, we are reasonably confident that the estimated first 12 ICs are reliable. Moreover, we rewrite $\hat{\Omega}^\top\hat{\Omega}$ as

$$\hat{U}_W^\top \hat{\Sigma} \hat{U}_W = \hat{\Omega}^\top \hat{\Omega} = \begin{pmatrix} \hat{\Omega}_{aa} & \hat{\Omega}_{ab} \\ \hat{\Omega}_{ba} & \hat{\Omega}_{bb} \end{pmatrix}, \tag{2.26}$$

where $\hat{\Omega}_{aa}$ is a 12×12 matrix. We gain $tr(\hat{\Omega}_{aa}) = 6.62$ and $tr(\hat{\Omega}^{\top}\hat{\Omega}) = 8.89$ by calculating. Thus, the major variation of the 30 variables are largely reflected by the 12 largest ICs.

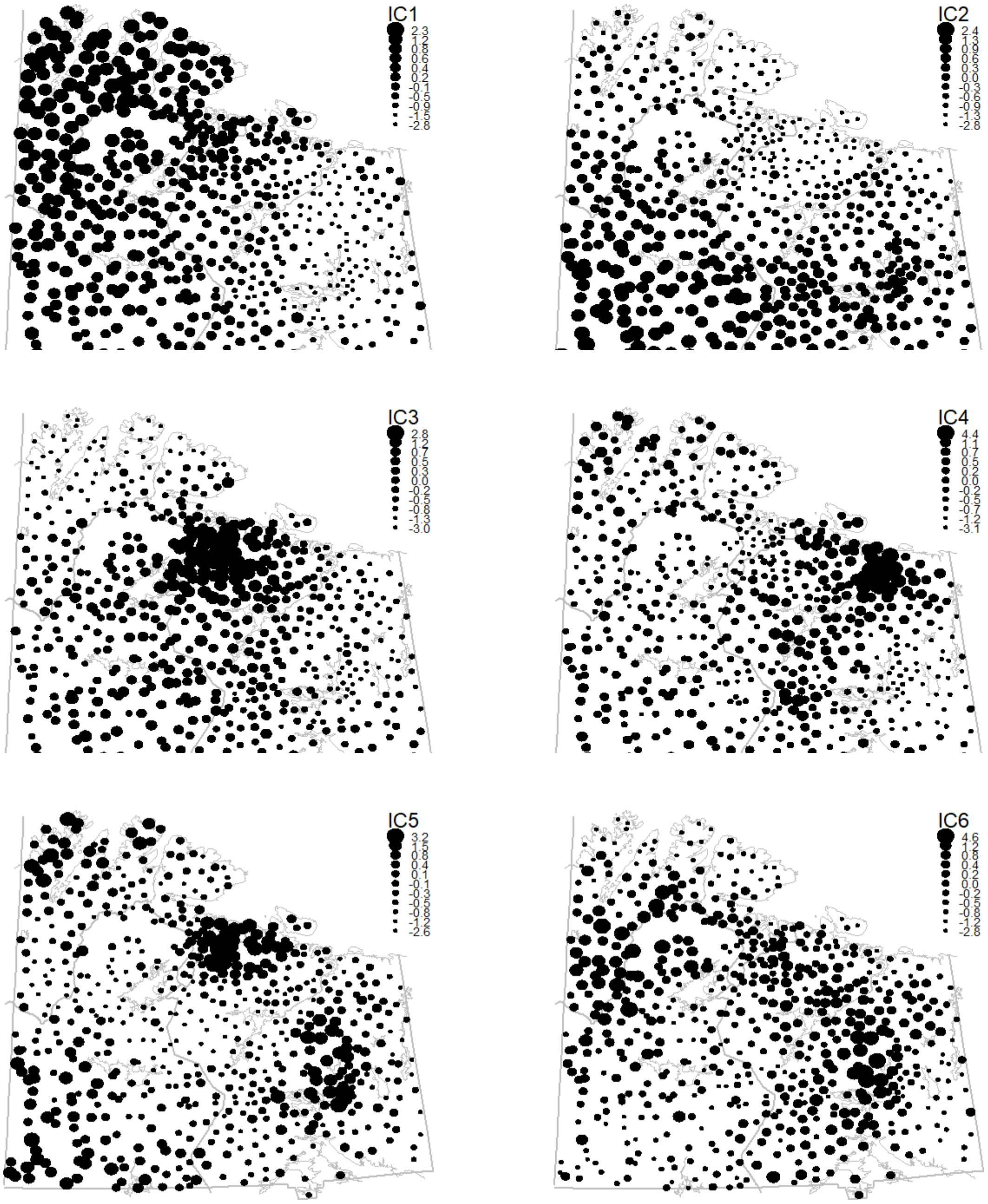

Fig. 2.8 The scores of the first six independent components over the 594 observation locations.

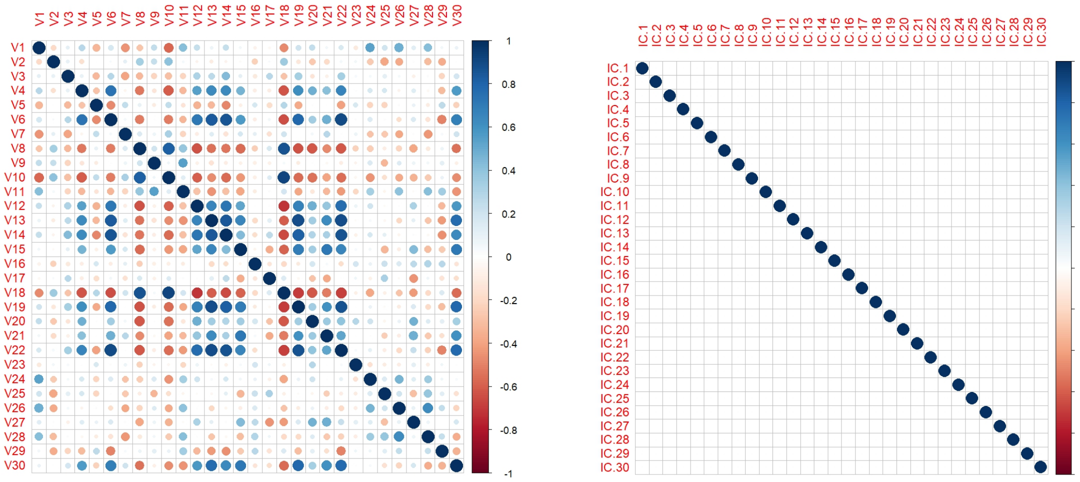

Fig. 2.9 Correlation Matrix between 30 components over 594 observation locations. The unprocessed data is on the left, and the BSS Over Space processed data is on the right.

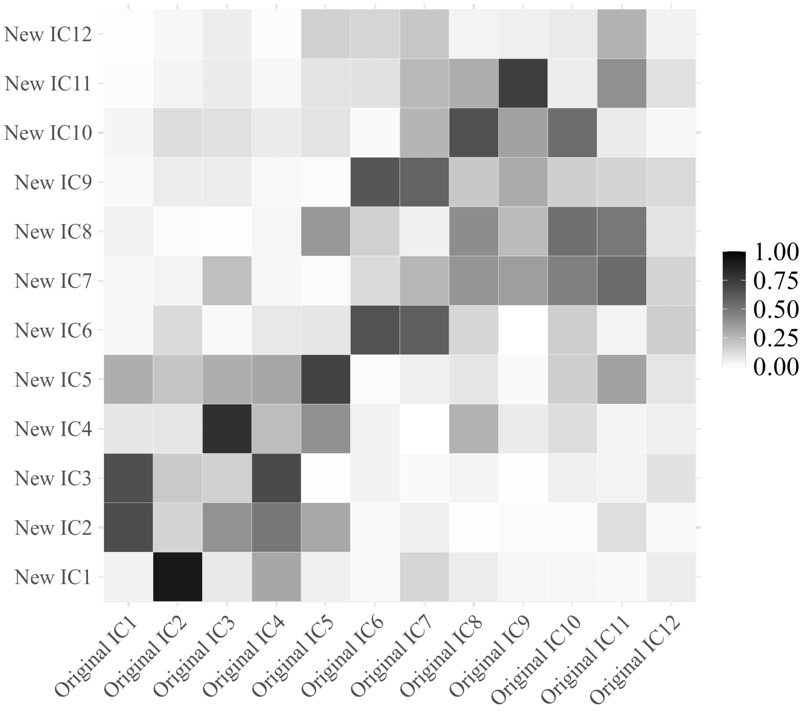

Fig. 2.10 The absolute correlation coefficients between the first 12 independent components derived from the proposed method (New) and those obtained in Nordhausen et al. (2015) (Original).

Supplementary Material

We provide the proofs of Theorems 1-2 in Appendix A.1. We also provide an additional example for simulations in Appendix A.2.

Chapter 3

Permutation Tests for Identifying Number of Factors for High-Dimensional Time Series

3.1 Introduction

With the technological advancements over the past decade, high-dimensional time series data have become increasingly important and accessible. This type of data is now prevalent across various industries, including finance, climate science, biology, and many others. Given the high dimensionality p, traditional multivariate methods may not be suitable for analyzing such large-scale data. When p is large, the number of parameters required by traditional tools, such as vector autoregressive models, can grow on the order of p^2. As a result, dimension reduction becomes essential. To address this challenge, numerous methods have been developed, which can be broadly classified into two categories: summarizing and extracting. An example of a summarizing method is the Principal Component Analysis (PCA) method, proposed by Wold, Esbensen, and Geladi (1987). PCA reduces dimensionality by transforming the data and identifying the first few principal components that capture the majority of the variability within the dataset. While summarizing methods like PCA effectively compress the data and reduce its dimensionality, the interpretability of the resulting components can be challenging, as these components may not have a clear or direct interpretation.

On the other hand, the second class of dimension reduction methods is based on the idea that the variation across p variables can be modeled by a smaller set of underlying variables. These methods assume the existence of a latent structure and reduce the dimensionality of the data by uncovering this low-rank structure. A typical approach in this category is factor modeling, where the p observed variables are assumed to be linear combinations of unobserved latent components, referred to as factors. For instance, in finance, Fama and French (1993) introduced a three-factor model for asset pricing, and subsequently, other asset pricing models with different numbers of factors have been proposed, such as the five-factor model introduced by Fama and French (2015). Other examples of factor modeling can be found in psychology (e.g. personality tests) and genetics research.

While many factor models determine the number of factors based on prior knowledge (e.g. the three-factor model in Fama and French (1993)), there are situations where little is known about the underlying latent structure, and the true number of factors must be identified solely from the observations. In such cases, various methods have been developed. Bai and Ng (2002) proposed a model selection approach, where the number of factors is estimated by solving an optimization problem, incorporating penalty terms to prevent overfitting. Another frequently used approach involves examining the rank of the factor loading matrix, which captures the linear relationship between factors and observed variables. For instance, Lam and Yao (2012a) introduced a ratio-based estimator for determining the number of factors, which is derived through eigenanalysis of a nonnegative definite matrix.

In this chapter, to address the limitations of ratio-based estimators, we introduce a new estimator based on the concept of permutation testing. Our approach follows the factor model framework proposed by Lam and Yao (2012a).

The remainder of the chapter is organized as follows: Section 3.2 provides a detailed description of the factor model and introduces the concept behind our proposed estimator. Section 3.3 explains the Permutation Testing procedure and discusses its significance level. Section 3.4 presents simulation results that demonstrate the advantages of our estimator over the ratio-based estimator proposed by Lam and Yao (2012a) across five different settings. Section 3.5 presents an analysis of real data using our proposed estimator. Finally, Section 3.6 concludes the chapter and outlines potential directions for future research.

3.2 Factor Model Theoretical Framework

3.2.1 Introduction to Factor Model

Let $y_t, t \geq 1$, be a $p \times 1$ time series. In factor model, it is assumed that y_t consists of 2 components: a dynamic time series x_t with lower dimension, which is called the *factors*, and a static component ε_t, which is seen as *noise*. More precisely, we assume:

$$\underset{(p\times 1)}{y_t} = \underset{(p\times r)}{\mathbf{A}}\underset{(r\times 1)}{x_t} + \underset{(p\times 1)}{\varepsilon_t}, \tag{3.1}$$

where x_t represents the latent factor time series with dimension r, and r denotes the number of latent factors, which is also unknown, and is independent of p and n. $\mathbf{A}$ represents the unknown factor loading matrix, and $\varepsilon_t \sim \mathrm{WN}(\mu_\varepsilon, \Sigma_\varepsilon)$ is a vector white-noise process with no temporal correlation. Under this model, the dynamics of y_t is driven by that of the latent process x_t with much smaller dimension, i.e. $r \ll p$. In such case, the decomposition of y_t using factor model can be seen as an effective dimension reduction procedure.

In model (3.1), we could only observe y_t, and there is no information on how changes in factors x_t are reflected on observation y_t, which is described in the factor loading matrix $\mathbf{A}$. The extent of influence from factors to observation via factor loading matrix is called **factor strength**. A factor is typically considered "strong" if the norm of its corresponding factor loading vector in factor loading matrix $\mathbf{A}$ is large, which means the factor explains a significant portion of the variance of the observed data. Conversely, a "weak" factor has a smaller factor loading, and thus, its contribution to the overall variance is minimal. The strength of a factor is quantified by its loading matrix. We define **factor strength** in the following way:

Definition 3.2.1. ***Factor Strength:***

Let $\mathbf{a} \asymp \mathbf{a}$ *if* $\mathbf{a} = O(\mathbf{a})$ *and* $\mathbf{a} = O(\mathbf{a})$*, assume that for* $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_r)$*,*

$$\|\mathbf{a}_j\|_2^2 \asymp p^{1-\delta_j}, \quad j = 1, \ldots, r,$$

where $\delta_j \in [0,1]$ *is a measure of strength of factor* $x_{t,j}$*.*

If the factor loading vector has a large magnitude, meaning that $\delta_j = 0$, the corresponding factor $x_{t,j}$ will have the strongest influence on y_t via factor loading matrix $\mathbf{A}$, and we call $x_{t,j}$ **strong factor**. When $\delta_j > 0$, we call the corresponding $x_{t,j}$ as **weak factor**. The smaller δ_j indicates the stronger factor strength of $x_{t,j}$. The key to understanding factor strength is the relative comparison of factors' contributions. Though by definition, a factor $x_{t,j}$ is not strong unless $\delta_j = 0$, yet it is not necessarily "weak" in an absolute sense. Whether it is weak or not depends on the specific context and the magnitude of the δ_j relative to other factors. $\delta > 0$ suggests that the factor contributes less compared to a factor with $\delta = 0$. However, it still could explain some portion of the variance in the data.

For the factor loading matrix $\mathbf{A}$, there exists another challenge related to identifiability. On the right-hand side of model (3.1), the pair $(\mathbf{A}, x_t)$ can be replaced by $(\mathbf{AH}, \mathbf{H}^{-1}x_t)$ for any invertible matrix $\mathbf{H}$, while y_t remains unchanged. Since x_t is not observable, the factor loading matrix $\mathbf{A}$ is not uniquely identifiable. To overcome this issue, one common approach is to estimate the $r-$dimensional linear space $\mathscr{M}(\mathbf{A})$ spanned by columns of $\mathbf{A}$, which is uniquely defined. For any invertible matrix $\mathbf{H}$, we have $\mathscr{M}(\mathbf{A}) = \mathscr{M}(\mathbf{AH})$. Based on this, we may set the following assumption on factor loading matrix $\mathbf{A}$:

Assumption 3.2.1. *Factor loading matrix* $\mathbf{A}$ *has rank r, and* $\mathbf{A}^\top \mathbf{A} = \mathbf{I}_r$*.*

Assumption 3.2.1 can be satisfied easily. If the rank of $\mathbf{A}$ is not r, model 3.1 would be expressed in a different r', where $r' \neq r$ instead. The second part of Assumption 3.2.1 is for the uniqueness of the factor loading matrix and factor components. Since the pair $(\mathbf{A}, x_t)$ can be replaced by $(\mathbf{AH}, \mathbf{H}^{-1}x_t)$ for any invertible matrix $\mathbf{H}$, we can always find an $\mathbf{H}$, such that $(\mathbf{AH})^\top(\mathbf{AH}) = \mathbf{I}_r$. Thus, having Assumption 3.2.1, we could obtain a unique factor loading matrix $\mathbf{A}$, therefore defining a unique factor process x_t as well.

3.2.2 Estimating Factor Loading Matrix A and Number of Factors r

Our objective is to identify the number of latent factors r. The main idea is to utilize the information from decomposing the covariance matrix of $\mathbf{Y}$ via eigenanalysis. First, the following assumptions are made:

Assumption 3.2.2. *x_t is weakly stationary.*

Assumption 3.2.3. *$Cov(x_t, \varepsilon_{t+k}) = 0$ for any $k \geq 0$.*

Assumption 3.2.2 on factors x_t is common among literature for factor modeling on time series, and Assumption 3.2.3 assumes that future noise has no correlation to factors at current time t. Define the covariance matrix for y_t, x_t, ε_t in model (3.1) in the following form:

$$\begin{aligned}\mathbf{\Sigma}_y(k) &= Cov(y_{t+k}, y_t), \quad \mathbf{\Sigma}_x(k) = Cov(x_{t+k}, x_t), \\ \mathbf{\Sigma}_{x\varepsilon}(k) &= Cov(x_{t+k}, \varepsilon_t)\end{aligned} \tag{3.2}$$

From (3.2) and factor model (3.1) above, we could build connection among covariance matrices of observation y_t, the latent factors x_t and noise ε_t as:

$$\mathbf{\Sigma}_y(k) = \mathbf{A}\mathbf{\Sigma}_x(k)\mathbf{A}^\top + \mathbf{A}\mathbf{\Sigma}_{x\varepsilon}(k), \quad k \geq 1. \tag{3.3}$$

To gather information from covariance matrix of $\mathbf{Y}$ over multiple lags (up to predetermined lag m), we would need to prevent cancellation of information among them, thus we introduce the following non-negative square matrix:

$$\mathbf{M} = \sum_{k=1}^{m} \mathbf{\Sigma}_y(k)\mathbf{\Sigma}_y(k)^\top, \quad m \geq 1. \tag{3.4}$$

Further, define matrix $\mathbf{B}$ to be an $p \times (p-r)$ orthogonal matrix that, combining columns of the factor loading matrix $\mathbf{A}$ and columns of $\mathbf{B}$ would create a $p \times p$ orthogonal matrix. Hence, $\mathbf{B}^\top\mathbf{B} = \mathbf{I}_{p-r}$ and $\mathbf{A}^\top\mathbf{B} = 0$. Following from (3.2) and the definition of matrix $\mathbf{B}$, we could reach the result that:

$$\begin{aligned}\mathbf{MB} &= \sum_{k=1}^{m} \boldsymbol{\Sigma}_y(k)[\mathbf{A}\boldsymbol{\Sigma}_x(k)\mathbf{A}^\top + \mathbf{A}\boldsymbol{\Sigma}_{x\varepsilon}(k)]^\top \mathbf{B} \\ &= \sum_{k=1}^{m} \boldsymbol{\Sigma}_y(k)[\boldsymbol{\Sigma}_x(k)\mathbf{A}^\top + \boldsymbol{\Sigma}_{x\varepsilon}(k)]^\top \mathbf{A}^\top \mathbf{B} \\ &= 0. \end{aligned} \tag{3.5}$$

Note that matrix **B** is orthogonal, therefore $\mathbf{MB} = 0$ implies columns of **B** can be seen as eigenvectors of **M** corresponding to zero-eigenvalues. Since matrix **B** has $(p-r)$ columns, we could argue that **M** has r non-zero eigenvalues, which matches the number of columns for matrix **A**. Therefore, we could reach the following conclusion:

> Eigenvectors of **M**, which corresponds to non-zero eigenvalues, can be taken as columns of factor loading matrix **A**. The uniquely defined linear space $\mathscr{M}(\mathbf{A})$ is spanned by eigenvectors of **M** which corresponds to non-zero eigenvalues.

Note that $k = 0$ is not included in equation (3.3). Because at $k = 0$, $Cov(\varepsilon_{t+k}, \varepsilon_t) = Var(\varepsilon_t) \neq 0$, and we would have the following result instead:

$$\boldsymbol{\Sigma}_y(k) = \mathbf{A}\boldsymbol{\Sigma}_x(k)\mathbf{A}^\top + \mathbf{A}\boldsymbol{\Sigma}_{x\varepsilon}(k) + Cov(\varepsilon_{t+k}, \varepsilon_t),$$

which would make $\mathbf{MB} \neq 0$. Consequently, equation (3.5) would be invalid, which is not desirable.

To estimate the number of factors r, we perform eigendecomposition on sample version of **M**:

$$\widehat{\mathbf{M}} = \sum_{k=1}^{m} \widehat{\boldsymbol{\Sigma}}_y(k)\widehat{\boldsymbol{\Sigma}}_y(k)^\top, \quad m \geq 1, \tag{3.6}$$

where,

$$\widehat{\boldsymbol{\Sigma}}_y(k) = \frac{1}{n-k}\sum_{t=1}^{n-k}(y_{t+k} - \bar{y})(y_t - \bar{y}), \quad \bar{y} = \frac{1}{n}\sum_{t=1}^{n} y_t. \tag{3.7}$$

With the eigenanalysis approach, the most direct way of estimating number of factors r would be to count the number of non-zero eigenvalues of $\mathbf{M}$. However, due to randomness within $\widehat{\mathbf{M}}$, the zero eigenvalues of $\mathbf{M}$ might not be exactly 0 in $\widehat{\mathbf{M}}$. To overcome this issue, one popular approach is to list out all estimated eigenvalues of $\widehat{\mathbf{M}}$ in the descending order, and find the position where the estimated eigenvalues is significantly closer to 0 than the previous eigenvalue. Based on this idea, some methods for estimating number of factors have been proposed. For example, Lam and Yao (2012a) proposed a ratio-based estimator:

$$\hat{r}_{Ratio1} = \arg\min_{1 \leq i \leq R} \frac{\hat{\lambda}_{i+1}}{\hat{\lambda}_i}, \tag{3.8}$$

where$\hat{\lambda}_1 \geq \ldots \geq \hat{\lambda}_p$ are eigenvalues from $\widehat{\mathbf{M}}$, and $r < R < p$ is a pre-specified constant. They look for the minimum of ratios of eigenvalues to identify the first zero eigenvalue. This method has a strong performance in general, yet when factors are weak and factor strength level varies ($\delta_i \neq \delta_j, i \neq j$), the factor loading matrix $\mathbf{A}$ carries only limited amount of information about factors x_t, it might not be able to provide good estimations consistently.

To be more specific, for their proposed estimator in (3.8) to perform, Lam and Yao (2012a) sets the condition of $\delta_j = \delta$ for all $j \in (1, \ldots, r)$, which means all r factors have same strength. In the case of multiple factor strength levels, they updated the factor model as

$$y_t = \mathbf{A}x_t + \varepsilon_t = \mathbf{A}_1 x_t^{(1)} + \mathbf{A}_2 x_t^{(2)} + \varepsilon_t, \tag{3.9}$$

where $x_t^{(1)}$ consists of r_1 factors of same strength $\delta^{(1)}$, and $x_t^{(2)}$ consists of r_2 factors of same strength $\delta^{(2)}$. Based on this model, they proposed a two-step ratio test. The estimation procedure is described as:

1. Use $\hat{r}_{Ratio1}$ to obtain an estimation on number of factors $\hat{r}_1$ and corresponding factor loading matrix $\hat{\mathbf{A}}_1$.

2. Perform the ratio-based estimation again on $y_t^* = y_t - \hat{\mathbf{A}}_1 \hat{\mathbf{A}}_1^\top y_t$ to obtain another estimation on number of factors $\hat{r}_2$.

3. The two-step estimation on the total number of factors $\hat{r}_{Ratio2} = \hat{r}_1 + \hat{r}_2$.

The two-step approach is suitable for the case where factors have two factor strength levels. However, choosing between the one-step and two-step method requires prior knowledge. If there is only one factor strength level and we used the two-step method, while $\hat{r}_2$ shall be 0, the minimum output from a ratio-based estimator is 1, thus $\hat{r}_{Ratio2}$ will overestimate the true number of factors. We conclude that, the one step estimator $\hat{r}_{Ratio1}$ could only work if $\delta_j = \delta$ for all $j \in (1,\dots,r)$, while the 2 step estimator $\hat{r}_{Ratio2}$ only works if $\exists\, i, j \in (1,\dots,r)\, s.t.\, \delta_i \neq \delta_j, i \neq j$.

The prior knowledge required on factor strength to select the better model is not observable. We propose a new method which does not require knowledge on factor strength. Define:

$$\mathbf{\Gamma} = (\mathbf{v}_1,\dots,\mathbf{v}_p), \tag{3.10}$$

where $(\mathbf{v}_1,\dots,\mathbf{v}_p)$ are eigenvectors of $\mathbf{M}$ in descending order of corresponding eigenvalues. Since matrix Γ includes all eigenvectors of $\mathbf{M}$, it can be seen as a combination of the factor loading matrix $\mathbf{A}$ and matrix $\mathbf{B}$, as eigenvectors corresponding to non-zero and zero eigenvalues are included. Further, define:

$$z_t = \mathbf{\Gamma}^\top y_t. \tag{3.11}$$

Ignoring the error term, using $\mathbf{\Gamma}$ as the factor loading matrix to recover the factors would give us a vector time series z_t with p components, where both factors and noise are included. Since only the first r eigenvalues of $\widehat{\mathbf{M}}$ are non-zero, the first r columns of $\mathbf{\Gamma}$ could be seen as an approximation to the factor loading matrix $\mathbf{A}$. Therefore, we would expect the first r components of z_t to display serial dependence, and the remaining $p-r$ components to be white noise. Hence, the estimation on number of factors r becomes the estimation of the number of components in z_t with serial dependence. To test for serial dependence of a time series, we introduce a non-parametric permutation testing procedure in the next session.

3.3 Permutation Testing for Serial Dependence in Time Series

When testing for serial dependence in time series, one common tool is the hypothesis test. Let $Z = (Z_1, \ldots, Z_n)$ be a univariate time series, we define the null and alternative hypotheses as:

- H_0: Z is white noise, with no serial dependence.
- H_A: Z is not white noise, and has serial dependence.

In hypothesis testing, we aim to reject the null hypothesis H_0 in favor of the alternative hypothesis H_A, thereby proving the existence of serial dependence within the tested series, if there is sufficient evidence. Based on this concept, numerous parametric tests have been developed. Some prevalent examples include the work by Box and Pierce (1970) and Ljung and Box (1978). Under null hypothesis, the test statistic for these parametric tests follows a chi-squared distribution. However, these tests typically rely on conditions involving finite moments to construct a valid distribution for the test statistic. A common assumption is finite variance, $E(x^2) < \infty$, as discussed in Box and Pierce (1970) and McLeod (1978). Further, Fisher and Gallagher (2012) introduced a weighted Portmanteau statistic, which, under the assumption $E(x^8) < \infty$, converges to a standard normal distribution after standardization, in addition to the chi-squared distribution result under finite variance. These assumptions are crucial for interpreting the test results and determining the significance level.

In contrast, we propose a non-parametric method that uses an empirical distribution derived from a large number of permutations, making it non-parametric and free from reliance on these moment conditions. This flexibility ensures that our test is robust and can be applied in situations where the assumptions of traditional parametric methods may not hold, offering valid inference without the need for finite moment assumptions. Thus, the permutation test provides a more universally applicable and robust alternative to traditional serial dependence tests.

3.3.1 Permutation Testing Framework

Permutation testing is a well-known non-parametric method that is relatively easy to implement. The core idea is to generate new series from the observed series Z_{obs} via permuting the observations $Z_{obs} = Z_1, \dots, Z_n$, and apply a test statistic $T(\cdot)$ on each of the permuted series. This generates an empirical distribution of the test statistic, from which the p-value is calculated. There have been related studies applying permutation testing to time series data. For example, Romano and Tirlea (2022) proposed a permutation testing procedure and derived the permutation distribution under their framework. In our work, we simplify this setting by focusing on the non-parametric aspects of permutation testing and demonstrate that our proposed procedure can control the Type I error (i.e., the probability of incorrectly rejecting the null hypothesis) at the desired level α.

The goal is to test whether the observed series is white noise. One commonly adopted measure for checking serial dependence is the autocorrelation:

$$\rho_k = \frac{Cov(Z_{t+k}, Z_t)}{Var(Z_t)} = \frac{E[(Z_t - \mu_t)(Z_{t+k} - \mu_{t+k})]}{E(Z_t - \mu_t)^2}, \tag{3.12}$$

where k is the lag k for ρ to be computed on. μ_t is the population mean for Z_t, and μ_{t+k} is the population mean for Z_{t+k}. The sample autocorrelation is defined as:

$$\hat{\rho}_k = \frac{\sum_{t=1}^{n-k}(Z_t - \bar{Z})(Z_{t+k} - \bar{Z})}{\sum_{t=1}^{n}(Z_t - \bar{Z})}, \quad \bar{Z} = \frac{1}{n}\sum_{i=1}^{n} Z_i, \tag{3.13}$$

where $\hat{\rho}_k$ is the sample correlation at lag k, n is the total number of observation, and $\bar{Z}$ is the sample mean. Based on sample autocorrelation, the hypothesis for testing serial dependence is defined as follows:

$$H_0 : \rho_k = 0 \quad \text{for all } k \geq 1, \quad v.s. \quad H_A : \exists k \geq 1\, s.t.\, \rho_k \neq 0. \tag{3.14}$$

Next, we introduce the idea of permutation testing for the above hypothesis. First, we define :

Definition 3.3.1. ***Permutation Function***

Let w be a permutation function defined on $\mathbb{R}^n$. For a time series $Z = (Z_1, \dots, Z_n)$, there exists at most $n!$ unique permutations, denoted by $(w_1, \dots, w_{n!})$. Denote the permuted series by $Z_{w_i} = (Z_{w_i(1)}, \dots, Z_{w_i(n)})$, where $i \in (1, \dots, n!)$.

The permutation testing procedure is as follows: Given a time series $Z = (Z_1, \dots, Z_n)$, let $T(\cdot)$ be a test statistic, w be the permutation function defined above. For each $i \in (1, \dots, n!)$, let $Z_{w_i} = (Z_{w_i(1)}, \dots, Z_{w_i(n)})$ represent a permuted series based on Z. We then calculate the test statistic for each of the $n!$ permuted series, which gives a sequence of test statistics $T(Z_{w_1}), \dots, T(Z_{w_{n!}})$. The p-value of the test is defined as:

$$\hat{p} = \frac{\sum_{i=1}^{n!} \mathbb{1}\{T(Z_{w_i}) \geq T(Z)\}}{n!}, \quad \hat{p} \in [0,1]. \tag{3.15}$$

The p-value is the proportion of the permuted series where its test statistic is as extreme as, or more extreme than the test statistic of the observed series Z. By comparing p-value with the pre-determined significance level α, we decide whether to reject or fail to reject H_0. The test statistic we choose is the weighted Ljung-Box test statistic from Fisher and Gallagher (2012), defined as:

$$T(Z) = n(n+2) \sum_{k=1}^{m} \frac{m-k+1}{m} \frac{\hat{\rho}_k^2}{n-k}, \tag{3.16}$$

where $\hat{\rho}_k$ is the sample autocorrelation of the observed series Z at lag k, and m is the maximal lag we are interested in.

To perform a permutation test, it is essential to verify that the observed series is exchangeable under the null hypothesis. Exchangeability is defined as follows:

Definition 3.3.2. ***Exchangeability of Random Variables***

A sequence of random variables $(Z_1, \dots, Z_n)$ is said to be exchangeable if the joint distribution of $(Z_1, \dots, Z_n)$ is equal to the joint distribution of any random permutation $w_i(Z_1, \dots, Z_n)$:

$$(Z_1, \dots, Z_n) \overset{d}{=} (w_i(Z_1, \dots, Z_n)),$$

where $i \in (1, \ldots, n!)$.

If the observed series to be tested satisfies exchangeability under H_0, then the significance level (i.e., probability of rejecting H_0 when H_0 is true, also known as *Type I Error*) can be controlled at a pre-determined level α. To fulfill this requirement, we make the following assumption:

Assumption 3.3.1. *Under H_0, the variables $(Z_1, \ldots, Z_n)$ within the observed series Z, are independent and identically distributed, and Z is exchangeable.*

Naturally, we would prefer to deplete all possible permutations of Z and perform $n!$ permutations. However, $n!$ is often too large, and calculating the test statistic for all $n!$ permuted series will be computationally expensive. In practice, when $n!$ is too large, we are limited to perform only a smaller number of permutations L, where $L < n!$. The corresponding p-value $\hat{p}_L$ for having L permuted series is defined as:

Definition 3.3.3. ***p-value with the number of permutation smaller than n!:***

For a series $Z = (Z_1, \ldots, Z_n)$, let $T(\cdot)$ be a test statistic, let w be the permutation function, and Z_{w_j} be a permutation on Z, where $j \in 1, \ldots, L$ and $L < n!$. Define the function $\hat{p}_L$ satisfying:

$$\hat{p}_L = \frac{1 + \sum_{j=1}^{L} \mathbb{1}\{T(Z_{w_j}) \geq T(Z)\}}{1 + L}, \quad \hat{p}_L \in [\frac{1}{1+L}, 1].$$

Note that for $\hat{p}_L$, the denominator and the numerator include an additional *"+1"*. his is because the observation itself is treated as one of the permutations. When using a limited number of permutations, $n!$ is typically very large, making it unlikely that any permuted series will be identical to the observed series within the L permutations.

3.3.2 Significance Level for the Permutation Test

The significance level of a hypothesis test refers to the probability of committing a Type I Error, which occurs when we reject the null hypothesis H_0 while it is actually true. Ideally, given a pre-determined significance level $\alpha \in [0, 1]$, the hypothesis testing procedure should control the probability of making a Type I Error at α. In this section, we show that under the null hypothesis H_0, the proposed permutation testing procedure for detecting serial dependence within a single time series controls the significance level at the specified level α.

Theorem 3.3.1. ***Significance Level of Permutation Test***

Let Assumption 3.3.1 hold. For any significance level $\alpha \in [0,1]$, the proposed permutation testing procedure controls probability of committing Type I Error at the desired level α.

When the size of observed series n is large, it's computationally difficult to use all $n!$ possible permutations for the test. Thus it's necessary to study by scenario, where 1) we use all $n!$ permutation of the observed series, and 2) we use $L < n!$ permutations of the observed series. In the following sections, we will show that the proposed permutation testing procedure controls type I error at the desired significance level for both scenarios.

3.3.2.1 Significance Level of Permutation Testing Using $n!$ Permuted Series

In this section, we provide a proof for the significance level of the proposed permutation testing procedure when using all possible permutations of the observed series Z. The main idea of the proof is to construct the empirical distribution of the selected test statistic and evaluate the probability of committing a Type I Error using this constructed empirical distribution. To begin, we define the permutation distribution for any test statistic as follows:

Definition 3.3.4. ***Permutation Distribution***

Let $T : \mathbb{R}^n \to \mathbb{R}$ be a test statistic on the series $Z = (Z_1, \dots, Z_n)$, let $w_1, \dots, w_{n!}$ be the $n!$ permutations of the observed series. Define the permutation distribution of test statistic $t \in \mathbb{R}$ to be:

$$\hat{F}(t) = \frac{1}{n!} \sum_{i=1}^{n!} \mathbb{1}\{T(Z_{w_i}) \leq t\}, \quad \hat{F}(t) : \mathbb{R} \to [0,1].$$

The permutation distribution $\hat{F}_Z(t)$ is constructed based on the empirical distribution of the test statistic, conditional on the observed series $Z = (Z_1, \dots, Z_n)$, and has the following properties:

1. $\hat{F}(t)$ is non-decreasing and right continuous w.r.t test statistic t.

2. $\lim_{t \to -\infty} \hat{F}(t) = 0$.

3. $\lim_{t\to\infty}\hat{F}(t) = 1$.

Therefore, $\hat{F}(t)$ can be seen as a probability distribution function, and we may define its inverse as

$$\hat{F}(t)^{-1}(\alpha) = \inf\{t|\hat{F}(t) \geq \alpha\}, \quad \hat{F}(t)^{-1}(\alpha) : [0,1] \to \mathbb{R},$$

where $\alpha \in [0,1]$ is the pre-specified significance level. Based on the permutation distribution we have constructed above, we could proceed to prove the following lemma:

Lemma 3.3.1. ***Significance Level of Permutation Testing (with $n!$ permuted series)***

For a permutation test using $n!$ permuted series, the probability of committing a type I error (falsely rejecting H_0) is less than or equal to the significance level α:

$$\mathbb{P}(\textit{Rejecting } H_0|H_0 \textit{ True, } Z_1,\ldots,Z_n) \leq \alpha.$$

Proof. Given that H_0 is true, committing a Type I Error is equivalent to rejecting the null hypothesis H_0, which which occurs when the p-value from permutation testing is below the significance level α. Under the condition that the series $Z = (Z_1,\ldots,Z_n)$ is fully observed, the test statistic $T(\cdot)$ for all $n!$ permuted series of Z should follow a uniform distribution. Based on this conclusion, we have:

$$\begin{aligned}
\mathbb{P}(Rejecting H_0|H_0 True) &= \mathbb{P}(\hat{p} \leq \alpha|H_0,Z_1,\ldots,Z_n) \\
&= \mathbb{P}(\frac{1}{n!}\sum_{i=1}^{n!}\mathbb{1}\{T(Z_{w_i}) \geq T(Z)\} \leq \alpha|H_0,Z_1,\ldots,Z_n) \\
&= \mathbb{P}(1-\hat{F}(T(Z)) \leq \alpha|H_0,Z_1,\ldots,Z_n) \\
&= \mathbb{P}(\hat{F}(T(Z)) \geq 1-\alpha|H_0,Z_1,\ldots,Z_n) \\
&\leq \alpha \\
\therefore \quad & \mathbb{P}(\text{Rejecting } H_0|H_0 \text{ True}) \leq \alpha.
\end{aligned}$$

Under the condition that the test statistic is uniform, using the properties of empirical distribution, we could derive that the probability for $T(Z)$ to be larger than or equal to the $(1-$

α)-th quantile of the empirical distribution is less or equal to α, and under Exchangeability Assumption 3.3.1, $T(Z)$ follows $\hat{F}$ unconditionally on $Z_1, \ldots, Z_n$, thus we could draw that condition and finish the proof. □

3.3.2.2 Significance Level of Permutation Testing Using Less Than $n!$ Permuted Series

In this sub-section, we discuss the case where we did not deplete all possible permutation of Z to perform permutation test (i.e., use $L < n!$ permuted series). For the observed series $Z = (Z_1, \ldots, Z_n)$, define permutation distribution when using less than $n!$ permuted series as:

$$\hat{F}_L(t) = \frac{1}{L} \sum_{j=1}^{L} \mathbb{1}\{T(Z_{w_j}) \leq t\}, \quad \hat{F}_L(t) : \mathbb{R} \to [0,1]. \tag{3.17}$$

Let significant permuted series to be those permuted series with test statistic larger than that of the observed series (i.e. $T(Z_{w_i}) \geq T(Z)$). Define the function for finding number of significant permuted series among all permuted series to be:

$$q(Z) := \sum_{i=1}^{n!} \mathbb{1}\{T(Z_{w_i}) \geq T(Z)\}, \tag{3.18}$$

and the function when using less than $n!$ permutations to be:

$$q_L(Z) := \sum_{j=1}^{L} \mathbb{1}\{T(Z_{w_j}) \geq T(Z)\}, \tag{3.19}$$

where $q_L(Z) : \mathbb{R}^n \to (1, \ldots, L)$, and $q(Z) : \mathbb{R}^n \to (1, \ldots, n!)$. The function q and q_L gives the number of significant permuted series. Under H_0, we may derive the distribution of q, as shown in the following proposition:

Lemma 3.3.2. ***Distribution of*** q ***under*** H_0

Given an observed series $Z = (Z_1, \ldots, Z_n)$*, under* H_0*, for any integer* $g \in (1, \ldots, n!)$*,*

$$\mathbb{P}(q(Z) = g) = \frac{1}{n!}. \tag{3.20}$$

Lemma 3.3.2 shows that, the probability of finding exactly g significant permuted series is a constant, regardless of the value that g takes. Hence we can say that $q(Z)$ is uniformly distributed on $[1,n!]$.

For Lemma 3.3.2 to hold, we need to introduce an assumption on uniqueness of the test statistic in permuted series.

Assumption 3.3.2. ***Uniqueness of Test Statistic***
Let $T(\cdot)$ be any sample-autocorrelation-based test statistics, For any pair of different permutations on the observed time series:

$$T(Z_{w_i}) \neq T(Z_{w_h}), \quad i,h \in (1,\ldots,n!), \quad i \neq h,$$

This uniqueness assumption ensures that the test statistic for different permuted series will be distinct, meaning no two distinct permuted series will have identical test statistics. This assumption is likely to hold for any test statistic based on the sample autocorrelation $\hat{\rho}_k$, because permuting the series changes its autocorrelation structure, thereby altering $\hat{\rho}_k$. The numerator of $\hat{\rho}_k$, as given in (3.13) directly links to the temporal ordering of the series. When the series is permuted, the values at time t no longer interact with those at time $t+k$in the same way, leading to different values for $\hat{\rho}_k$. Therefore, Assumption 3.3.2 is likely to hold.

Now, we are ready to prove Lemma 3.3.2:

Proof. Write out LHS of equation (3.20) using all permuted series of the observed series:

$$\begin{aligned}
\mathbb{P}(q(Z)=g) &= \mathbb{P}(\frac{1}{n!}\sum_{i=1}^{n!} q(Z_{w_i})=g) && (3.21)\\
&= \frac{1}{n!}\sum_{i=1}^{n!} \mathbb{P}(q(Z_{w_i})=g) && (3.22)\\
&= \frac{1}{n!}\sum_{i=1}^{n!} E(\mathbb{1}\{q(Z_{w_i})=g\}) && (3.23)\\
&= \frac{1}{n!}E(\sum_{i=1}^{n!} \mathbb{1}\{q(Z_{w_i})=g\}) && (3.24)\\
&= \frac{1}{n!}E(\sum_{i=1}^{n!} \mathbb{1}\{\sum_{h=1}^{n!} \mathbb{1}\{T(w_h(Z_{w_i}))\geq T(Z_{w_i})\}=g\}) && (3.25)
\end{aligned}$$

To evaluate the RHS of the last line from above, it's important to construct an interpretation towards the following part:

$$E(\sum_{i=1}^{n!} \mathbb{1}\{\sum_{h=1}^{n!} \mathbb{1}\{T(w_h(Z_{w_i}))\geq T(Z_{w_i})\}=g\}). \tag{3.26}$$

To start with, consider the inner indicator function

$$\sum_{h=1}^{n!} \mathbb{1}\{T(w_h(Z_{w_i}))\geq T(Z_{w_i})\}=g. \tag{3.27}$$

In equation 3.27, $Z_{w_i}, i\in(1,\dots,n!)$ is a permuted series from the observed series Z, and $w_h(Z_{w_i}), h\in(1,\dots,n!)$ is a permutation of the permuted series Z_{w_i}, and we refer to them as twice-permuted series. LHS of equation 3.27 counts the number of significant twice-permuted series, which is defined as $T(w_h(Z_{w_i}))\geq T(Z_{w_i})$. Therefore, equation 3.27 can be interpreted as follows:

> *For each of the permuted series $Z_{w_i}, i\in(1,\dots,n!)$, the number of significant twice-permuted series $w_h(Z_{w_i}), h\in(1,\dots,n!)$, comparing to Z_{w_i}, is equal to g, where $g\in(1,\dots,n!)$.*

Equation 3.27 described an event for a permuted series Z_{w_i} with a pre-determined integer g, and equation 3.26 evaluates the expected number of occurrence of such an event among all possible permuted series $Z_{w_i}, i \in (1,\dots,n!)$. If we can show that equation 3.26 equals 1, then we would have proved Lemma 3.3.2.

For the permuted series and the twice-permuted series from it, under exchangeability Assumption 3.3.1, we have that:

$$w_h(Z_{w_i}) = Z_{w_h}, h, i \in (1,\dots,n!).$$

This result indicates that, the series after 2 permutations $w_h(Z_{w_i})$ is equivalent to directly performing the second permutation on the observed series Z, and therefore a twice-permuted series $Z_{w_h} = w_h(Z_{w_i})$ and the series Z_{w_i} can be seen as 2 series within one set of possible permutations $h, i \in (1,\dots,n!)$ on the observed series Z.

Then, following Assumption 3.3.2 on uniqueness of test statistic for permuted series, we could argue that, for a sequence of permutations $n!$ on the observed series Z, if we order the test statistics of all permuted series in descending order, we could obtain the following series:

$$T(Z_{w_{(1)}}) > T(Z_{w_{(2)}}) > \cdots > T(Z_{w_{(n!)}}). \tag{3.28}$$

where

Now, consider a permuted series $Z_{w_i}, i \in (1,\dots,n!)$ that, g of $n!$ permuted series have test statistic as large or larger than $T(Z_{w_i})$. The question of finding out how many Z_{w_i}s would satisfy the condition of having exactly g series more significant than themselves, is equivalent to finding a position in the sequence 3.28 and inserting $T(Z_{w_i})$ so that there are g series on its left. Under Assumption 3.3.2, since each different permuted series has different test statistic, there will only be only **one** position to insert $T(Z_{w_i})$, for any pre-specified g. Since in (3.28), any $T(Z_{w_{(i)}}), i \in (1,\dots,n!)$ is strictly larger or smaller than others, there will only be one permuted series that satisfies the condition of having g other permuted series in front of itself.

To express this conclusion in mathematical language, we write that

$$\sum_{i=1}^{n!} \mathbb{1}\{\sum_{h=1}^{n!} \mathbb{1}\{T(w_h(Z_{w_i})) \geq T(Z_{w_i})\} = g\} = 1, \tag{3.29}$$

For example, let $g = 2$, which means we are looking for a permuted series Z_{w_i}, $i \in (1,\ldots,n!)$ such that there are exactly 2 series with larger test statistic. Among all possible permutations, there will exist only one unique permutation i satisfying this condition. Now, using this conclusion, we may proceed with the proof:

$$\begin{aligned}
\mathbb{P}(q(Z) = g) &= \frac{1}{n!}E(\sum_{i=1}^{n!} \mathbb{1}\{\sum_{h=1}^{n!} \mathbb{1}\{T(w_h(Z_{w_i})) \geq T(Z_{w_i})\} = g\}) \\
&= \frac{1}{n!}E(\sum_{i=1}^{n!} \mathbb{1}\{\sum_{h=1}^{n!} \mathbb{1}\{T(Z_{w_h}) \geq T(Z_{w_i})\} = g\}) \\
&= \frac{1}{n!} \times 1 \\
&= \frac{1}{n!}
\end{aligned}$$

□

The main objective of this subsection is to discuss the significance level of our proposed permutation testing procedure when only a limited number of permutations is available. To this end, we introduce the following lemma:

Lemma 3.3.3. ***Level of significance for Permutation Testing (Using Less Than $n!$ permuted series)***

Given a series $Z = Z_1,\ldots,Z_n$, under H_0, for a permutation test with limited $L < n!$ series available, the probability of committing a Type I Error (falsely rejecting H_0) is less than or equal to the significance level α:

$$\mathbb{P}(\textit{Type I Error}) \leq \alpha.$$

Based on the definitions and lemmas introduced earlier, we are ready to prove Lemma 3.3.3:

Proof. Under H_0, we have that

$$\begin{aligned}
\mathbb{P}(\text{Rejecting } H_0) &= \mathbb{P}(\hat{p}_L \leq \alpha) \\
&= \mathbb{P}(\frac{1+\sum_{j=1}^{L} \mathbb{1}\{T(Z_{w_j}) \geq T(Z)\}}{1+L} \leq \alpha) \\
&= \mathbb{P}(\sum_{j=1}^{L} \mathbb{1}\{T(Z_{w_j}) \geq T(Z)\} \leq \alpha(1+L)-1) \\
&= \sum_{g=1}^{n!} \mathbb{P}(\sum_{j=1}^{L} \mathbb{1}\{T(Z_{w_j}) \geq T(Z)\} \leq \alpha(1+L)-1 | q(Z)=g) \times \mathbb{P}(q(Z)=g)
\end{aligned}$$

The last line can be evaluated as two parts separately. The latter part $\mathbb{P}(q(Z)=g)$ has been shown in Lemma 3.3.2 that, under H_0, equals to $\frac{1}{n!}$. The remaining part, which is

$$\mathbb{P}(\sum_{j=1}^{L} \mathbb{1}\{T(Z_{w_j}) \geq T(Z)\} \leq \alpha(1+L)-1 | q(Z)=g),$$

can be better understand by explaining it explicitly. Under the limited permuted series setting, for each of the permuted series $Zw_j, j \in 1,\ldots,L$, the probability of it being significant follows a Bernoulli distribution:

$$\mathbb{1}\{T(Z_{w_j}) \geq T(Z)\} \sim Bernoulli(p=\frac{g}{n!}).$$

Further, we impose the condition that out of all $n! = n!$ possible series, there will be g significant series. Then within the limited permuted series $Zw_j, j \in 1,\ldots,L$, the conditional probability of having the number of significant series less than or equal to $\alpha(1+L)-1$, which is the threshold for rejecting H_0, can be seen as sum of Bernoulli random variables, which follows a Binomial distribution:

$$\sum_{j=1}^{L} \mathbb{1}\{T(Z_{w_j}) \geq T(Z)\} \sim Binom(n=L, p=\frac{g}{n!}).$$

Proceeding the proof by using the expression of Binomial distribution instead, we have:

$$\begin{aligned}
\mathbb{P}(\hat{p_L}(Z_1,\ldots,Z_n) \leq \alpha) &= \sum_{g=1}^{n!} \mathbb{P}(\sum_{j=1}^{L} \mathbb{1}\{T(Z_{w_j}) \geq T(Z)\} \leq \alpha(1+L)-1|q(Z)=g) \times \mathbb{P}(q(Z)=g) \\
&= \sum_{g=1}^{n!} \mathbb{P}(X \leq \alpha(1+L)-1) \times \frac{1}{n!} \\
&= \sum_{g=1}^{n!} \frac{1}{n!} \sum_{k=0}^{\alpha(1+L)-1} \binom{L}{k} (\frac{g}{n!})^k (1-\frac{g}{n!})^{L-k} \\
&= \sum_{k=0}^{\alpha(1+L)-1} \frac{1}{n!} \sum_{g=1}^{n!} \binom{L}{k} (\frac{g}{n!})^k (1-\frac{g}{n!})^{L-k} \\
&= \sum_{k=0}^{\alpha(1+L)-1} \binom{L}{k} \frac{1}{n!} \sum_{g=1}^{n!} (\frac{g}{n!})^k (1-\frac{g}{n!})^{L-k}
\end{aligned}$$

Here, if we set $x = \frac{g}{n!}$, then from the perspective of Riemann Sum Approximation, we could obtain the following substitution:

$$\frac{1}{n!} \sum_{g=1}^{n!} x^k (1-x)^{L-k} \approx \int_0^1 x^k (1-x)^{L-k} dx.$$

Thus,

$$\begin{aligned}
\mathbb{P}(\hat{p}_L(Z_1,\dots,Z_n) \leq \alpha) &= \sum_{k=0}^{\alpha(1+L)-1} \binom{L}{k} \frac{1}{n!} \sum_{g=1}^{n!} (\frac{g}{n!})^k (1-\frac{g}{n!})^{L-k} \\
\text{(by Riemann Sum Approximation)} &\approx \sum_{k=0}^{\alpha(1+L)-1} \binom{L}{k} \int_0^1 x^k (1-x)^{L-k} dx, \quad x = \frac{g}{n!} \\
&= \sum_{k=0}^{\alpha(1+L)-1} \binom{L}{k} Beta(z_1 = k+1, z_2 = L-k+1) \\
\text{(by Properties of Beta Function)} &= \sum_{k=0}^{\alpha(1+L)-1} \binom{L}{k} \frac{\Gamma(k+1)\Gamma(L-k+1)}{\Gamma(L+2)} \\
\text{(by Properties of Gamma Function)} &= \sum_{k=0}^{\alpha(1+L)-1} \binom{L}{k} \frac{\Gamma(k+1)\Gamma(L-k+1)}{\Gamma(L+1)(L+1)} \\
&= \sum_{k=0}^{\alpha(1+L)-1} \binom{L}{k} \frac{k!(L-k)!}{L!(L+1)} \\
&= \sum_{k=0}^{\alpha(1+L)-1} \binom{L}{k} \binom{L}{k}^{-1} \frac{1}{L+1} \\
&= \sum_{k=0}^{\alpha(1+L)-1} \frac{1}{L+1} \\
&= \frac{1}{L+1} \times (\alpha(1+L) - 1 + 1) \\
&= \alpha
\end{aligned}$$

□

Following Lemma 3.3.1 and Lemma 3.3.3, it is sufficient to prove Theorem 3.3.1. Therefore, we have shown that whether using all $n!$ permuted series or $L < n!$ permuted series, our proposed testing procedure controls the probability of committing Type I Error below α. Note that the proof does not rely on properties of the test statistic $T(\cdot)$, thus we could say that, the significance level of our proposed permutation testing can be controlled by using any test statistic.

3.3.3 Identify Number of Factors using Permutation Tests

Using the permutation testing procedure described in the previous section, we test for serial dependence across all p components of z_t, resulting in a sequence of p-values: $\hat{p}_1, \ldots, \hat{p}_p$. By comparing each p-value against the pre-specified significance level α, we decide whether a component of z_t exhibits serial dependence. Based on the number of rejections, we then develop our estimator for the number of factors.

This process involves testing p hypotheses simultaneously, which is known as a multiple testing procedure. If we were to use a single hypothesis testing procedure, the probability of committing at least one Type I Error across p trials would be $p \times \alpha$, which exceeds the desired significance level α. To control the probability of Type I Error in a multiple testing procedure, we use the concept of False Discovery Rate (FDR), introduced by Benjamini and Hochberg (1995). The FDR is defined as the expected proportion of rejected null hypotheses that are incorrectly rejected. The process is carried out as follows:

> Let $p_{(1)} \leq p_{(2)} \leq \ldots \leq p_{(p)}$ be the ordered sequence of p-values $p_1, \ldots, p_p$, and $H(i)$ be the null hypothesis of corresponding $p_{(i)}$. Let q be the largest i for which $p_{(i)} \leq \frac{i}{p}\alpha$. Then reject all $H(i), i = 1, \ldots, q$.

Applying this multiple testing procedure would control the False Discovery Rate at level α. Define $\alpha_{FDR} = \frac{i}{p}\alpha$, and compare the sequence of p-values with α_{FDR} to identify the number of components in z_t with serial components, which gives our estimation of the number of factors. Note that the sequence of p-values from our proposed method is monotonically increasing. Therefore, we can begin the comparison from p_1, and stop when we find the first $p_i > \alpha_{FDR}$ to conclude that $\hat{r} = i - 1$. Hence, our proposed estimator for number of factors is:

$$\hat{r}_{PT} = \underset{1 \leq i \leq R}{\arg\min} \{p_i : p_i > \alpha_{FDR}\} - 1, \tag{3.30}$$

where $r < R < p$ is a constant. We usually choose $R = p/2$ in practice.

The property of monotonic increase for the sequence of p-values originates from the way it was constructed. When constructing z_t from (3.11), the columns of $\mathbf{\Gamma}$ were arranged in descending order of corresponding eigenvalues $\hat{\lambda}_i$ of $\widehat{\mathbf{M}}$. A larger $\hat{\lambda}_i$ indicates higher likelihood that the i-th component in z_t is a factor, which also suggests a higher probability of serial dependence for the i-th component, leading to smaller p_i. Therefore, following the monotonically decreasing order of $\hat{\lambda}_i$, p_i should be monotonically increasing.

However, there are two exceptions within the sequence of p_i's that might break the natural ordering, which we refer to as **Spurious Correlation** and **Outliers**. Spurious correlation occurs when a white noise series shows a significant result in the test for serial dependence, making the p-value from the test lower than expected. Outliers refer to those series that should have serial dependence but do not yield a significant result from the test, making the p-value higher than expected. Both cases break the monotonic increasing order of the sequence of p_i's from the permutation test and affect the estimation process. For the sequence of p-values $p_1,\ldots,p_p$ from permutation test, We designed the following rules to identify **Spurious Correlation** and **Outliers** from the sequence of p-values:

- **Spurious Correlation**: For $j \in 1,\ldots,p$, call p_j Spurious Correlation if $p_j < \alpha$, while $p_{j-1} > \alpha$ and $p_{j+1} > \alpha$.
- **Outliers**: For $j \in 1,\ldots,p$, call p_j an Outlier if $p_j > \alpha$, while $p_{j+1} < \alpha$ and $p_{j+2} < \alpha$.

For our estimator, outliers in the first r components will significantly influence our estimation, as the monotonic increasing trend of p-values will be interrupted. In contrast, spurious correlation is of less concern because it typically affects components that are supposed to be white noise, which would not appear in the first r components and, therefore, are usually not included in the estimation process. To eliminate the influence of spurious correlation and outliers, we use the following steps to process outliers without affecting our estimator:

1. Identify and remove results due to *Spurious Correlation* to get the sequence of p-values $\hat{p}_1,\ldots,\hat{p}_p$
2. Identify outliers from $\hat{p}_1,\ldots,\hat{p}_p$ using the definition above.

3. Record the amount and position of all outliers, then remove these outliers from the sequence of p-values to get an updated sequence.

4. Apply multiple testing procedure on updated sequence, and use (3.30) to obtain an estimator $\hat{r'}$.

5. If position of the outlier $\leq (\hat{r'} + \#outliers)$, this outlier should be included. Define these outliers as **important outliers**.

6. The estimator after outlier processing is: $\hat{r}$=$\hat{r'}$ + **#important outliers**.

When dealing with outliers, we need to check their position before deciding how to handle them. Outliers are supposed to be significant results, meaning that the corresponding component in the series should have serial dependence. In such cases, simply removing outliers from the sequence of p-values would cause the estimator to underestimate the true number of factors. In practice, some observations at the end of the sequence of p-values might be mistakenly identified as outliers due to randomness. Therefore, we classify outliers that appear at the front of the sequence as **important outliers** and include these outliers when estimating the number of factors.

The identification of spurious correlations and outliers is critical for improving the accuracy of our estimation. By systematically flagging these anomalies, we can reduce the impact of random noise and ensure that only components with genuine serial dependence are retained, thereby improving the overall estimation of the number of factors. Moreover, the criteria for detecting these issues are set very strictly as a precaution for very rare cases. This strictness minimizes the risk of false identification, thus preserving the robustness and reliability of our estimation procedure.

3.4 Simulations

To demonstrate the performance of our proposed estimator, we report results from Monte Carlo simulations. Under model (3.1), we designed 5 different simulation settings, where strength of factors varies. In all settings, we set true number of factors $r = 9$, number of observations $n \in (400, 900, 1600, 2500, 3600)$, and number of variates $p \in (\sqrt{n}, 0.1n, n)$. Each factor $x_{t,j}, j \in 1, \ldots, r$ is generated from an AR(1) process, with autoregressive coefficients randomly drawn from the interval $[-0.95, -0.4] \cup [0.4, 0.95]$, and have independent $N(0,1)$ innovations. Elements of the factor loading matrix $\mathbf{A}$ are randomly drawn from $N(0,1)$, and components of ε_t are independently drawn from $N(0,1)$. The maximal lag takes value within $m \in (1,2,3)$.

For each setting, we repeat the process for 200 iterations. For our proposed estimator, we set the number of permutations $L = 2000$, and we use only the first $R = max(20, p/4)$ columns of z_t to estimate r. The significance level is set at $\alpha = 0.01$. For comparison, we also present results from the one-step and two-step ratio-based estimators proposed in Lam and Yao (2012a). We denote our proposed estimator as $\hat{r}_{PT}$, and the ratio-based estimators as $\hat{r}_{Ratio1}$ and $\hat{r}_{Ratio2}$.

The selection of the parameter R plays an important role in the effectiveness of our factor estimation approach. The number of factors r is typically much smaller than the total number of variates p, thus R should be large enough to capture the relevant structure but not too large to render the dimension reduction ineffective. A common choice for R is $p/2$ which balances capturing the factor structure while avoiding overfitting. However, R should not be too small, as this could lead to missing important factors. Experimenting with different values of R can help ensure that the estimated number of factors r remains stable and reliable.

The choice of the number of permutations L is crucial for the accuracy and validity of the permutation test. Ideally, L should be as large as possible, with $n!$ permutations providing the most thorough empirical distribution for the test statistic. However, $n!$ is often too large to compute practically, especially for larger sample sizes. When L is too small, the empirical distribution generated from the permutations may not adequately represent the true null distribution, leading to invalid results. The appropriate value of L depends on the desired precision of the p-value. If a very low p-value is required, a larger L is necessary to ensure that the approximation $1/L$ is sufficiently precise. Empirically, $L = 2000$ is often sufficient to obtain reliable p-values, such as those at the 0.05 significance level, providing a reasonable trade-off between computational efficiency and precision.

The details for each setting are described as follows:

- **Setting 1 (1 factor strength level):** $\delta_1, \ldots, \delta_9 = 0$.
- **Setting 2 (2 factor strength levels):** $\delta_1, \ldots, \delta_4 = 0.5$, $\delta_5, \ldots, \delta_9 = 0$.
- **Setting 3 (3 factor strength levels):** $\delta_1, \ldots, \delta_3 = 0.7$, $\delta_4, \ldots, \delta_6 = 0.3$, $\delta_7, \ldots, \delta_9 = 0$.
- **Setting 4 (r factor strength levels, $\delta \sim Unif[0, 0.9]$):** Each factor strength $\delta_1, \ldots, \delta_9$ are randomly drawn from $\sim Unif[0, 0.9]$ at the beginning, and the same set of factor strength is used throughout each iteration in Setting 4.
- **Setting 5 (r factor strength levels, $\delta \sim Unif[0, 0.5]$):** Each factor strength $\delta_1, \ldots, \delta_9$ are randomly drawn from $\sim Unif[0, 0.5]$, which means that factor strength is stronger than in Setting 4, and the same set of factor strength is used throughout each iteration in Setting 5.

The first two settings are designed to fit the model of the two ratio-based estimators. Setting 1 has one factor strength level, where $\hat{r}_{Ratio1}$ should perform very well, and Setting 2 has two factor strength levels, where $\hat{r}_{Ratio2}$ should perform well. Setting 3 challenges both ratio-based estimators, and Settings 4 and 5 are more extreme, where all factors have different strengths. The range of factor strengths in Setting 4 is wider than in Setting 5, and Setting 5 has stronger factors by limiting the factor strength to $\delta \sim Unif[0, 0.5]$. Below, we report the simulation results by presenting the estimation accuracy via the relative frequency $\hat{r} = r$, and the consistency of estimation via the mean and standard deviation of the estimators from 200 iterations.

The relative frequency of $\hat{r} = r$ is reported in Figure 3.1-3.5. Table 3.1-3.5 reports the means and standard deviations of estimators under different settings. These tables demonstrate the consistency of estimations within each setting. Overall, our proposed estimator $\hat{r}_{PT}$ shows strong consistency (low σ_{PT}) within each setting, and is robust across different settings. In terms of estimation accuracy, Figure 3.1-3.5 shows that $\hat{r}_{PT}$ performs well when either $p = \sqrt{n}$ or $m = 1$. When $p > \sqrt{n}$ and $m > 1$, if n increases, $\hat{r}_{PT}$ seems to converge towards $r \times m$. Considering that m is a parameter we choose, we could always choose $m = 1$ to obtain good estimates using $\hat{r}_{PT}$.

In all figures (Figure 3.1-3.5), we observe that for $\hat{r}_{PT}$, $m = 1$ does not always provide the best estimation. The optimal value of m depends on the generating process of the factors. Since the factors were generated using an $AR(1)$ model, autocorrelation at lag 2 will also appear. Given that our proposed method relies directly on testing autocorrelation, it is reasonable to expect that $m = 2$ could provide additional information on serial dependence for $\hat{r}_{PT}$. However, as m exceeds the optimal value, additional noise is introduced, which leads to a slight deterioration in the performance of $\hat{r}_{PT}$.

For $\hat{r}_{Ratio1}$, the estimator performs nearly perfectly in Setting 1, which was intentionally designed to achieve this result. As shown in Figures 3.1-3.5, $\hat{r}_{Ratio1}$ demonstrates improvement as p increases. This phenomenon, often referred to as the "Blessing of Dimensions", is discussed in Lam and Yao (2012a). However, in other settings where multiple factor strength levels are present, $\hat{r}_{Ratio1}$ tends to underestimate the true number of factors r. Its inconsistency is evident from the high standard deviation, which significantly undermines the reliability of its estimation.

$\hat{r}_{Ratio2}$ generally exhibits unstable estimations. Setting 2 was designed to highlight the advantages of $\hat{r}_{Ratio2}$, but as shown in Figure 3.2, it only performs well when $m > 1$ and $p > \sqrt{n}$. In other setting and scenarios, $\hat{r}_{Ratio2}$ produces highly volatile estimations. As observed in Tables 3.1–3.5, the mean estimation is sometimes significantly different from the true value of r, and the standard deviation can be quite high. Unlike $\hat{r}_{Ratio1}$, which consistently underestimates r in Setting 2-5, the direction of mis-estimation for $\hat{r}_{Ratio2}$ is not consistent. In Setting 1, $\hat{r}_{Ratio2}$ always overestimates r, which can be explained. Recall that $\hat{r}_{Ratio2}$ takes an additional step based on the residuals of $\hat{r}_{Ratio1}$. Since there is only one factor level in Setting 1, $\hat{r}_{Ratio1}$ would have already identified all estimators, leaving $\hat{r}_2$ dominated by randomness. Given that minimum value for ratio-based estimators is 1, the estimation from second step will always be equal to or greater than 1. Therefore $\hat{r}_{Ratio2}$ will overestimate r when all factors have the same strength δ.

Selecting the optimal estimator between $\hat{r}_{Ratio1}$ and $\hat{r}_{Ratio2}$ would require prior knowledge of the number of factor strength levels, which is unobservable. In contrast, our proposed estimator$\hat{r}_{PT}$ is robust across any number of factor strength levels, and at $m = 1$, $\hat{r}_{PT}$ is able to consistently provide accurate estimations.

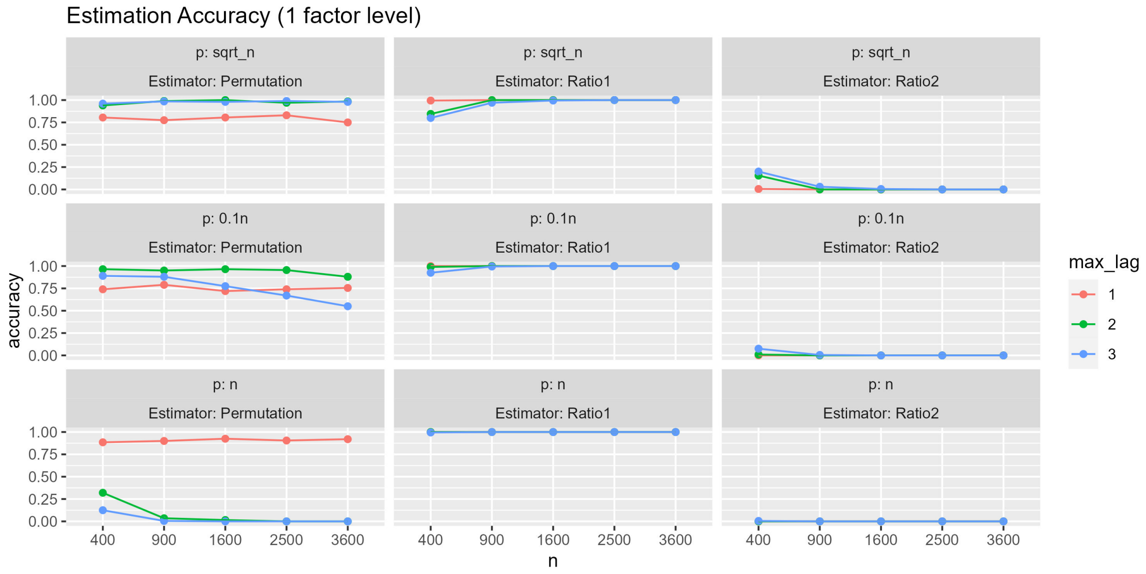

Fig. 3.1 Relative frequency estimates for $\hat{r} = r$ in the simulation with 200 replications for **Setting 1**, where $r = 9$ and $\delta_1, \ldots, \delta_9 = 0$.

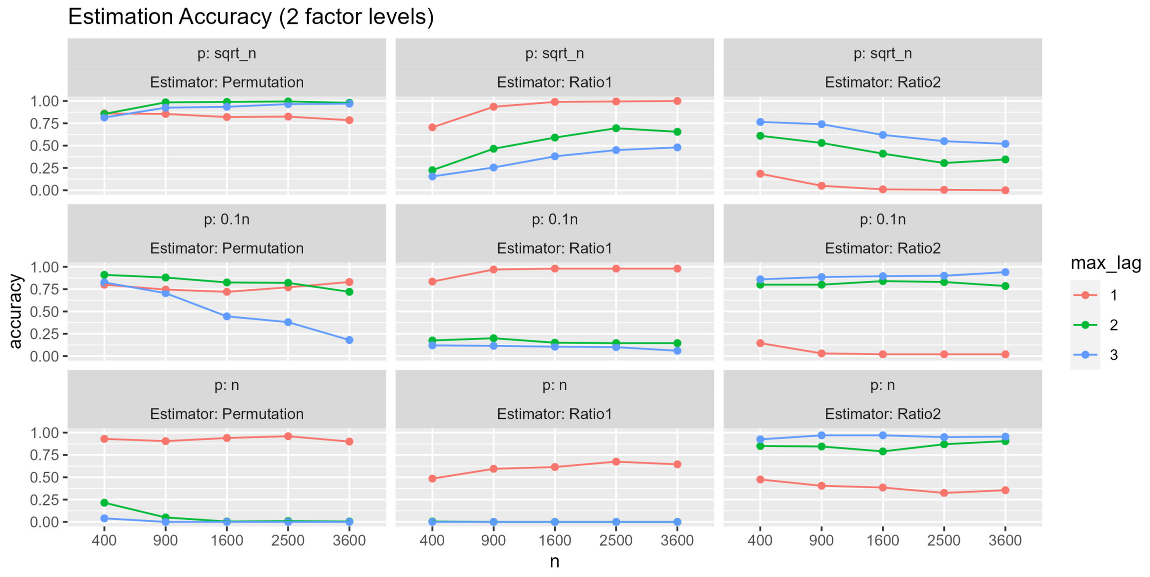

Fig. 3.2 Relative frequency estimates for $\hat{r} = r$ in the simulation with 200 replications for **Setting 2**, where $r = 9$ and factors have 2 strength levels: $\delta_1, \ldots, \delta_4 = 0.5$, $\delta_5, \ldots, \delta_9 = 0$.

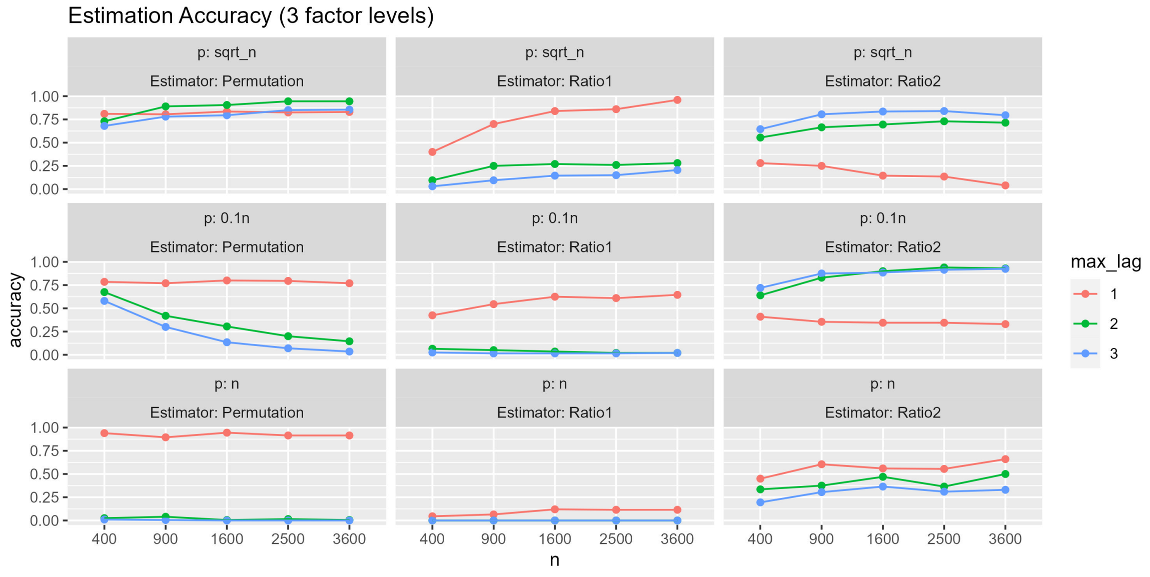

Fig. 3.3 Relative frequency estimates for $\hat{r} = r$ in the simulation with 200 replications for **Setting 3**, where $r = 9$ and $\delta_1, \ldots, \delta_3 = 0.7$, $\delta_4, \ldots, \delta_6 = 0.3$, $\delta_7, \ldots, \delta_9 = 0$.

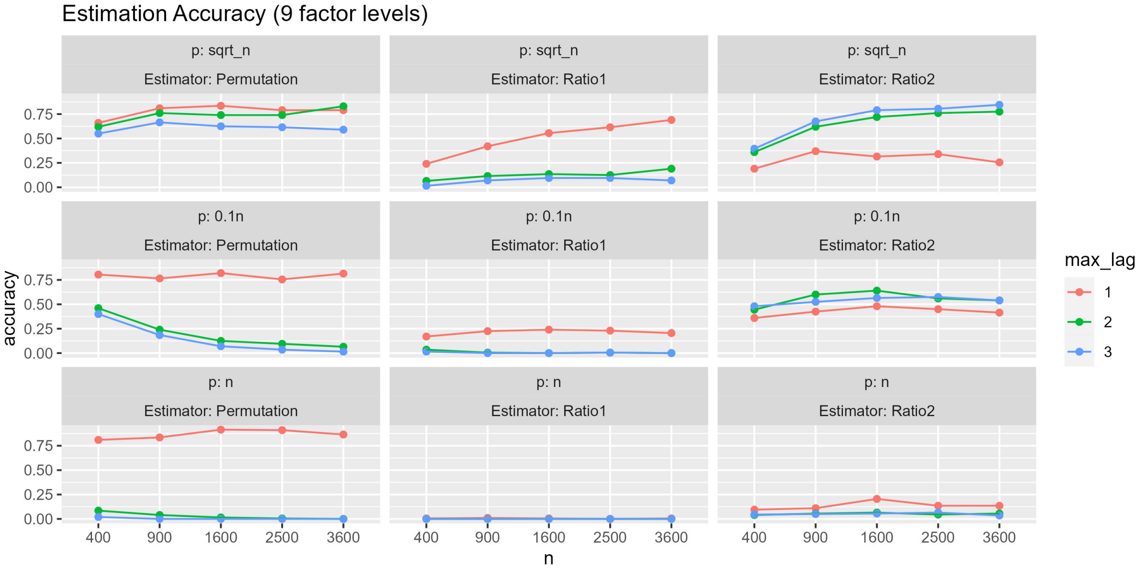

Fig. 3.4 Relative frequency estimates for $\hat{r} = r$ in the simulation with 200 replications for **Setting 4**, where $r = 9$ and $\sim Unif[0,0.9]$.

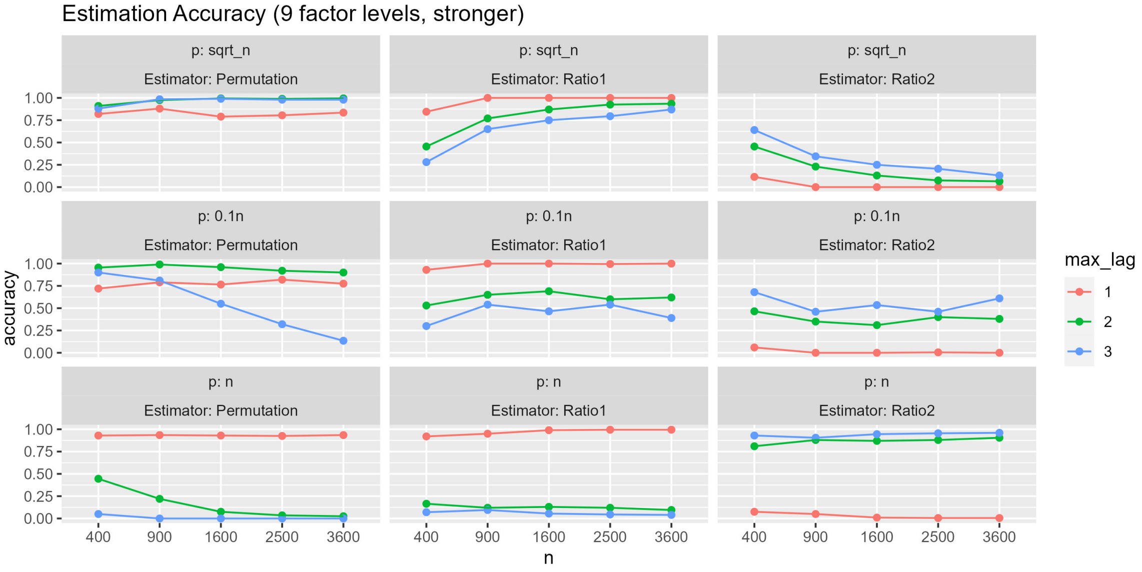

Fig. 3.5 Relative frequency estimates for $\hat{r} = r$ in the simulation with 200 replications for **Setting 5**, where $r = 9$ and $\sim Unif[0,0.5]$.

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{PT}$	m=1	p=$\sqrt{n}$	8.84(0.97)	9.14(0.72)	9.19(0.79)	9.19(0.63)	9.3(0.72)
		p=0.1n	8.91(1.13)	9.17(1)	9.28(0.98)	9.37(0.99)	9.37(0.93)
		p=n	8.77(0.76)	8.95(0.97)	9.07(0.65)	9.07(0.72)	9.13(0.89)
	m=2	p=$\sqrt{n}$	8.92(0.44)	9(0.1)	9(0)	9.03(0.17)	9.02(0.12)
		p=0.1n	8.98(0.28)	9.02(0.26)	9.04(0.18)	9.07(0.45)	9.15(0.51)
		p=n	10.05(1.25)	12.06(1.71)	13.67(1.96)	14.67(1.64)	15.55(1.49)
	m=3	p=$\sqrt{n}$	8.96(0.29)	9.02(0.12)	9.02(0.14)	9.01(0.1)	9.02(0.14)
		p=0.1n	8.98(0.54)	9.13(0.37)	9.3(0.72)	9.69(1.24)	9.98(1.61)
		p=n	12.89(3.53)	20.25(5.55)	24.18(4.44)	26.02(2.89)	26.5(1.71)

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{Ratio1}$	m=1	p=$\sqrt{n}$	8.99(0.07)	9(0)	9(0)	9(0)	9(0)
		p=0.1n	9(0)	9(0)	9(0)	9(0)	9(0)
		p=n	9(0)	9(0)	9(0)	9(0)	9(0)
	m=2	p=$\sqrt{n}$	8.36(1.94)	9(0)	9(0)	9(0)	9(0)
		p=0.1n	8.93(0.75)	9(0)	9(0)	9(0)	9(0)
		p=n	9(0)	9(0)	9(0)	9(0)	9(0)
	m=3	p=$\sqrt{n}$	7.86(2.59)	8.8(1.25)	8.96(0.57)	9(0)	9(0)
		p=0.1n	8.51(1.86)	8.96(0.57)	9(0)	9(0)	9(0)
		p=n	8.96(0.57)	9(0)	9(0)	9(0)	9(0)

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{Ratio2}$	m=1	p=$\sqrt{n}$	18.58(1.12)	21.31(3.3)	25(5.13)	26.73(8.46)	30.06(9.75)
		p=0.1n	23.02(6.58)	31.5(18)	31.32(29.94)	30.52(39.16)	34.44(52.88)
		p=n	11.22(11.87)	10.47(1.12)	10.62(1.94)	10.73(2.81)	10.48(0.98)
	m=2	p=$\sqrt{n}$	17.2(3.57)	21.58(3.2)	23.42(6.42)	26.32(8.12)	28.58(10.66)
		p=0.1n	23.12(6.62)	27.8(18.22)	36.02(32.41)	32.38(40.88)	28.31(47.11)
		p=n	11.19(9.55)	10.49(0.97)	10.57(0.88)	10.62(0.84)	10.65(1.03)
	m=3	p=$\sqrt{n}$	16.72(3.92)	21.87(3.34)	24.41(5.81)	26.4(8.41)	29.18(10.49)
		p=0.1n	21.79(7.58)	26.55(17.56)	32.18(30.19)	32.98(40.57)	29.47(45.91)
		p=n	10.54(0.9)	10.67(0.94)	10.81(0.97)	10.87(1.15)	11.05(1.22)

Table 3.1 Means and standard deviations (in parentheses) for $\hat{r}_{PT}$, $\hat{r}_{Ratio1}$ and $\hat{r}_{Ratio2}$ in the simulation with 200 replications for **Setting 1**, where $r = 9$ and $\delta_1, \ldots, \delta_9 = 0$.

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{PT}$	m=1	p=$\sqrt{n}$	8.88(0.8)	9.11(0.72)	9.22(0.64)	9.24(0.69)	9.37(0.84)
		p=0.1n	9.12(0.83)	9.32(0.74)	9.43(0.85)	9.4(1.04)	9.27(0.77)
		p=n	9.01(0.63)	9.09(0.68)	9.13(0.63)	9.07(0.56)	9.23(0.9)
	m=2	p=$\sqrt{n}$	8.87(0.51)	9.01(0.12)	9.01(0.1)	9.01(0.07)	9.02(0.14)
		p=0.1n	9.01(0.32)	9.15(0.46)	9.23(0.59)	9.32(0.86)	9.52(1.12)
		p=n	12.22(2.64)	14.62(2.37)	15.91(1.95)	16.3(1.7)	16.82(1.67)
	m=3	p=$\sqrt{n}$	8.95(0.55)	9.09(0.39)	9.07(0.27)	9.04(0.22)	9.04(0.32)
		p=0.1n	9.14(0.55)	9.52(1.03)	10.32(1.63)	10.8(2.01)	12.03(2.39)
		p=n	17.72(4.95)	23.96(3.6)	26.32(2.16)	26.9(1.25)	27.04(1.06)

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{Ratio1}$	m=1	p=$\sqrt{n}$	7.85(2.34)	8.68(1.42)	8.95(0.54)	8.96(0.57)	9(0)
		p=0.1n	8.2(2.11)	8.87(0.9)	8.9(0.74)	8.92(0.56)	8.92(0.56)
		p=n	6.81(2.43)	7.42(1.93)	7.46(1.95)	7.7(1.88)	7.58(1.92)
	m=2	p=$\sqrt{n}$	4.86(3.26)	6.5(3.14)	6.86(3.07)	7.62(2.49)	7.47(2.53)
		p=0.1n	5.14(2.99)	5.42(2.8)	5.68(2.56)	5.64(2.74)	6.3(2.95)
		p=n	5.38(3.22)	5.94(3.7)	6.74(3.82)	6.17(3.11)	5.89(2.7)
	m=3	p=$\sqrt{n}$	4.4(3.06)	5.42(3.1)	6.02(3.08)	6.57(2.95)	6.56(2.93)
		p=0.1n	4.38(2.92)	4.43(2.92)	5.04(2.54)	5.31(2.33)	5.43(1.91)
		p=n	4.26(1.9)	4.86(2.05)	5.1(2.47)	5.42(3.2)	5.53(3.09)

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{Ratio2}$	m=1	p=$\sqrt{n}$	16.11(4.56)	20.23(4.92)	24.18(5.93)	27.06(7.67)	26.46(11.44)
		p=0.1n	21.07(7.87)	28.84(18.16)	33.04(29.76)	33.9(42.78)	26.68(43.93)
		p=n	9.56(1.13)	10.11(3.27)	9.88(0.95)	9.91(0.89)	9.91(1.07)
	m=2	p=$\sqrt{n}$	11.3(4.41)	14.85(6.7)	18.05(8.82)	21.06(10.89)	21.66(12.65)
		p=0.1n	11.64(6.26)	13.39(12.07)	12.36(13.89)	12.35(13.94)	16.45(30.8)
		p=n	10.37(13.62)	9.91(2.27)	10.27(2.57)	9.81(2.12)	9.62(1.91)
	m=3	p=$\sqrt{n}$	10.41(3.64)	12.2(5.75)	15.06(8.35)	16.39(10.18)	18.54(12.3)
		p=0.1n	11.04(5.87)	11.26(8.37)	11.86(13.18)	12.34(18.11)	11.6(19.1)
		p=n	8.79(0.81)	9.04(1.21)	9.21(1.63)	9.48(2.36)	9.51(2.36)

Table 3.2 Means and standard deviations (in parentheses) for $\hat{r}_{PT}$, $\hat{r}_{Ratio1}$ and $\hat{r}_{Ratio2}$ in the simulation with 200 replications for **Setting 2**, where $r = 9$ and factors have 2 strength levels: $\delta_1, \ldots, \delta_4 = 0.5$, $\delta_5, \ldots, \delta_9 = 0$.

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{PT}$	m=1	p=$\sqrt{n}$	8.86(0.89)	9.1(0.66)	9.15(0.52)	9.19(0.49)	9.22(0.65)
		p=0.1n	9.05(0.92)	9.33(0.77)	9.23(0.85)	9.38(1.04)	9.41(0.93)
		p=n	9.06(0.61)	9.16(0.7)	9.14(0.74)	9.2(0.8)	9.16(0.87)
	m=2	p=$\sqrt{n}$	8.67(0.7)	9.09(0.34)	9.08(0.32)	9.07(0.31)	9.07(0.33)
		p=0.1n	9.18(0.96)	10.24(1.39)	10.67(1.44)	11.15(1.44)	11.34(1.33)
		p=n	12.16(2.6)	13.77(2.6)	15.29(2.03)	15.87(2.21)	16.5(2.34)
	m=3	p=$\sqrt{n}$	8.78(0.74)	9.19(0.58)	9.23(0.57)	9.19(0.56)	9.2(0.54)
		p=0.1n	9.34(0.98)	10.85(1.83)	12.38(2.28)	14.08(3.06)	15.8(3.88)
		p=n	17.89(3.84)	22.25(2.64)	24.02(2.36)	24.75(2.75)	25.58(2.41)

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{Ratio1}$	m=1	p=$\sqrt{n}$	6.7(2.87)	7.66(2.55)	8.32(1.92)	8.3(2.06)	8.8(1.12)
		p=0.1n	6.22(3.11)	6.9(2.85)	7.5(2.43)	7.44(2.36)	7.54(2.31)
		p=n	4.76(2.3)	5.36(1.9)	5.42(2.19)	5.44(1.97)	5.7(1.93)
	m=2	p=$\sqrt{n}$	4.86(2.93)	5.7(2.93)	5.36(3.15)	5.88(2.94)	5.92(2.87)
		p=0.1n	3.97(2.8)	4.7(2.6)	4.8(2.54)	4.72(2.29)	4.66(2.35)
		p=n	3.73(1.78)	3.61(1.7)	4.07(2.13)	4.28(2.13)	4.77(2.62)
	m=3	p=$\sqrt{n}$	3.48(2.59)	4.7(2.95)	5.22(2.92)	4.81(3.01)	5.38(3.08)
		p=0.1n	3.9(2.46)	4.4(2.5)	4.34(2.3)	4.41(2.3)	4.52(2.27)
		p=n	3.24(1.8)	3.55(1.73)	3.73(1.76)	3.8(1.66)	3.87(1.68)

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{Ratio2}$	m=1	p=$\sqrt{n}$	13.53(5.29)	17.98(6.6)	21.3(7.73)	24.12(10.02)	28(11.14)
		p=0.1n	15.04(8.85)	19.26(16.5)	23.75(26.08)	27.94(39.66)	22.48(39.59)
		p=n	7.92(3.04)	8.22(1.52)	8.3(1.63)	8.31(1.58)	8.56(1.42)
	m=2	p=$\sqrt{n}$	10.13(3.9)	11.94(5.72)	12.89(7.27)	13.52(8.68)	14.27(10.21)
		p=0.1n	8.97(3.95)	9.58(5.28)	9.32(5.65)	9.02(1.45)	8.89(0.75)
		p=n	7.2(1.65)	7.28(1.57)	7.68(1.81)	7.46(1.82)	8(2.18)
	m=3	p=$\sqrt{n}$	8.87(2.42)	9.93(3.81)	11.01(5.59)	11.49(6.77)	13.47(10.11)
		p=0.1n	8.82(3.38)	9.24(4.32)	8.78(0.84)	9.07(2.4)	10.6(17.54)
		p=n	6.88(1.45)	7.11(1.49)	7.22(1.6)	7(1.55)	7.16(1.44)

Table 3.3 Means and standard deviations (in parentheses) for $\hat{r}_{PT}$, $\hat{r}_{Ratio1}$ and $\hat{r}_{Ratio2}$ in the simulation with 200 replications for **Setting 3**, where $r = 9$ and $delta_1, \ldots, \delta_3 = 0.7$, $\delta_4, \ldots, \delta_6 = 0.3$, $\delta_7, \ldots, \delta_9 = 0$.

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{PT}$	m=1	p=$\sqrt{n}$	8.68(1.09)	9.2(0.59)	9.22(0.67)	9.31(0.9)	9.28(0.79)
		p=0.1n	9.12(0.74)	9.29(0.88)	9.3(0.9)	9.42(0.93)	9.3(0.76)
		p=n	9.15(0.89)	9.2(0.73)	9.05(0.61)	9.23(1.03)	9.3(0.95)
	m=2	p=$\sqrt{n}$	8.65(0.8)	9.07(0.6)	9.23(0.65)	9.23(0.56)	9.16(0.43)
		p=0.1n	9.04(0.9)	9.97(1.19)	10.47(1.26)	10.9(1.51)	11.32(1.54)
		p=n	11.7(2.32)	13.33(2.58)	14.32(2.39)	15.01(2.08)	15.79(1.74)
	m=3	p=$\sqrt{n}$	8.48(0.84)	9.18(0.7)	9.43(0.69)	9.44(0.77)	9.51(0.88)
		p=0.1n	9.23(1.11)	10.81(1.86)	12.18(2.39)	13.21(2.61)	15.42(3.44)
		p=n	16.3(2.86)	20.66(1.94)	22.8(2.02)	23.58(2.2)	24.11(1.96)

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{Ratio1}$	m=1	p=$\sqrt{n}$	5.04(3.25)	5.96(3.34)	6.62(3.23)	7.08(3.05)	7.52(2.78)
		p=0.1n	4.12(3.09)	4.86(3.19)	5.04(3.16)	4.77(3.12)	4.61(3.17)
		p=n	2.44(1.75)	3.25(2.19)	2.79(2.12)	2.95(2.1)	2.74(2.07)
	m=2	p=$\sqrt{n}$	3.88(2.76)	4.3(2.93)	4.62(2.97)	4.5(3)	4.89(3.12)
		p=0.1n	3.22(2.53)	3.15(2.29)	2.92(2)	2.98(1.95)	2.74(1.87)
		p=n	2.27(1.35)	2.3(1.41)	2.2(1.46)	2.11(1.18)	2.02(1.2)
	m=3	p=$\sqrt{n}$	3.18(2.28)	3.74(2.72)	3.99(2.92)	4.24(2.92)	4.04(2.79)
		p=0.1n	2.78(2.18)	2.73(1.92)	2.72(1.91)	2.9(1.77)	2.71(1.9)
		p=n	2.23(1.28)	2.18(1.18)	2.15(1.28)	2.09(1.21)	2.08(1.3)

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{Ratio2}$	m=1	p=$\sqrt{n}$	11.15(5.67)	13.79(6.94)	16.77(8.87)	19.66(10.77)	22.22(12.72)
		p=0.1n	9.95(6.84)	13.08(13.32)	13.12(16.21)	13.83(23.96)	11.61(21.33)
		p=n	5.76(2.08)	6.26(1.95)	6.26(2.17)	6.29(1.96)	6.17(2.01)
	m=2	p=$\sqrt{n}$	9.26(4.02)	9.93(4.5)	10.37(5.03)	11.27(7.14)	12.96(9.48)
		p=0.1n	8.25(3.93)	8.09(1.5)	8.16(1.47)	8.64(9.05)	7.93(1.55)
		p=n	5.73(1.84)	5.94(1.79)	6.06(1.78)	5.82(1.75)	5.71(1.69)
	m=3	p=$\sqrt{n}$	7.86(2.92)	9.35(3.71)	10.05(4.56)	10.47(5.78)	10.38(6.21)
		p=0.1n	7.82(3.01)	7.78(1.81)	7.86(1.72)	8.07(1.5)	7.92(1.62)
		p=n	5.74(1.78)	5.92(1.74)	5.8(1.79)	5.91(1.77)	5.77(1.87)

Table 3.4 Means and standard deviations (in parentheses) for $\hat{r}_{PT}$, $\hat{r}_{Ratio1}$ and $\hat{r}_{Ratio2}$ in the simulation with 200 replications for **Setting 4**, where $r = 9$ and $\sim Unif[0,0.9]$.

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{PT}$	m=1	p=$\sqrt{n}$	8.8(1.04)	9.07(0.57)	9.23(0.58)	9.28(0.79)	9.28(0.9)
		p=0.1n	9.06(1.05)	9.3(0.96)	9.36(1.1)	9.24(0.61)	9.34(0.85)
		p=n	8.98(0.61)	9.19(0.78)	9.11(0.86)	9.16(0.84)	9.08(0.44)
	m=2	p=$\sqrt{n}$	8.88(0.49)	9.01(0.2)	9.01(0.07)	9.01(0.1)	9.02(0.28)
		p=0.1n	9.04(0.51)	9.01(0.1)	9.04(0.2)	9.08(0.27)	9.11(0.32)
		p=n	10.11(1.52)	12.1(3.09)	14.01(2.98)	15.2(2.89)	16(2.51)
	m=3	p=$\sqrt{n}$	8.93(0.46)	9.03(0.23)	9.01(0.1)	9.02(0.14)	9.02(0.14)
		p=0.1n	9.05(0.36)	9.26(0.66)	9.74(1.1)	10.53(1.61)	11.95(2.29)
		p=n	16.31(5)	25.44(3.53)	26.82(1.62)	27.08(0.65)	27(0.57)

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{Ratio1}$	m=1	p=$\sqrt{n}$	8.31(2.09)	9(0)	9(0)	9(0)	9(0)
		p=0.1n	8.71(1.39)	9(0)	9(0)	8.99(0.07)	9(0)
		p=n	8.62(1.59)	8.76(1.23)	8.98(0.22)	8.96(0.57)	8.96(0.57)
	m=2	p=$\sqrt{n}$	6.43(3.23)	7.82(2.58)	8.45(1.84)	8.62(1.61)	8.71(1.37)
		p=0.1n	6.46(3.23)	7.48(2.71)	7.34(2.95)	7.06(3.01)	7.32(2.82)
		p=n	4.68(3.13)	4.66(3)	4.96(2.92)	4.78(2.91)	4.8(2.81)
	m=3	p=$\sqrt{n}$	5.03(3.33)	7.12(3.05)	7.56(2.82)	8.01(2.4)	8.31(2.06)
		p=0.1n	4.93(3.4)	6.84(3.07)	6.61(3.07)	6.69(3.11)	6.18(3.19)
		p=n	3.83(2.7)	4.34(2.95)	4.32(2.67)	4.21(2.72)	3.81(2.61)

Estimator	m	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{Ratio2}$	m=1	p=$\sqrt{n}$	17.45(3.4)	21.8(2.85)	23.76(6.07)	25.95(8.96)	28.36(10.59)
		p=0.1n	23.14(6.82)	29.73(18.62)	31.8(28.83)	35.18(42.87)	31.5(49.34)
		p=n	10.25(0.93)	10.36(0.84)	10.42(0.77)	10.45(0.84)	10.49(0.86)
	m=2	p=$\sqrt{n}$	13.71(4.92)	18.82(6.12)	22.15(7.35)	25.39(9.05)	27.52(11.27)
		p=0.1n	16.94(8.77)	22.88(17.72)	24(26.07)	23.05(33.26)	17.87(31.62)
		p=n	9.28(1.11)	9.16(0.5)	9.2(0.57)	9.3(1.96)	9.13(0.48)
	m=3	p=$\sqrt{n}$	11.66(4.48)	17.26(6.66)	20.24(8.3)	21.75(10.19)	25.04(12.17)
		p=0.1n	13.23(7.41)	20.23(16.64)	18.5(22.78)	20.69(31.36)	16.22(29.8)
		p=n	9.12(0.53)	9.13(0.44)	9.09(0.41)	9.07(0.39)	9.05(0.26)

Table 3.5 Means and standard deviations (in parentheses) for $\hat{r}_{PT}$, $\hat{r}_{Ratio1}$ and $\hat{r}_{Ratio2}$ in the simulation with 200 replications for **Setting 5**, where $r = 9$ and $\sim Unif[0,0.5]$.

Although the estimator $\hat{r}$ may not always provide a precise estimate of the true number of factors r, it can still be useful depending on the direction of the estimation error. In factor models, if $\hat{r} \neq r$, overestimation $\hat{r} > r$ is generally preferred over underestimation $\hat{r} < r$. When $\hat{r} > r$, all factors have been identified, and the additional $(\hat{r} - r)$ factors are effectively noise. While the inclusion of this extra noise may lower the quality of the estimation, it ensures that no information about the latent factors is lost. Conversely, if $\hat{r} < r$, we would be missing information from unidentified factors, which is undesirable. Thus, when choosing between two estimators that perform equally in terms of estimation accuracy, we would prefer the one that consistently overestimates rather than underestimates. Under this principle, if $\hat{r} = r + 1$ is deemed an acceptable estimate, we can re-evaluate the performance of both $\hat{r}_{PT}$ and the ratio-based estimators from a different perspective. Below, we present the results for the relative frequency of $\mathbf{P}(\hat{r} \in (r, r+1))$ and $\hat{r} = r$ for setting 4 at $m = 1$.

Estimator	p	n=400	n=900	n=1600	n=2500	n=3600
$\hat{r}_{PT}$	p=$\sqrt{n}$	0.76(0.66)	0.96(0.81)	0.96(0.84)	0.94(0.79)	0.94(0.79)
	p=0.1n	0.92(0.8)	0.92(0.76)	0.95(0.82)	0.92(0.76)	0.95(0.81)
	p=n	0.87(0.81)	0.92(0.84)	0.94(0.92)	0.95(0.91)	0.94(0.86)
$\hat{r}_{Ratio1}$	p=$\sqrt{n}$	0.24(0.24)	0.42(0.42)	0.56(0.56)	0.62(0.62)	0.69(0.69)
	p=0.1n	0.17(0.17)	0.22(0.22)	0.24(0.24)	0.23(0.23)	0.2(0.2)
	p=n	0(0)	0.01(0.01)	0(0)	0(0)	0(0)
$\hat{r}_{Ratio2}$	p=$\sqrt{n}$	0.19(0.19)	0.38(0.37)	0.34(0.32)	0.4(0.34)	0.35(0.26)
	p=0.1n	0.37(0.36)	0.48(0.42)	0.54(0.48)	0.52(0.45)	0.48(0.42)
	p=n	0.1(0.1)	0.12(0.11)	0.21(0.2)	0.14(0.14)	0.14(0.14)

Table 3.6 Relative frequency estimates for $\mathbf{P}(\hat{r} \in (r, r+1))$ and $\mathbf{P}(\hat{r} = r)$ (in parentheses) for $\hat{r}_{PT}$, $\hat{r}_{Ratio1}$ and $\hat{r}_{Ratio2}$ in the simulation with 200 replications for **Setting 4**, where $r = 9$ and $\delta = (0.8, 0.7, 0.3, 0.4, 0.9, 0.5, 0.6, 0.2, 0)$.

From Table 3.6, we observe that by relaxing our criterion to accept $\hat{r} \in (r, r+1)$, the relative frequency for $\hat{r}_{PT}$ shows significant improvement, reaching a satisfactory level. This adjustment allows for a broader range of acceptable estimates, acknowledging that slight overestimation of the number of factors, $\hat{r} = r + 1$, still retains the essential factors. Overall, this approach highlights the effectiveness of $\hat{r}_{PT}$ under less stringent criteria and suggests that it provides a more reliable estimation compared to other methods under these conditions.

To summarize, our proposed estimator demonstrates strong performance, especially when $p \leq \sqrt{n}$ or $m = 1$, where it consistently provides accurate estimates. It outperforms ratio-based estimators in terms of both consistency and robustness, p articularly in settings with multiple factor strength levels, where the ratio-based estimators struggle to capture the true number of factors. $\hat{r}_{PT}$ exhibits greater stability in the face of varying factor structures, making it more adaptable to different scenarios.

Furthermore, when we relax our standard to consider $\hat{r} = r + 1$ as an acceptable estimate, our proposed estimator shows considerable improvement.This flexibility allows it to better handle the inherent variability in the estimation process, especially in cases where slight overestimation is more beneficial than underestimation, ensuring no factors are missed. Thus, our method not only outperforms existing ratio-based approaches but also proves to be more versatile under varying conditions, especially when dealing with complex factor structures and when slight overestimation is acceptable.

3.5 Real Data Example

In this section, we apply our proposed method to the log-transformed daily returns of 480 stocks from the S&P 500 index, covering the period from 2013-01-01 to 2023-11-30. The log return R_t of a selected stock is defined as:

$$R_t = log(\frac{p_t}{p_{t-1}}), \tag{3.31}$$

where p_t is the closing price of the stock on day t. Log transformation is a common preprocessing technique for time series data, offering benefits such as variance stabilization and linearization of trends. It requires the data to exclude zeros or negative values, which is not a limitation in our case, as the price of a stock $p_t \leq 0$ occurs only in very extreme situations for daily stock returns.

The dataset consists of $n = 2748$ observations and $p = 480$ variables, representing 480 stocks selected from the SP 500 index based on the completeness and reliability of their records. These stocks were chosen to ensure a comprehensive and high-quality dataset for analysis. We present estimations from three different methods: $\hat{r}_{PT}$, $\hat{r}_{Ratio1}$ and $\hat{r}_{Ratio2}$, across different maximal lags $m = (1,2,3,4,5)$. These estimations are evaluated to assess the accuracy and robustness of each method under varying lag structures, providing a comprehensive comparison of the performance of each estimator in capturing the number of factors.

	m=1	m=2	m=3	m=4	m=5
Ratio Test $\hat{r}_{Ratio1}$	1	1	1	1	1
Ratio Test $\hat{r}_{Ratio2}$	2	2	2	2	2
Permutation Test $\hat{r}_{PT}$	6	7	7	9	8

Table 3.7 Estimation on r for Real Data, maximal lag $m = (1,2,3,4,5)$ for both estimators.

Table 3.7 presents the estimation results from all three estimators across different values of m. We observe that both $\hat{r}_{Ratio1}$ and $\hat{r}_{Ratio2}$ remain invariant as m changes, yet both are consistently lower than $\hat{r}_{PT}$ for any m. The stable estimation from $\hat{r}_{Ratio2}$ suggests the presence of weak factors, as the estimator is not sensitive to the change in lag. The relationship $\hat{r}_{PT} > \hat{r}_{Ratio2} > \hat{r}_{Ratio1}$ further emphasizes that the strength of factors varies significantly, with $\hat{r}_{PT}$ capturing more factors than the ratio-based estimators. These findings strongly suggest that the ratio-based estimators tend to underestimate the true number of factors in this setting, as they fail to account for weaker but still significant factors that $\hat{r}_{PT}$ identifies more effectively.

	p-value	Smaller than α_{FDR}?
1	0.00000	TRUE
2	0.00050	TRUE
3	0.00000	TRUE
4	0.00000	TRUE
5	0.00000	TRUE
6	0.00030	TRUE
7	0.52024	FALSE
8	0.00050	TRUE
9	0.46227	FALSE (outlier)
10	0.00000	TRUE
11	0.00000	TRUE
12	0.01799	FALSE

Table 3.8 First 10 p-values from permutation testing procedure on Real Data at $m = 1$, $\alpha_{FDR} = 0.000625$.

To better understand how our estimator $\hat{r}_{PT}$ is derived, we can examine the intermediate steps involved. As an example, we listed out the first 12 p-values from the permutation testing at $m = 1$ in table 3.8. Here we identify p_9 as an outlier because both $p_{10} < \alpha_{FDR}$ and $p_{11} < \alpha_{FDR}$. However, p_9 is not considered a significant outlier. Given that $\hat{r}_{PT} = 6$ and $6+1 < 9$, it indicates that p_9 does not fall within the range of important outliers. p_7 is not considered as an outlier because p_8 and p_9 are not significant at the same time. Consequently, the estimator ignores this outlier and concludes that $\hat{r}_{PT} = 6$.

In contrast, for the ratio-based estimators, with $\hat{r}_{Ratio1} = 1$ and $\hat{r}_{Ratio2} = 2$ across all lags, we observe that each step of these estimators identifies only one factor. This result suggests the existence of multiple levels of factor strength. To further illustrate this, we present the first 10 eigenvalues $\hat{\lambda}_i$s and their ratios $\hat{\lambda_{i+1}}/\hat{\lambda}_i$ from each step of the estimation process:

index	$\hat{\lambda}_i$	$\hat{\lambda_{i+1}}/\hat{\lambda}_i$
1	0.0000474	0.1238725
2	0.0000059	0.1634941
3	0.0000010	0.9265492
4	0.0000009	0.6227170
5	0.0000006	0.7877621
6	0.0000004	0.9287467
7	0.0000004	0.5896377
8	0.0000002	0.9848705
9	0.0000002	0.9277768
10	0.0000002	0.9313042

Table 3.9 First 10 eigenvalues $\hat{\lambda}_i$of $\widehat{\mathbf{M}}$ and their ratios at $m = 1$ for $\hat{r}_{Ratio1}$ on Real Data.

index	$\hat{\lambda}_i$	$\hat{\lambda_{i+1}}/\hat{\lambda}_i$
1	0.0000057	0.1618048
2	0.0000009	0.9190688
3	0.0000008	0.6506619
4	0.0000006	0.7887723
5	0.0000004	0.9271086
6	0.0000004	0.5924685
7	0.0000002	0.9651012
8	0.0000002	0.9383784
9	0.0000002	0.9393057
10	0.0000002	0.8524536

Table 3.10 First 10 eigenvalues λ_i of $\widehat{\mathbf{M}}$ and their ratios at $m = 1$ for the second step of $\hat{r}_{Ratio2}$ on Real Data.

In both Table 3.9 and Table 3.10, the ratio λ_{i+1}/λ_i reaches its minimum at $i = 1$, suggesting that each step identifies only one significant factor. If there were only one factor, or if all factors had the same strength δ, $\hat{r}_{Ratio2}$ would be dominated by randomness. However, since $\hat{r}_{Ratio2}$ provides consistent results across $m = (1,2,3,4,5)$, we can conclude that in this real dataset, there are at least two distinct factor strength $\delta_i \neq \delta_j$, and each δ has at least one correaponding factor, indicating $r \geq 2$.

When comparing the ratio-based estimators with our proposed estimator in Table 3.7, we observe that $\hat{r}_{PT}$ consistently produces significantly higher estimates. While the ratio-based estimators suggest the presence of two factors, our method identifies four additional factors. To validate whether these extra components are indeed valid factors, we recover them using equation (3.11) and present their autocorrelation and partial autocorrelation functions to demonstrate the presence of serial dependence.

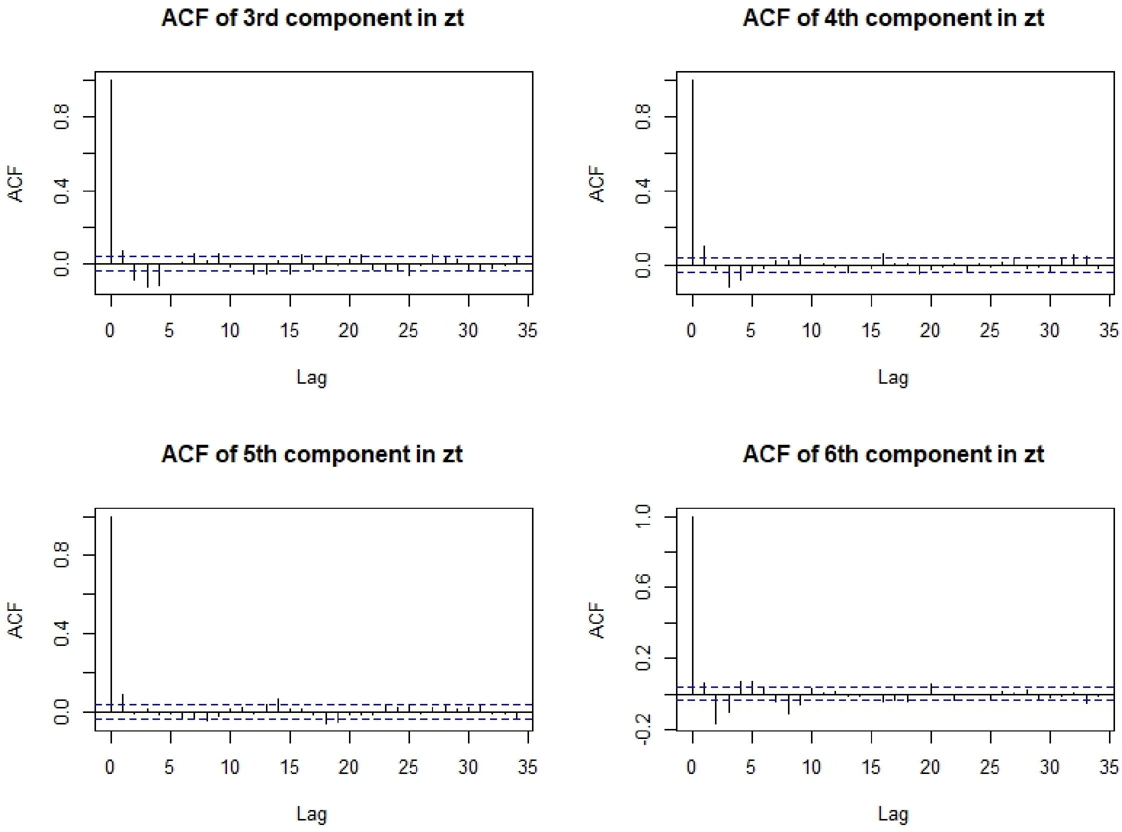

Fig. 3.6 Autocorrelation of 3_{rd} to 6_{th} component of z_t at $m = 1$ on real data.

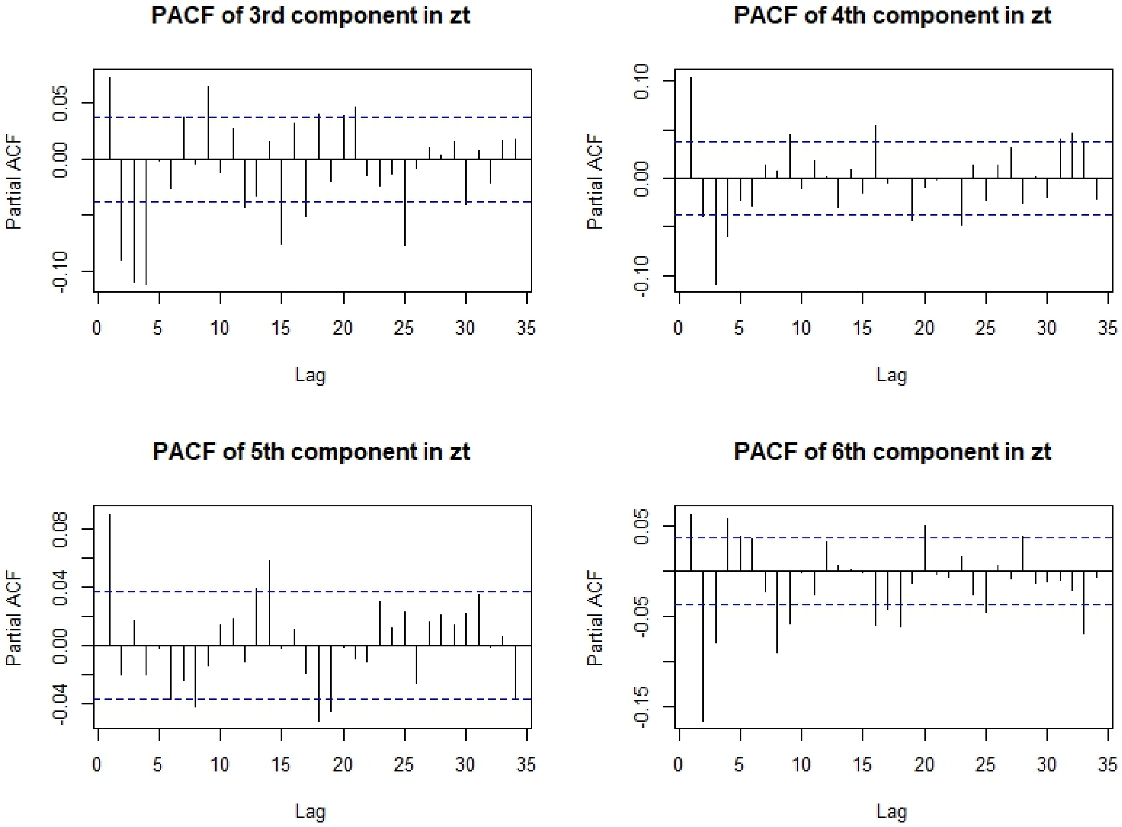

Fig. 3.7 Partial Autocorrelation of 3_{rd} to 6_{th} component of z_t at $m = 1$ with real data.

From Figures 3.6 and 3.7, we observe that both the autocorrelation and partial autocorrelation of the 3_{rd} to 6_{th} components in z_t exhibit clear serial dependence. Based on this observation, these components should also be considered as factors. Therefore, we conclude that $\hat{r}_{PT}$ does not overestimate the true number of factors, which suggests that the ratio-based estimators are underestimating true r.

In conclusion, the application of our proposed estimator $\hat{r}_{PT}$ to real data demonstrates its robustness and accuracy in identifying the true number of factors, particularly when compared to ratio-based estimators. While $\hat{r}_{PT}$ captures a broader range of factors and shows consistency across various lags, the ratio-based estimators tend to underestimate the true number of factors, especially when factor strengths vary. The validation of the additional factors identified by $\hat{r}_{PT}$ through autocorrelation and partial autocorrelation further confirms the reliability of our method in real-world settings.

3.6 Conclusion and Further Discussion

To summarize, under the assumption of exchangeability, we have introduced a non-parametric testing method for determining the number of factors in high-dimensional time series. The proposed procedure is applicable up to the m-th order autocorrelation when $p \leq \sqrt{n}$, and it demonstrates strong performance at $m = 1$ for any value of p. A notable advantage of our method is its robustness across varying levels of factor strength, coupled with its consistency in estimation. Additionally, we have demonstrated that the proposed estimator effectively controls Type I error at the desired significance level, ensuring reliable inference in high-dimensional settings.

There are several open questions concerning the proposed method. First, the power of the test has not been fully addressed. The power of a hypothesis test is defined as $\mathbb{P}(Reject\, H_0 | H_A\, true)$, and ideally, we would aim for a power of 1. While we have demonstrated that the significance level of our proposed test is independent of the choice of test statistic, the power of the test is directly influenced by it. Another key observation pertains to the behavior of our proposed estimator, $\hat{r}_{PT}$, when $m > 1$ and $p > \sqrt{n}$. From Tables 3.1-3.5, we note that when $p > \sqrt{n}$, $\hat{r}_{PT}$ tends to overestimate the number of factors, with the degree of overestimation varying across different settings.

The numerical values suggest that as n and p increase, $\hat{r}_{PT}$ appears to approach $m \times r$. One possible explanation for this behavior is that when p increases faster than $\sqrt{n}$, the sample autocovariance matrix $\widehat{\mathbf{\Sigma}}_y(k)$ becomes an inconsistent estimator for the true covariance matrix $\mathbf{\Sigma}_y(k)$, leading to inconsistent estimates of the eigenvalues $\hat{\lambda}_i$ in $\widehat{\mathbf{M}}$. Several studies in random matrix theory have shown that the empirical covariance matrix may not be a reliable estimator of the population covariance matrix when p is large. If a more consistent estimator for $\mathbf{\Sigma}_y(k)$ were used, the performance of our proposed estimator would likely improve at higher lags. For example, Bickel and Levina (2008) proposed a method for regularizing the covariance matrix via hard thresholding, which is consistent when $log(p)/n \to 0$. Nevertheless, covariance is typically captured adequately by lower lags, which allows our proposed estimator to provide reliable estimates despite the high dimensionality.

Chapter 4

Electricity Load Forecasting by Factor Models, TS-PCA and Matrix TS Models

4.1 Introduction

Electricity load forecasting is the process of predicting electricity consumption at a specific place over a defined timeframe. Typical forecasting horizons are categorized as long-term, mid-term, short-term, and ultra-short-term, corresponding to annual, monthly, daily, and hourly predictions, respectively (Nti et al. (2020)). Long-term and mid-term forecasts guide infrastructure planning, while short-term forecasts support load management within energy systems. Accurate electricity load predictions help optimize energy generation and allocation to meet demand, minimizing waste and preventing shortages that could destabilize the grid.

In short-term electricity load prediction, various modeling approaches have been studied, including traditional statistical models, time series models, and machine learning techniques. Regression models, one of the most widely used traditional statistical methods, are commonly applied to long-term predictions (Kuster, Rezgui, and Mourshed (2017)) but can also be effective for short-term forecasts. Specifically, generalized linear models (GLMs) are frequently used for their simplicity and interpretability, making them popular in the industry. For instance, the French energy company EDF (Pierrot and Goude (2011)), employs a specific GLM variant called the Generalized Additive Model proposed by Hastie and Tibshirani

(1987), which models electricity load as an additive combination of functions of dependent variables. In this model, variables with nonlinear relationships are modeled using smooth functions. However, the selection of variables to smooth and the choice of smooth functions are not automated, as they depend on prior knowledge or exploratory data analysis.

Time series models, such as Autoregressive Integrated Moving Average (ARIMA), are popular in short-term load forecasting. With the increasing availability of high-resolution electricity load data, aided by the widespread use of smart meters, these models can now provide more precise forecasts. However, as time series models predict future value based on past observations solely, they do not incorporate dependent variables, such as temperature and seasonal factors, into the modeling process, making them relatively less sensitive to changes in dependent variables.

Recently, neural-network-based models, including Long Short-Term Memory (LSTM) and transformer-based models (Wang et al. (2022)) have been introduced to electricity load forecasting. These deep learning models are capable of modeling nonlinear relationships, yet challenges remain, such as the gradient vanishing issue, which, despite being mitigated in LSTM, still poses problems in deep time series modeling.

Beside the classification of methods by model, there are other methods from different perspectives. For instance, probabilistic quantile forecasting by Xu et al. (2020) predicts a range of potential values with a given confidence level instead of a single estimate. Hierarchical modeling integrates regional forecasts into national-level predictions, offering a more structured forecasting approach (Brégère and Huard (2022), Antoniadis, Gaucher, and Goude (2023)) .

Considering that EDF has been using Generalized Additive Model (GAM) as its operating model for the time being, in order to further enhance the model's forecasting performance, a hybrid approach that combines different modeling techniques appears most suitable. Building on GAM as a foundation, Cho et al. (2013) proposed a method that models the residuals of GAM as a curve to capture short-term dependencies that GAM might overlook. Similarly, J. d. Vilmarest et al. (2023) and Obst, De Vilmarest, and Goude (2021) developed a state-space model that incorporates the smoothed variables from GAM, allowing for adaptive load estimation. These approaches use GAM as the first step in a model-stacking process, with additional methods layered on to enhance predictive accuracy. Likewise, we are interested in adding time series analysis towards the forecasting of electricity load.

To examine the necessity of adding time series analysis to the modeling of electricity load data, we conduct some exploratory analysis on the national-level electricity load data. Given the resolution of the dataset, we have 48 observations within one day's time, and we refer to these 48 half-hour intervals as "*time of day*" (tod for short), where $tod = 0$ refers to $00:00 \sim 00:30$, and $tod = 47$ refers to $23:30 \sim 00:00$ within a day. With the given resolution of data, we were able to find clear intra-day pattern, as well as weekly pattern at the same time interval of the day. In Figure 4.1, a clear intra-day pattern is visible across each day within the selected week. Additionally, Figure 4.2 illustrates that different times of the day exhibit varying median values and inter-quartile ranges, highlighting unique load patterns at specific intervals within a single day.

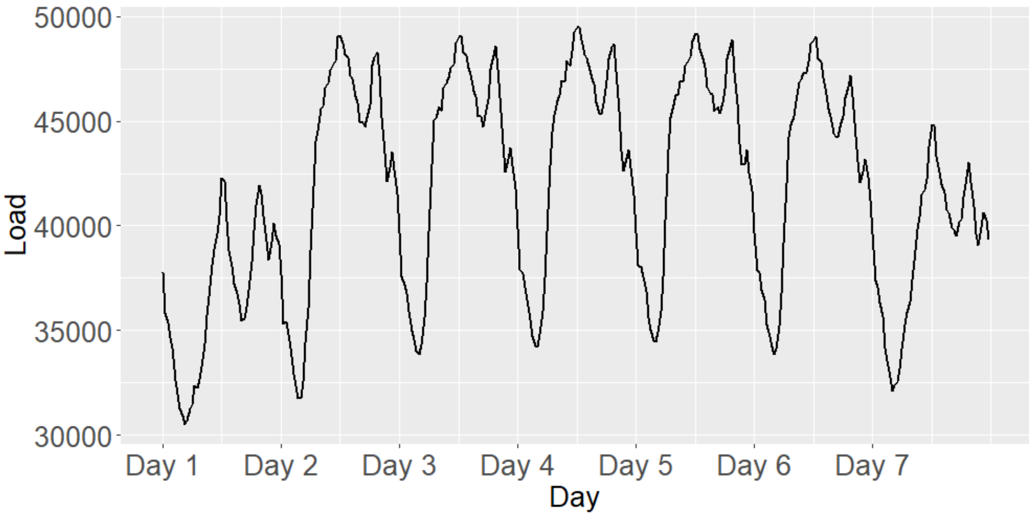

Fig. 4.1 France national electricity load (in MW) from 2023/09/23 to 2023/09/30.

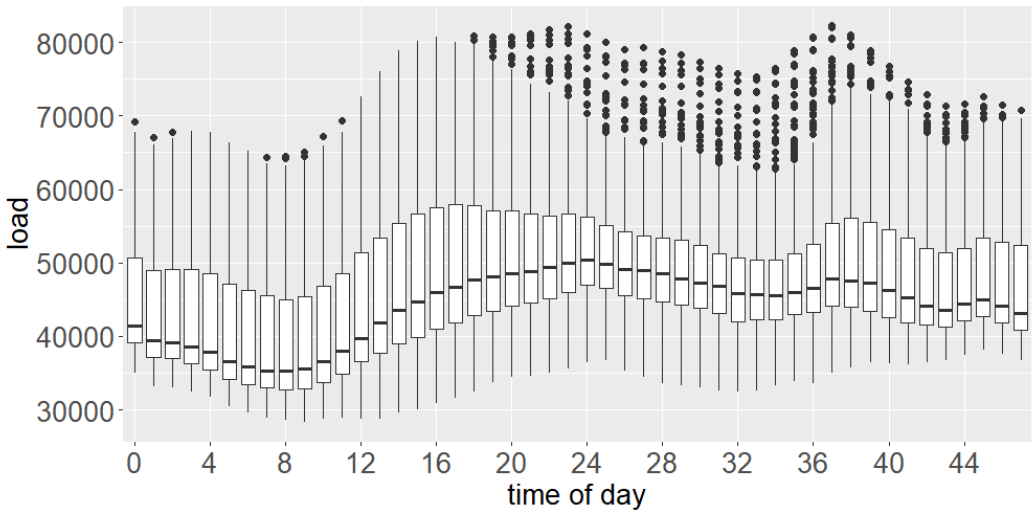

Fig. 4.2 Boxplot of France national electricity load (in MW) at different time of day, from 2023/01/01 to 2023/09/30.

Traditional time series approaches predict the electricity load for day $t+1$ using past observations in chronological order. However, significant intra-day fluctuations make it challenging to efficiently capture patterns that are unique to different times of the day. It would be beneficial to perform data decomposition, such that the series can be break down and modeled by several simpler models, reducing complexity and improving forecasting performance. One straightforward solution is to break the univariate series into 48 sub time series by the half-hour interval, and model each of the 48 half-hour intervals independently. Yet, this approach overlooks interactions between different times of day. Treating the 48 univariate time series as a 48-variate vector time series and applying VAR models could address these interactions, but the model's complexity increases with the high dimensionality, limiting its performance.

Within the framework of model stacking, we propose two time series modeling approaches to address the high dimensionality of the dataset. Our method involves modeling the residuals from GAM and reshaping the residual data to preserve its interactive structure. We propose to reduce the dimension of the reshaped residual data using factor model (Lam and Yao (2012a)) and principal component analysis for time series data (Chang, Guo, and Yao (2018a)). Both methods target the latent structure of the residual data, and find a linear transformation that recovers the latent process. The factor model assumes the presence of a lower-dimensional latent process, achieving dimension reduction by recovering this latent structure. PCA for time series, although also assumes the existence of latent structure, further assumes that there is a latent segmentation of components, and the groups after segmentation are independent of each other, allowing separate modeling without the need of considering cross dependence between groups. In the context of electricity load forecasting, both approaches demonstrate significant improvements over the baseline GAM forecasts.

The rest of the chapter is organized as follow: Section 4.2 describes the French electricity load dataset provided by EDF. In Section 4.3, we present the Generalized Additive Model (GAM) used by EDF and examine the residual information left by the GAM model. Section 4.4 introduces the framework of our proposed method and other methods used for comparison. In Section 4.5, we present a numerical analysis of the forecasts produced by different models. Finally, Section 4.6 concludes with our findings on the modeling process, along with remarks and directions for future research.

4.2 Description of the Dataset

The dataset gathered electricity consumption in France, both at national level and regional level (12 metropolitan regions only). The electricity load data is collected at half-hour resolution for 3865 days, ranging from March 2013 to October 2023. Inside the dataset, we define electricity load as the independent variable, and the rest as the dependent variables. These dependent variables can be classified into 3 categories: **calendar data, electricity load data, and meteorological data**. Calendar data includes basic time variables (date, month and year, etc.), categorical variables (weekday type, time of day, etc.), and dummy variables (for holiday, summer, etc.). Meteorological data consists of variables such as temperature, wind intensity, and nebulosity, sourced from the World Meteorological Organization (WMO). The electricity load data, as the dependent variable, also includes lagged versions of itself (with 1-day and 7-day lags) and is measured in megawatts (MW). A complete list of variables is provided in Appendix B. Both meteorological and electricity load data are available at the national and regional levels, enabling modeling at both scales.

4.3 Prediction Model

4.3.1 The Generalized Additive Models Framework

Generalized Additive Model (GAM) is a type of Generalized Linear Models, which models the independent variable as a sum of smooth non-parametric functions of the dependent variables. Given observations y_t and covariates $x_t^{(1)}, x_t^{(2)}$, where y_t is the electricity load at time t, $x_t^{(1)} = (x_{t,1}^{(1)}, \ldots, x_{t,d_1}^{(1)})^\top$ and $x_t^{(2)} = (x_{t,1}^{(2)}, \ldots, x_{t,d_2}^{(2)})^\top$ are dependent variables at time t. d_1 represents the dependent variables with linear relation to y_t, d_2 represents the dependent variables with nonlinear relation to y_t. GAM model has the form:

$$y_t = \beta_0 + \sum_{i=1}^{d_1} \beta_i x_{t,i}^{(1)} + \sum_{j=1}^{d_2} f_j(x_{t,j}^{(2)}) + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma^2), \tag{4.1}$$

where β_0 is the intercept, and ε_t is a i.i.d. random variable following normal distribution. β_i are the coefficients for the linear terms $x_t^{(1)}$, and $f_j(\cdot)$ are smooth functions for non-linear terms $x_t^{(2)}$. Unlike GLM, which assume that the dependent variables have a linear relationship with the independent, GAM is suitable for modeling nonlinear relationships. The smooth function transforms nonlinear relationships between $x_t^{(2)}$ and y_t, allowing model (4.1) to become linear. Each $x_{t,j}^{(2)}$ have its own smooth function, which is a flexible, non-parametric way to model the nonlinear relationship. One common type of smooth functions used in GAMs is called **splines**, which is a piece-wise polynomial function defined by a set of basis functions and knots. The basis functions are combined linearly with coefficients, which are estimated during the model fitting process. For model (4.1), We selected thin plate splines, which is particularly powerful for multi-dimensional smooths.

By choosing smooth functions to be splines, we could estimate GAM as a GLM via regression with penalty, in which we aim to minimize the following objective function:

$$\mathscr{L}(\beta,f)=\sum_{t=1}^{n}(y_t-\beta_0-\sum_{i=1}^{d_1}\beta_i(x_{t,i}^{(1)})-\sum_{j=1}^{d_2}f_j(x_{t,j}^{(2)}))^2+\sum_{j=1}^{d_2}\lambda_j f_j^\top S_j f_j, \tag{4.2}$$

where $\lambda_1,\ldots,\lambda_{d_2}$ is called smoothing parameter, and S_j is the penalty matrix for the j-th smoothing term depending on the spline. λ_j controls the balance between model fitness and smoothness, where $\lambda_j = 0$ leads to an unpenalized fit, and $\lambda_j \to \infty$ leads to a straight line estimate for $f_j(\cdot)$. This problem can be solved via Restricted Maximum Likelihood (REML). This method involves both the linear and smooth components $x_t^{(1)}$, $x_t^{(2)}$ because REML estimates the variance components (i.e., the smoothing parameters) while adjusting for the uncertainty in the fixed effects (which include both linear terms and smooth terms). We use the R-package `mgcv`, which has this method implemented for GAM modeling. More details regarding REML and other sommthing functions can be found in Wood (2017).

4.3.2 Model Specifications for Predicting Electricity Load

The objective is to predict the electricity load for the next day. Based on observations from Figure 4.1 and 4.2, we see that the electricity load $y_{t,i}$ at $tod = i, i \in [0,\ldots,47]$ on day t is influenced not only by $y_{t,i-1}$, but also by the load at the same tod one day prior, $y_{t-1,i}$. Following this insight, we could perform data decomposition by reshaping the univariate series y_t (dimension 185520×1) into a 48-variate vector time series with $n = 3865$, where each variate represents daily observations at the same time of day over the entire period. This decomposition enables individual modeling for each time of day. Accordingly, we split the dependent variables into 48 subgroups and fit a separate GAM for each time of day, using the following model:

$$\begin{aligned} y_{t,i} = &\sum_{j=1}^{7} m_j \mathbb{1}_{day\_type\_week_t=j} + f(temperature_{t,i}) \\ &+ f(Time_{t,i}) + f(nebulosity_{t,i}) + k\mathbb{1}_{day\_type\_jf_t=1} \\ &+ l\mathbb{1}_{period\_holiday_t=1} + f(Load\_d1) + f(Load\_d7) \\ &+ \varepsilon_{t,i} \end{aligned} \tag{4.3}$$

where $\mathbb{1}$ is the indicator function, and the variables are:

- $y_{t,i}$: electricity load on day t at $tod = i$ of the day.
- *day_type_week*: Day of the week (factor with levels at 0-6) on day t.
- *temperature*: Weighted temperature on day t at $tod = i$.
- *Time*: Mark the ordering of all 185520 observations by time, with 2013-03-02 00:00:00 labelled as 1, and 2023-09-30 23:30:00 labelled as 185520.
- *nebulosity*: Describing the cloudiness on day t at $tod = i$, ranges between 19.38 and 99.33.
- *day_type_jf*: Binary variable on day t, 1 on bank holidays, else 0.
- *period_holiday*: Binary variable on day t, 1 on French national holidays (in all regions), else 0.
- *Load_d1*: Electricity load at $tod = i$ from 1 day ago.
- *Load_d7*: Electricity load at $tod = i$ from 7 days ago.

The selection of variables and smoothing function are recommended by the EDF team based on their past studies. The dataset is split into training data and testing data by year. The training data, covering March 2013 to December 2022, with $n_{train} = 172416$ observations, is used to train the selected models. The testing data, spanning January 2023 to September 2023, includes $n_{test} = 13104$ observations. To forecast the electricity load for day $t+1$, predictions are first generated from each of the 48 individual GAMs. These predictions are then combined into a single vector $\widehat{y}_{t+1}^{GAM}$, ordered chronologically by time of day.

4.3.3 Residuals of GAM Estimations

In this section, we evaluate the effectiveness of model (4.3) by analyzing the model residual. The residual series $\varepsilon_t^{GAM} = y_t - \widehat{y}_t^{GAM}$ represents the error of GAM forecast, where y_t is the actual national-level electricity load in the testing data, and $\widehat{y}_t^{GAM}$ is the forecast generated by the GAM model for the testing period. If model (4.3) has fully captured the dynamics of y_t, then ε_t^{GAM} should behave as a random variable following $N(0,\sigma^2)$, not carrying any useful information. To check whether any information remains in the residuals, we use Quantile-Quantile plots to compare the standardized ε_t^{GAM} against the standard normal distribution $N(0,1)$. A Q-Q plot visualizes how closely the residual data aligns with $N(0,1)$ by comparing the quantiles of the residuals with those of the standard normal distribution.

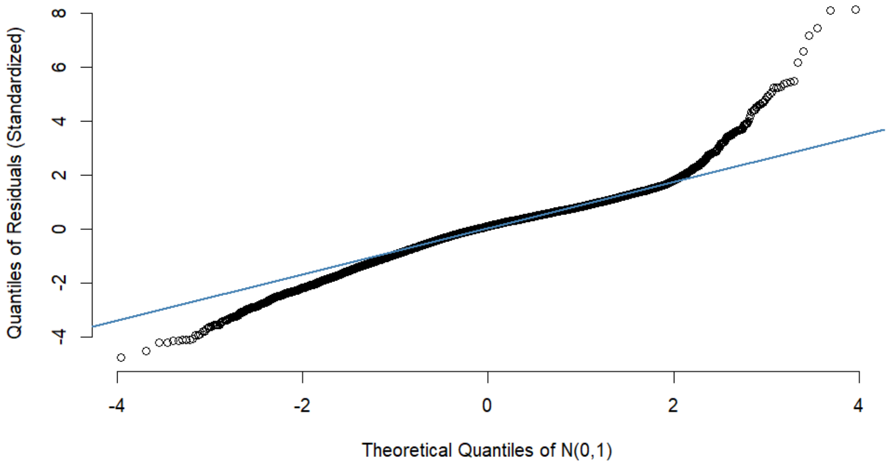

Fig. 4.3 Q-Q plot of ε^{GAM} from test data against $N(0,1)$.

In Q-Q plot, if the two distributions are similar, then the points should fall on the diagonal line $y = x$. However, as shown in Figure 4.3, the points did not align on the $y = x$ line, indicating a significant deviation from $N(0,1)$. Further, the S-shaped pattern suggests that ε_t^{GAM} is more skewed and has heavier tail. This result confirms that ε_t^{GAM} does not follow a normal distribution, implying that the GAM model in 4.3 did not fully capture the relationship between y_t and x_t.

To investigate the potential presence of information within ε_t^{GAM} from a time series perspective, we reshape ε_t^{GAM} into 48 residual sub-time series by time of day (tod), and test for significant autocorrelation within each series using the Ljung-Box test. The Ljung-Box test is commonly used in time series modeling, and it tests whether any autocorrelations up to a specified lag is significantly different from zero. The null hypothesis H_0 states that all autocorrelations up to a given lag k are zero, indicating that the data is independently distributed (no serial correlation). The alternative hypothesis H_1 states that at least one of the autocorrelations at lags 1 to k is non-zero, meaning that there is some autocorrelation present in the time series. We set $k = 7$, and plot the 48 p-values from the Ljung-Box test for all 48 residual sub-series below:

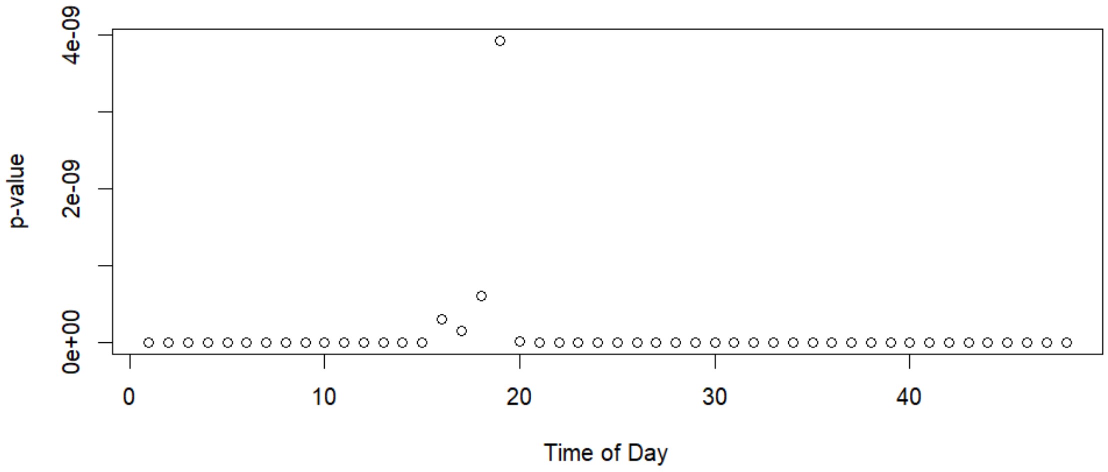

Fig. 4.4 The p-values of 48 residual sub-series ε_t^{GAM} from Ljung-Box test. Maximum lag for Ljung-Box test is set at $k = 7$.

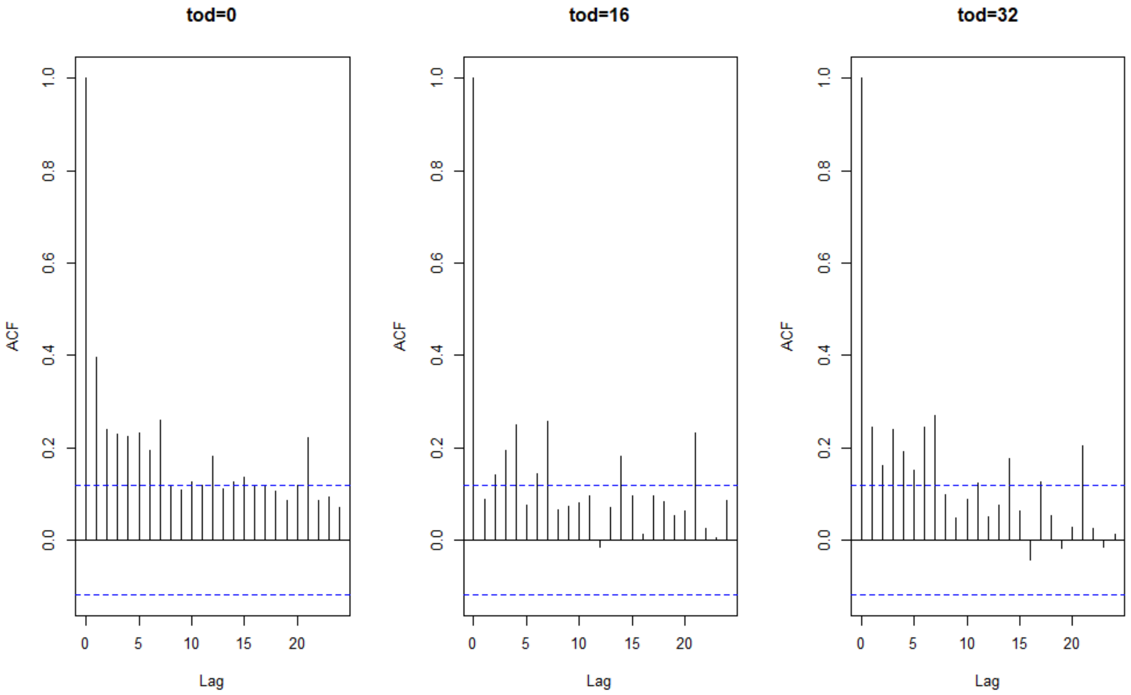

Fig. 4.5 ACF plot of 3 residual sub-series at selected *tod*.

Based on Figure 4.4, we observe that all p-values are significantly lower than the common critical value $p = 0.05$, indicating the existence of autocorrelation in each residual sub-series. To further investigate the pattern of serial correlation, we selected three specific time of day tod and presented their ACF plots in Figure 4.5. The wave-like pattern in all three ACF plots strongly indicates the presence of seasonality within each selected series. This result indicates confirms the existence of time series information within ε_t^{GAM}, which implies that model (4.3) is insufficient to capture time series related characteristics effectively.

Additionally, we are interested in exploring the cross-correlation among all 48 residual sub-series. The cross-correlation at lag k for any pair of time series z_i and z_j is defined as:

$$\rho_{i,j}(k) = \frac{Cov(z_{i,t}, z_{j,t+k})}{\sigma_{z_i}\sigma_{z_j}}, \tag{4.4}$$

where the approximate critical value of $\rho_{i,j}(k)$ at a 95% confidence interval is $\pm\frac{1.96}{\sqrt{n}}$. We calculated pairwise cross-correlation for all 48 residual sub-series, with maximum lag $k = 7$, which allows us to analyze 15 lags in total ($k \in [-7, \ldots, 0, \ldots, 7]$). The cross correlation at any lag k is considered significant if $|\rho_{i,j}(k)| \geq \frac{1.96}{\sqrt{273}}$. Below we present a heat map showing the number of significant lags for each pair of residual sub-series.

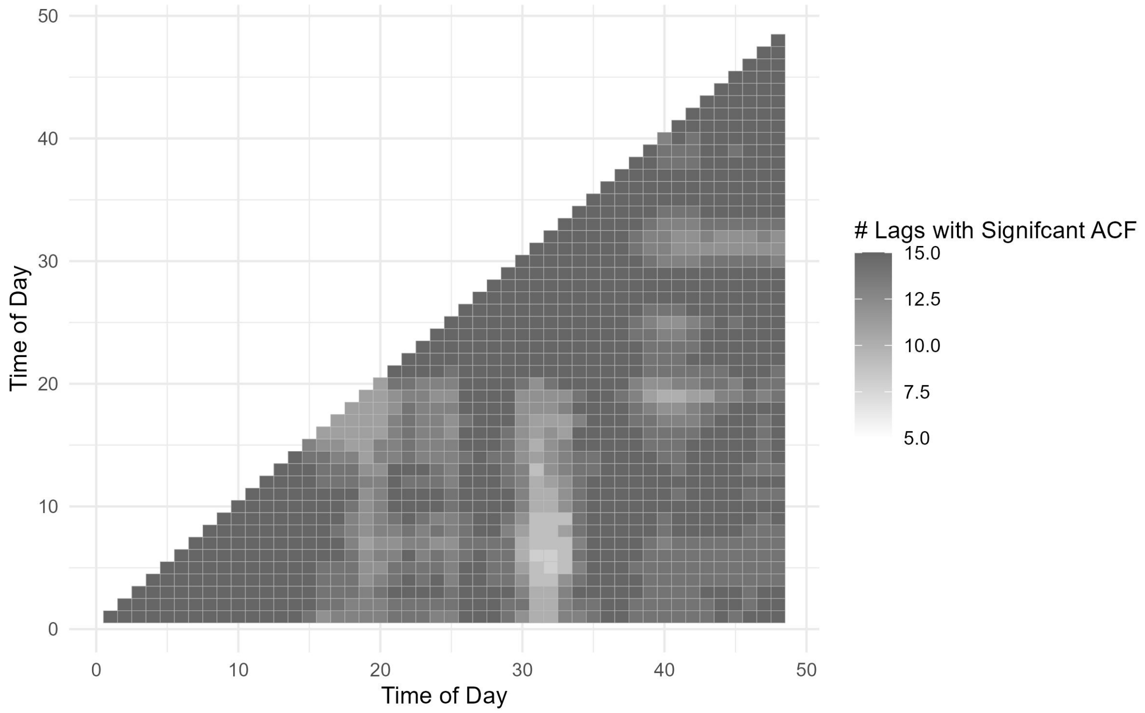

Fig. 4.6 Heat map on number of significant lags for each pair of time series.

Based on Figure 4.6, the lowest number of significant lags across all pairs of residual sub-series is 7, suggesting that every pair of the 48 residual sub-series exhibits significant cross-correlation in at least half of the selected lags. Out of $48 \times 47/2 = 1128$ pairs, 645 pairs have all 15 lags identified as significant, accounting for more than half of the total pairs. This finding indicates a strong cross-correlation among the residual sub-series, restating the need to model the residual series ε_t^{GAM} to capture these dependencies effectively.

4.4 Residual Fitting with Time Series Analysis

In this section, we propose several time-series based methods for modeling the residual series ε_t^{GAM}. Our objective is to model and predict $\widehat{\varepsilon}_t^{GAM}$, and use it to refine the electricity load forecast via $\tilde{y}_t = \widehat{y}_t^{GAM} + \widehat{\varepsilon}_t^{GAM}$. We name this 2-step prediction procedure as residual stacking model. In practice, the EDF team has implemented a similar approach, where they developed another model based on dynamic Kalman Filter, stacking on the outcome from GAM. We refer to this model as the benchmark model, and will introduce more details in section 4.4.5.

When forecasting electricity load for the next day y_{t+1}, time series methods did not use information from x_{t+1} to predict the future residual ε_{t+1}^{GAM}. During the residual modeling process, rather than using the univariate ε_{t}^{GAM} series, we model the 48-variate sub-series instead. A natural choice for multivariate time series modeling would be the vector autoregressive (VAR) model. However, with the high-dimensionality and limited observations in the training dataset, over-parameterization in VAR model becomes a concern. VAR model requires $p+k \times p^2$ parameters, where p is number of variables, and k is the lag. If $p = 48$ and $k = 2$, then the model requires $48+2 \times 48^2 = 4656$ parameters, while the training set contains only $n_{train} = 3592$ observations, making the model insufficiently supported. To overcome the over-parameterization issue, we propose 2 models for multivariate time series modeling on the residual vector time series.

4.4.1 Principal Component Analysis for Vector Time Series

The first method we introduce is Principal Component Analysis for Vector Time Series (TS-PCA), developed by Chang, Guo, and Yao (2018a). This approach applies principal component analysis to high-dimensional time series data, segmenting it into smaller, uncorrelated groups. This segmentation allows each group to be modeled independently, thus alleviating the over-parameterization issue by reducing p in $p+k \times p^2$. The method starts by transforming the original vector time series via performing an eigenanalysis on a positive definite matrix derived from the series' cross-correlation matrix, effectively grouping components based on their correlations.

4.4.1.1 Theoretical Framework for TS-PCA

Given y_t is a $p \times 1$ weakly stationary time series, TS-PCA assumes that y_t has a latent segmentation, which can be represented as

$$y_t = \mathbf{A} x_t, \tag{4.5}$$

where the transformation matrix $\mathbf{A}$ is an unknown $p \times p$ matrix, and the transformed series x_t is a $p \times 1$ unobservable weakly stationary time series, and its p components can be segmented into q groups of sub-series, denoted as:

$$x_t = \begin{pmatrix} x_t^{(1)} \\ \vdots \\ x_t^{(q)} \end{pmatrix}, \tag{4.6}$$

where each group of sub-series $x_t^{(i)}$ is both contemporaneously and serially uncorrelated with the others; that is, for any $i \neq j$, $Cov(x_t^{(i)}, x_s^{(j)}) = 0$ for all $t, s \in 1, \ldots, n$. TS-PCA estimates the transformed series x_t in 2 steps:

1. Let $\Sigma_y(k) = Cov(y_{t+k}, y_t)$ be the autocovariance matrix of y_t at lag k. Define $W_y = \sum_{k=0}^{k_0} \Sigma_y(k) \Sigma_y(k)^\top$, where k_0 is the maximum lag considered for autocovariance matrices. To proceed, perform eigenanalysis on $\widehat{W}_y$ and let $\mathbf{\Gamma}_y$ be the orthogonal matrix with columns being eigenvectors of $\widehat{W}_y$.

2. The matrix $\mathbf{\Gamma}_y$ is a column permutation of the transformation matrix $\mathbf{A}$, and $z_t = \mathbf{\Gamma}_y^\top y_t$ serves as an approximation to the transformed series x_t. The components of z_t can be segmented into q groups of uncorrelated sub-series, allowing for independent modeling of each group.

Note that $\mathbf{\Gamma}_y$ only estimates the true transformation matrix $\mathbf{A}$ up to columns, thus z_t might be different from the true transformed series x_t. This discrepancy does not affect our objective, as we ultimately aim to estimate $y_{t+1} = \mathbf{A} x_{t+1} = \mathbf{\Gamma}_y z_{t+1}$. For further details, please refer to Chang, Guo, and Yao (2018a).

4.4.1.2 Identifying Sub-groups in z_t

The segmentation of components in the estimated transformed series z_t, is based on evaluating the pairwise cross correlations among the 48 components at different time lags. Then, we connect all correlated components and put them in the same group, based on pairwise cross correlation in equation (4.4). Using ccf, we compute cross-correlation between $z_{i,t}$ and $z_{j,t}$ at any given lag k, denote as $\rho_{i,j}(k)$ for simplicity.

To determine whether two components $z_{i,t}$ and $z_{j,t}$ are cross-correlated, we conduct the following hypothesis test to check whether the cross-correlation for all selected lags are zero:

$$H_0 : \rho_{i,j}(k) = 0, for all k = -k_0, \dots, k_0, \tag{4.7}$$

where k_0 is pre-specified as the maximum lag in testing cross correlation. If H_0 is rejected for a pair (i, j), we conclude that $(z_{i,t}, z_{j,t})$ is a connected pair and should be grouped to the same sub-group.

To test the significance of pairwise cross-correlations, Chang, Guo, and Yao (2018a) proposed a measure called **Maximum Cross-Correlation** (MCC), which is defined as:

$$L_n(i, j) = \max_{|k| \leq k_0} |\rho_{i,j}(k)|, \tag{4.8}$$

where $\rho_{i,j}(k)$ is the cross correlation defined in (4.4). To determine the significance of MCC, we could either compare $L_n(i, j)$ with a predetermined threshold value h, where $L_n(i, j) > h$ indicates significant cross correlation, or use the ratio-based method proposed in Chang, Guo, and Yao (2018a), which is detailed as:

1. Compute MCC for all $p_0 = p(p-1)/2$ pairs to get a sequence of L_n. Rank this sequence by ordering $L_n(i, j)$ in descending order $\hat{L}_1 \geq \cdots \geq \hat{L}_{p_0}$.

2. Define the following ratio for the ranked $\hat{L}$ values:

$$\hat{R}_L = arg \max_{1 \leq j \leq p_0} \hat{L}_j / \hat{L}_{j+1}, \tag{4.9}$$

and reject H_0 in (4.7) for all pairs corresponding to $\hat{L}_1 \geq \cdots \geq \hat{L}_{\hat{R}_L}$.

To better explain the idea of this ratio-based approach, we could assume that there are only c connected pairs among all $p_0 = p \times (p-1)$ pairs. Hence, the $c+1$-th pair would be insignificant, and $\hat{L}_{c+1}$ should be very close to 0, while $\hat{L}_c$ is not, resulting in a large value for $\hat{L}_c/\hat{L}_{c+1} = \infty$. As $c+2$-th pair is also insignificant, $\hat{L}_{c+2}$ would be close to zero as well, and the ratio then drops. Thus the ratio is maxed at c, which justifies the above method.

After identifying significantly correlated pairs, groups are created via merging any connected pairs. For instance, if both $(z_{1,t}, z_{2,t})$ and $(z_{2,t}, z_{3,t})$ are significantly correlated, we then merge $z_{1,t}, z_{2,t}$ and $z_{3,t}$ into a single group. Starting with each component as a separate group, we iteratively combines groups until all connected pairs are in the same group. A VAR model can then be applied to each of the q groups to generate predictions $\hat{z}_{t+1} = (\hat{z}_{t+1}^{(1)}, \ldots, \hat{z}_{t+1}^{(q)})$, and the final forecast is obtained as $\hat{y}_{t+1} = \mathbf{\Gamma}_y \hat{z}_{t+1}$.

In ideal situation, this grouping process would mitigates the over-parameterization issue by transforming it into several smaller, uncorrelated sub-series, allowing easier analysis and forecasting. Yet if the underlying latent structure is not well-balanced (e.g. a very large group and several much smaller group), this method contributes less effectively in reducing the number of parameters.

To address this, we propose a new grouping method called **trimmed grouping**, which sets an upper limit s on the size of each of the q groups. Based on the same hypothesis test in (4.7), this approach ensures that no group exceeds the specified size limit significantly, balancing the group sizes to ensure the reduce in complexity of the model.

1. Calculate $L_n(i, j)$ for all pairs of components. Filter those with significant values (e.g. larger than the threshold h).

2. For each component, record all other components with significant correlation in descending order of $L_n(i, j)$. This yields p initial groups.

3. Starting with the first group, if its size exceeds g, select the g most correlated components, record these and remove them from the remaining $p-1$ groups.

4. For the next group, remove any components that were previously selected, then choose the g most correlated components from the remaining, and remove these from the other $p-2$ groups. Repeat until all p components have been assigned.

5. For any groups containing only one component, merge them into existing groups if they have a significant correlation with other components.

This procedure limits the maximum group size to approximately g, by initially keeping only the most significant g pairs in each group and removing less significant components. Although the final group sizes may slightly exceed g due to the last merging step, this excess is minimal and should not impact the effectiveness of the trimmed grouping method in reducing parameter counts within the VAR models, thereby ensuring an efficient segmentation that reduces parameter requirements.

The prediction process then follows the same steps as in the previous methods: a VAR model is applied to each of the q groups to predict $\hat{z}^q_{t+1}$, and $\hat{y}_{t+1} = \mathbf{\Gamma}_y \hat{z}_{t+1}$. We perform the TS-PCA transformation and segmentation on the 48-variate residual series ε_t^{GAM}, and use the predicted value for refining y_{t+1}^{GAM}.

4.4.2 Factor Model with Permutation Testing

Another time-series model for reducing the number of parameters in VAR models is the **factor model**. Factor modeling is particularly effective for high-dimensional multivariate time series, serving as a powerful dimension-reduction tool. This model assumes that the observed multivariate series is driven by a lower-dimensional latent structure and can be formulated as:

$$y_t = \mathbf{A}x_t + \varepsilon_t, \tag{4.10}$$

where x_t is the latent process with dimension $r \times 1$, $\mathbf{A}$ is the factor loading matrix with dimension $p \times r$, and ε_t is noise. It is assumed that the number of latent factors r is much smaller than the dimension of the observed multivariate series p. This assumption enables us to describe the serial dependence in y_t using a much lower-dimensional latent process x_t.

To estimate the factor loading matrix $\mathbf{A}$ and recover the latent process x_t, we apply a similar approach to TS-PCA by defining a non-negative definite matrix based on the summed lagged autocovariance matrices and then perform eigenanalysis.

1. Let $\Sigma_y(k) = Cov(y_{t+k}, y_t)$ be the autocovariance matrix of y_t at lag k. Define $\mathbf{M} = \sum_{k=1}^{k_0} \Sigma_y(k)\Sigma_y(k)^\top$, where k_0 is the maximum lag used for autocovariance matrices. Perform eigenanalysis on $\widehat{\mathbf{M}}$, the sample estimate of $\mathbf{M}$, and let $\mathbf{\Gamma}$ be the orthogonal matrix whose columns are the eigenvectors of $\widehat{\mathbf{M}}$.
2. The matrix $\mathbf{\Gamma}$ is a column permutation of the factor loading matrix $\mathbf{A}$, and $z_t = \mathbf{\Gamma}^\top y_t$ is a $p \times 1$ series that can be seen as a combination of the latent series x_t and random noise, containing r factor components and $p - r$ noise components.
3. The estimate of the factor loading matrix consists of the first r columns in $\mathbf{\Gamma}$, denoted as $\widehat{A} = \mathbf{\Gamma}_{1:r}$.

The key part in factor model is the estimation on number of factors r, which cannot be observed directly. Various methods have been proposed for this purpose, including the ratio test from Lam and Yao (2012a). Their approach estimates $\hat{r}$ based on the ratio of ordered eigenvalues from $\widehat{\mathbf{M}}$:

$$\hat{r}_{Ratio} = \underset{1 \le i \le p}{\arg\min}\, \hat{\lambda}_{i+1}/\hat{\lambda}_i. \tag{4.11}$$

The intuition behind ratio test is, eigenvectors associated with factors will have non-zero eigenvalues, and those associated with white noise will have zero eigenvalues. Since there are r factors, we should have r non-zero eigenvalues and $p - r$ zero eigenvalues. As eigenvalues come with a natural descending order, the ratio λ_{r+1}/λ_r shall be zero. In practice, the estimated "zero" eigenvalues are slightly greater than zero, so the ratio $\lambda_{r+2}/\lambda_{r+1}$ will generally be larger than λ_{r+1}/λ_r. Therefore, by finding the smallest ratio, we can estimate $\hat{r}$.

The ratio test method works effectively when all factors have similar strengths, where factor strength refers to the magnitude of influence each latent factor exerts on the observed data. However, in real-world applications, factors often have varying strengths. To address this, Lam and Yao (2012a) developed a 2-step ratio test. This method first estimates a factor loading matrix $\hat{\mathbf{A}}_1$ using ratio test, then perform the estimation again on $y_t^* = y_t - \hat{\mathbf{A}}_1\hat{\mathbf{A}}_1^\top y_t$ to obtain a new factor loading matrix $\hat{\mathbf{A}}_2$, as well as the total number of factors $\hat{r} = \hat{r}_1 + \hat{r}_2$. However, this approach is only capable of handling 2 different factor strength levels. To handle a broader range of factor strength levels, we developed a new non-parametric method for estimating r, which works well under varying factor strengths.

Given the transformed vector time series $z_t = z_{1,t}, \dots, z_{p,t}$, we are interested in testing the following hypothesis for each component of z_t:

- H_0: $z_{i,t}$ is white noise, with no serial dependence.
- H_0: $z_{i,t}$ is not white noise, and has serial dependence.

To test this hypothesis, we permute the series $z_{i,t}$ multiple times, and apply a test statistic on all permuted series. We calculate the p-value of the test via finding the percentile of test statistic from the original series within the sequence of test statistics from permutations.

$$\hat{p} = \frac{\sum_{j=1}^{n!} \mathbb{1}\{T(z_{w_j,t}) \geq T(Z)\}}{n!}, \quad \hat{p} \in [0,1], \tag{4.12}$$

where w_j is a permutation function on $1,\dots,n$, and $z_{w_j,t} = (z_{w_i(1),t}, \dots, z_{w_i(n),t})$.

The choice of test statistic $T(\cdot)$ does not affect the outcome of the test, as the calculation of p-value does not rely on the theoretical distribution of the selected test statistics. Typically, $T(\cdot)$ will be based on sample autocorrelation of the series, which is defined as:

$$\hat{\rho}_k = \frac{\sum_{t=1}^{n-k}(z_t - \bar{z})(z_{t+k} - \bar{z})}{\sum_{t=1}^{n}(Z_t - \bar{z})}, \quad \bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i, \tag{4.13}$$

where $\hat{\rho}_k$ is the sample correlation at lag k, n is the total number of observation, and $\bar{z}$ is the sample mean. To obtain the most accurate empirical distribution, we need to perform $n!$ permutations on z_t. This would quickly becomes computationally intensive for any $n > 10$. Instead, a good approximation to the empirical distribution can be achieved with only a few thousand permutations. It can be shown that, even with a limited number of permutations, the significance level of permutation testing can be controlled at the desired level, namely:

$$\mathbb{P}(\text{Rejecting } H_0 | H_0 \text{ true, } z_1, \ldots, z_n) \leq \alpha. \tag{4.14}$$

This testing procedure remains effective even with varying factor strengths, enhancing its capability to estimate r accurately in real-world data.The methodologies of the factor model and TS-PCA may appear similar, as both employ eigenanalysis on a summed covariance matrix over multiple lags. However, the underlying assumptions differ between the two models. TS-PCA assumes that y_t can be transformed into several smaller, uncorrelated groups of sub-series, where factor model assumes the existence of a lower-dimensional latent process that drives the observed series. Both methods address the issue of over-parameterization, but the choice of method depends on prior knowledge of the dataset.

4.4.3 Simultaneous Decorrelation of Matrix Time Series (Regional Data)

For regional data, we fit separate GAMs for each of the 12 regions to predict regional electricity load. The residuals from these 12 regions then form 12 separate 48-variate time series, which we reshape into a matrix time series of dimension 12×48, in order to capture the dynamics across the 12 region. An intuitive approach for modeling this matrix time series is the Matrix Autoregressive (MAR) model, an extension of the Vector Autoregressive (VAR) model for matrix-valued series.

The MAR model requires fewer parameters than the VAR model, specifically ($m \times n + k(m^2 + n^2)$ v.s. $(p + k \times p^2) \times 12$). The MAR model imposed a shared structure across the matrix time series, which leads to fewer autoregressive parameters compared to having VAR models for vectorized series. The reduction is significant because the number of parameters in the MAR model depends on 12^2 instead of 48^2, which is a substantial reduction. Although there is no over-parameterization, we may still explore the underlying structure of this residual matrix time series. In Han et al. (2023), they proposed a bilinear transformation method to simplify the modeling and forecasting of high-dimensional matrix time series.

This transformation addresses the challenges of modeling matrix time series with large dimensions by reducing cross-correlations between matrix elements.

Similar to TS-PCA, the core idea is to transform the matrix time series into a set of uncorrelated sub matrix series through a bilinear transformation. This method allows for more efficient modeling by decoupling the original series into smaller subseries that can be modeled independently. The paper by Han et al. (2023) demonstrates that this approach reduces model complexity while preserving the integrity of the data's linear dynamics, making it easier to forecast. Moreover, the transformed matrix time series provides superior forecasting performance, even when the true dynamics deviate from the assumptions.

Given any $p \times q$ matrix time series $Y_t = Y_{t,i,j}$, where $i \in 1,\ldots,p$, $j \in 1,\ldots,q$, we look for a bilinear transformation that can lead to the following segmentation:

$$Y_t = \mathbf{B}X_t\mathbf{A}^\top, \tag{4.15}$$

where $\mathbf{A}$ and $\mathbf{B}$ are unknown constant matrix with dimension $p \times p$ and $q \times q$. The transformed matrix time series X_t is unobservable with the structure:

$$X_t = \begin{bmatrix} X_{t,1,1} & X_{t,1,2} & \cdots & X_{t,1,n_c} \\ X_{t,2,1} & X_{t,2,2} & \cdots & X_{t,2,n_c} \\ & \vdots & & \\ X_{t,n_r,1} & X_{t,n_r,2} & \cdots & X_{t,n_r,n_c} \end{bmatrix}, \tag{4.16}$$

where each X_{t,p_i,q_j} is a sub matrix series with dimension $p_i \times q_j$, n_r and n_c are number of groups in row and columns, and $\sum_i^{n_r} p_i = p$, $\sum_j^{n_c} q_j = q$. All sub matrix series are uncorrelated with any other sub matrix series across all time lags. By modeling each sub matrix series independently via MAR model, the modeling process shall be more efficient.

The estimation for $\mathbf{A}$ and $\mathbf{B}$ follows a similar procedure to factor models and TS-PCA, where we perform eigenanalysis on non-negative definite matrices that are constructed from aggregating covariance matrices over multiple time lags:

1. Standardize Y_t by estimating the row-wise and column-wise covariance matrices:

$$\widehat{\Sigma}_Y^{(1)} = \frac{1}{n \times p} \sum_{t=1}^{n} Y_t^\top Y_t, \quad \widehat{\Sigma}_Y^{(2)} = \frac{1}{n \times q} \sum_{t=1}^{n} Y_t Y_t^\top,$$

   where

$$Y_t = {\Sigma_Y^{(1)}}^{1/2} \mathbf{B}^* X_t^* \mathbf{A}^{*\top} {\Sigma_Y^{(2)}}^{1/2}, \tag{4.17}$$

   such that $\mathbf{A}^*$ and $\mathbf{B}^*$ are orthogonal.

2. Define cross covariance matrices $\widehat{V}^{(1)}$ and $\widehat{V}^{(2)}$ that capture correlation between rows and columns:

$$\begin{aligned}
\widehat{V}_{k,i,j}^{(1)} &= (\widehat{\Sigma}_Y^{(1)^{-1/2}}) \frac{1}{n-k} \sum_{t=1}^{n-k} Y_{t+k}^\top E_{i,j} Y_t \widehat{\Sigma}_Y^{(1)^{-1/2}} \\
\widehat{V}_{k,i,j}^{(2)} &= (\widehat{\Sigma}_Y^{(2)^{-1/2}}) \frac{1}{n-k} \sum_{t=1}^{n-k} Y_{t+k} E_{i,j} Y_t^\top \widehat{\Sigma}_Y^{(2)^{-1/2}},
\end{aligned}$$

   where $E_{i,j}$ is a unit matrix that extracts the element in the (i,j)-th position, and k is the lag.

3. To estimate $\mathbf{A}^*$ and $\mathbf{B}^*$, construct the following non-negative definite matrices:

$$\begin{aligned}
\widehat{W}^{(1)} &= \frac{1}{p^2} \sum_{k=-k_0}^{k_0} \sum_{i=1}^{p} \sum_{j=1}^{p} \widehat{V}_{k,i,j}^{(1)} (\widehat{V}_{k,i,j}^{(1)})^\top, \\
\widehat{W}^{(2)} &= \frac{1}{q^2} \sum_{k=-k_0}^{k_0} \sum_{i=1}^{q} \sum_{j=1}^{q} \widehat{V}_{k,i,j}^{(2)} (\widehat{V}_{k,i,j}^{(2)})^\top.
\end{aligned}$$

4. Perform eigenanalysis on $\widehat{W}^{(1)}$ and $\widehat{W}^{(2)}$, and eigenvectors corresponding to these matrices are the columns of $\mathbf{A}^*$ and $\mathbf{B}^*$.

Once $\mathbf{A}^*$ and $\mathbf{B}^*$ are estimated, we can then estimate the transformed matrix series X_t via

$$\hat{X}_t = \mathbf{B}^{*\top} (\widehat{\Sigma}_Y^{(2)})^{-1/2} Y_t (\widehat{\Sigma}_Y^{(1)})^{-1/2} \mathbf{A}^* \tag{4.18}$$

This transformed matrix series $\hat{X}_t$ can be segmented into uncorrelated sub matrix series, and each sub matrix can then be modeled independently. The segmentation of columns of $\hat{X}_t$ is proceeded by the following steps:

1. Define Z_t as the matrix series obtained by applying the column transformation matrix $\mathbf{A}^*$ to the standardized original matrix series:

$$Z_t = Y_t(\widehat{\Sigma}_Y^{(1)})^{-1/2}\mathbf{A}^*. \tag{4.19}$$

2. Similar to equation (4.8), compute the maximum cross-correlation for each pair of columns in Z_t,:

$$\rho_{l,m}(k) = \max_{1\le i,j\le p,|k|\le k_0} |\frac{Cov(Z_{t+k,i,l}, Z_{t,j,m})}{\sigma_{Z_{t,i,l}}, \sigma_{Z_{t,j,m}}}|. \tag{4.20}$$

   Here, $Z_{t,i,l}$ denotes element in the i-th row and l-th column of matrix series Z_t. This equation finds the maximum cross correlation between column l and m across all time lags within the range $-k_0, \dots, k_0$.

3. Once all significant pairs are identified, iteratively merge connected columns into the same block until all connected pairs are grouped together.

4. For row segmentation, use $(\widehat{\Sigma}_Y^{(2)})^{-1/2}\mathbf{B}^*$ to replace $(\widehat{\Sigma}_Y^{(1)})^{-1/2}\mathbf{A}^*$ in the first step, and repeat the remaining steps.

With X_t segmented into n_r row groups and n_c column groups, we obtain $n_r \times n_c$ blocks of sub-matrix series. For each block, we fit a MAR model to make predictions at $t+1$. By merging predictions from all blocks we could obtain $\hat{X}_{t+1}$, then following equation (4.17), we obtain the final prediction $\hat{Y}_{t+1}$.

4.4.4 Long Short Term Memory

Long Short-Term Memory (LSTM) networks, first introduced in (Hochreiter (1997)), is a specific type of recurrent neural network (RNN), which is designed to handle sequential data and long-range dependencies. Compared to traditional RNNs, LSTMs have longer memory, making them particularly suitable for time series forecasting tasks with extended temporal dependencies.

The LSTM architecture includes three main components within its memory cell structure: the input gate, the forget gate, and the output gate. These gates modulate the flow of information through the network, selectively deciding which information to retain or discard. This gating mechanism enables LSTMs to capture both short-term and long-term dependencies in data. In this setup, let the input be y_t, input gate i, output gate o, forget gate f, memory cell c, and W and U represent the weight matrices for each gate respectively.

- Input Gate: Controls the extent to which new information enters the memory cell.

  $$i_t = f_{ReLU}(W_i y_t + U_i h_{t-1} + b_i), \tag{4.21}$$

  where $f_{ReLU}() = \max(x, 0)$ is the ReLU activation function.

- Forget Gate: Determines what proportion of past information in the memory cell is retained.

  $$f_t = f_{ReLU}(W_f y_t + U_f h_{t-1} + b_f) \tag{4.22}$$

- Output Gate: Controls how much of the memory cell's content influences the current output.

  $$o_t = f_{ReLU}(W_o y_t + U_o h_{t-1} + b_o) \tag{4.23}$$

- Memory Cell: Carries information across time steps. c_t takes the initial value of 0, and will be updated at each step via adding or removing information through the 3 gates above. We first generate a candidate memroy state

  $$\tilde{c}_t = tanh(W_c y_t + U_c h_{t-1} + b_c), \tag{4.24}$$

  where $tanh()$ is the hyperbolic tangent function limiting the range of $\tilde{c}_t$ between $[-1, 1]$. Then we update the memory cell c_t by combining the input gate i_t and the forget gate

f_t with $\tilde{c}_t$:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \tag{4.25}$$

where $\odot$ is the element-wise product.

- Hidden State: Short-term memory of the network, complementing the long-term memory represented by the memory cell state. h_t takes the initial value of 0, and is updated based on the updated memory cell and the output gate, making it a filtered version of the memory cell that emphasizes important information at each time step.

$$h_t = o_t \odot tanh(c_t) \tag{4.26}$$

Although LSTMs are known for its ability to handle complex, nonlinear relationships in time series data, they require a substantial number of parameters, demanding a significant amount of observations to train effectively. The number of parameters for a single LSTM layer is calculated as follows:

$$\text{Total Parameters} = 4 \times \text{hidden units} \times (\text{input dim} + \text{hidden units}) + 4 \times \text{hidden units},$$

where **hidden units** is the number of neurons in the LSTM layer, **input dim** is the number of input features to the LSTM layer, and the number 4 represents the four gates in an LSTM. In our case, we take the 48-variate residual data ε_t^{GAM} as input and predict $\hat{\varepsilon}_{t+1}^{GAM}$. A single 50 neuron layer would have 19800 parameters, which is far more than the size of training data. However, studies including Canatar, Bordelon, and Pehlevan (2021) have shown that, having more parameters than training samples in neural network based models can surprisingly lead to an acceptable generalization, which differs from traditional statistical models. Thus, despite serious over-parameterization, we added LSTM to fit and predict residual series $\hat{\varepsilon}_t^{GAM}$.

4.4.5 Benchmark: Dynamic Kalman Filter

Instead of using GAM prediction as the final forecast, the EDF team developed a two-step method to refine the estimation, where they implemented a method named *dynamic Kalman filter*. The Kalman filter, first introduced in Kalman (1960), is designed to estimate the state of a dynamic system from a sequence of noisy measurements within the framework of state-space models.

A state-space model is a theoretical framework that describes the behavior of dynamic systems over time. It represents the system's internal states and how these states evolve. Both Kalman filter and state-space model are widely used in control systems, navigation, signal processing, and time series analysis. The main purpose of the Kalman filter is to predict the future state of a system while correcting the current state estimation based on new incoming measurements, even if these measurements are noisy. Connecting GAM with Kalman filter combines the flexibility of GAM for capturing nonlinear relationships with the dynamic capabilities of the Kalman filter to handle uncertainties and time-varying behaviors in electricity load forecasting. To link both methods, we follow the setting used in Obst, De Vilmarest, and Goude (2021), J. d. Vilmarest et al. (2023), where we employ the Kalman filter to adaptively estimate and update a vector θ_t such that:

$$\mathbb{E}[y_t|x_t] = \theta_t^\top f(x_t), \tag{4.27}$$

and $f(x)$ is defined as:

$$f(x_t) = \begin{bmatrix} x_t^{(1)} \\ f(x_t^{(2)}) \end{bmatrix},$$

where $x_t^{(1)} = (x_{t,1}^{(1)}, \dots, x_{t,d_1}^{(1)})^\top$ and $f(x_t^{(2)}) = (f_1(x_{t,1}^{(2)}), \dots, f_{d_2}(x_{t,d_2}^{(2)}))^\top$. For simplicity, we fix the nonlinear smoothing functions $f(\cdot)$ obtained from training dataset, and assume that the linear components evolve independently.

To apply Kalman filter on GAM, we transform GAM from a multivariate linear system into the following Gaussian state-space model:

$$\theta_{t+1} = \theta_t + w_t, \quad w_t \sim N(0, Q), \tag{4.28}$$

$$y_t = \theta_t^\top f(x_t) + v_t, \quad v_t \sim N(0, \sigma_v^2), \tag{4.29}$$

Equations 4.28 and 4.29 set up a state-space model for Kalman filtering. In this context, the Kalman filter is used to estimate the latent state θ_t of the system. These equations model the state transitions and observations, where the transition matrix is influenced by the process noise covariance matrix Q, and the measurement noise is captured by σ_v^2. They transform the GAM model into a recursive state-space model to account for noise and uncertainty in the system's behavior. Here $Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}$, where Q_1, Q_2 are for linear state variables $x_t^{(1)}$ and nonlinear state variables $x_t^{(2)}$ respectively. v_t is the measurement noise representing the variance of observation, and σ_v^2 is the space noise variance.

To adaptively estimate θ_t, we can use the recursive formula of Kalman filter, assuming the prior distribution of $\theta_1 \sim N(\hat{\theta}_1, P_1)$, where $P_1 \in \mathbf{R}^{d \times d}$, $d = d_1 + d_2$ is positive definite. Then at each time $t \in 1, \dots, n$, we predict:

$$\mathbb{E}[y_t | x_{1,\dots,t}, y_{1,\dots,t-1}] = \hat{\theta}_t^\top f(x_t), \tag{4.30}$$

where $\hat{\theta}_t$ and P_t are update recursively with:

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \frac{P_t f(x_t)}{f(x_t)^\top P_t f(x_t) + \sigma_v^2} (y_t - \hat{\theta}_t^\top f(x_t)), \tag{4.31}$$

$$P_{t+1} = P_t - \frac{P_t f(x_t) f(x_t)^\top P_t}{f(x_t)^\top P_t f(x_t) + \sigma_v^2} + Q. \tag{4.32}$$

Equations 4.31 and 4.32 describe the recursive updates for the Kalman filter. Equation 4.31 updates the estimate of the state θ_t, while Equation 4.32 updates the covariance matrix P_t, both of which evolve over time. These updates allow the Kalman filter to refine the state estimate by incorporating new observations and reducing uncertainty. In this setting, both Q and σ_v are are unknown. The term *dynamic* in dynamic Kalman filter refers to a specific setting, where $Q \neq 0$, indicating a non-constant state vector, and similarly, σ_v not being constant as well. This allows the system's behavior to vary over time, which may be casued by external disturbances, changing conditions, or unmodeled dynamics. A dynamic Kalman filter estimates Q with an online manner, improving the filter's ability to track the state accurately under different noise conditions and adjust to time-varying system behavior.

To be more specific, dynamic Kalman filter initializes $Q(0) = 0$ and updates its diagonal elements iteratively by adjusting one entry at a time. For each potential adjustment, we choose the change that maximizes the likelihood through a grid search over a predefined set of values. In our setting, we use $\{2^j, -30 \leq j \leq 0\}$ as the candidate of parameter for grid search updates.

The estimation of Q and σ_v are conducted through an iterative greedy procedure implemented in the R-package `viking` (J. d. Vilmarest and Wintenberger (2024)). By estimating these parameters adaptively, the dynamic Kalman filter is able to handle time-varying or uncertain system behavior, and can improve the accuracy and reliability of state estimates over time, which is crucial especially during times such as the COVID-19 pandemic period.

4.5 Performance Analysis

With the numerous methods introduced in the previous section, we now turn to evaluating each method's performance in forecasting electricity load. To compare the quality of forecast from each method, we use the root mean square error (RMSE) as our evaluation metric:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}. \tag{4.33}$$

Among the methods introduced in section 4.4, all except the Dynamic Kalman Filter predict the residual $\hat{\varepsilon}^{GAM}$ from GAM. We compute the final load forecast by

$$\tilde{y}_{t+1} = \hat{y}^{GAM}_{t+1} + \hat{\varepsilon}^{GAM}_{t+1},$$

and calculate the RMSE to compare forecast accuracy.

When predicting GAM residuals at national level, we treat the input as a 48-variate time series. For regional-level predictions, we can either treat each of the 12 regions separately, using approaches such as the Factor Model or TS-PCA, or reshape the data into a matrix time series to apply Simultaneous Decorrelation of Matrix Time Series. Alternatively, we can vectorize the matrix time series, and use methods from national level predictions as well. In the following sections, we present the prediction RMSE for each of these approaches.

For predictions, we use a common time series forecasting technique called **One-Step-Ahead** prediction. This approach predicts the time series value at the next time point using all data available up to the current time. At time t, we use all observations up to t to train the selected model, and the model predicts only one future time point $t+1$. As new data becomes available with each time step, it is added to the training dataset to generate the next one-step-ahead prediction. This method avoids the compounding errors that occur in multi-step ahead time series forecasting.

While using all available data allows one to capture long term trends and provides more observations for model fitting, it also intensifies computational burden. Additionally, if seasonality or other patterns change over time, older data may no longer be relevant and could degrade forecast accuracy. To address this, we also experimented with training models on more recent data only, selected via **sliding window approach**. In this method, we set a fixed window length for the training data, which moves forward as new data arrives, ensuring that the model always trains on the most recent and relevant information.

4.5.1 Prediction on National Level

In this section, we focus on predicting electricity load data at the national level. The national electricity data is reshaped into a 48-variate time series based on the time of day. We use six different methods for making predictions on this 48-variate time series: univariate AR model, vector AR model, TS-PCA, Factor Model, LSTM, and dynamic Kalman filter. The first 5 models predicts the residual series $\hat{\varepsilon}_{t+1}^{GAM}$ based on residuals of GAM forecasts, refining the GAM output by capturing additional time-dependent structures within the residuals. The dynamic Kalman filter, in contrast, bypasses the residual correction approach, aiming to directly estimate $\hat{y}_{t+1}$ by adaptively incorporating new information from new observations. This array of models provides a comprehensive comparison, allowing us to evaluate their relative strengths in enhancing GAM estimations. In the sections that follow, we will analyze the prediction accuracy of each method.

4.5.1.1 Model Specifications

In the univariate AR model, we treat the 48-variate residual series as 48 independent time series, fitting a separate AR model for each and predicting the value at $t+1$ for each series. The lag for each univariate AR model is automatically selected based on the Akaike Information Criterion (AIC). The VAR model, in contrast, models the entire residual series as a single multivariate process, capturing the interactions across all 48 time slots. Like the univariate AR model, the lag of the VAR model is also automatically chosen using AIC.

For TS-PCA model, we present two prediction results under different grouping settings: the default ratio-based grouping approach originally proposed in Chang, Guo, and Yao (2018a), and the proposed trimmed grouping approach in section 4.4.1. We set the threshold for identifying significant cross-correlation at 0.08, which is the first quartile of maximum cross-correlation values across all pairs, and cap the group size at 10.

Likewise, for factor model, we also present two prediction results, using different approach for estimating number of factors. The first approach is the ratio-test based approach described in Lam and Yao (2012a), and the second being permutation testing approach described in section 4.4.2, which is a more flexible alternative when factor strength varies. We set maximum lag $m = 1$ for both methods, and number of permutation to be 2000.

In the construction of the LSTM model, we constructed a 3-layer LSTM, each layer consisting of 50 cells. The training procedure involves running the model for 200 epochs with a batch size of 128. We used the ReLU (rectified linear unit) activation function for the gating mechanisms in LSTM cells. The use of ReLU facilitates faster convergence and allows the model to better capture complex temporal patterns. The LSTM model takes the

48-variate residual series as input, and predicts its value at $t+1$. The timestep we select is 7, which allows us to incorporate one week of data into each training iteration.

For dynamic Kalman filter, we process any nonlinear variables using thin plate regression splines, as inherited from GAM. The iterative estimation of process noise covariance matrix Q starts from a zero matrix, and space noise variance σ_v^2 starts from 1.

4.5.1.2 Prediction Performance Comparison

Table 4.1 displays the RMSE of predictions from each method, providing a clear comparison of forecast quality. All methods achieve lower RMSEs than GAM, indicating that each method enhances prediction accuracy over the test period. Among the 10 settings, we use univariate AR and VAR as the benchmark model for time series models. The dynamic Kalman filter we used is a simplified version of the operating model used by EDF. Overall, the factor model stands out with the lowest RMSE, surpassing both time series benchmark models and the EDF operational model. Both AR models showed moderate improvement relative to GAM. VAR performs well, mostly contributed by its ability to capture interactions among the different variates. Notably, despite the excessive number of parameters in the VAR model relative to the sample size, it still achieves a reasonable performance, which is somewhat surprising given the issue of over-parameterization.

	Data	Method	RMSE
1	All Time	Raw Residual from GAM	1537
2	All Time	Dynamic Kalman Filter	1205
3	All Time	Uni-AR	1310
4	All Time	VAR	1270
5	All Time	TS-PCA (Ratio-Based)	1359
6	All Time	TS-PCA (Trimmed Grouping)	1301
7	All Time	Factor Model (1-Step Ratio)	1340
8	All Time	Factor Model (2-Step Ratio)	1237
9	All Time	Factor Model (Permutation Testing)	1197
10	All Time	LSTM	1351

Table 4.1 RMSE of predictions from each listed methods using all historical data as training data. Predictions are made from January 2023 to September 2023. All models except dynamic Kalman filter are trained and used following the One-Step-Ahead approach.

As shown, the TS-PCA method did not outperform VAR, yet the proposed trimmed grouping approach has better performance than the default ratio-based TS-PCA. For all listed factor models, the only difference in model specification is the approach for estimating number of factors. The RMSE results suggest that the permutation testing approach for identifying the number of factors $\hat{r}$ has the best performance overall. Further details about TS-PCA and factor models will be discussed in the next section.

For the LSTM model, the less-accurate performance is expected due to model complexity, highlighted by its parameter count. As stated earlier, we used a 3-layer LSTM. The first layer, which takes 48 input features and has 50 neurons, requires $4 \times 50 \times (48+50) + 4 \times 50 = 19800$ parameters, and the second and third layer requires $4 \times 50 \times (50+50) + 4 \times 50 = 20200$ parameters. Additionally, the output layer has $50 \times 48 + 48 = 2448$. Summing up parameters from each layer, the LSTM model has a total of 62648 parameters, which far exceeds the number of observations. Despite this over-parameterization, the LSTM's performance demonstrates its ability to handle complex data and learn effectively even with a large number of parameters.

While RMSE gives a measure of accuracy for forecasts from different methods, it does not provide sufficient evidence to tell if one model is statistically better than another. To better compare forecasting quality, we employ the Diebold-Mariano (DM) test proposed by Diebold and Mariano (2002). The DM test is designed to check if the difference in forecasts from two different models is statistically significant. It tests the null hypothesis that, given two forecasts from two distinct methods $\widehat{y}_{i,t}$ and $\widehat{y}_{j,t}$, the population mean of the loss differential series $d_t = g(e_{i,t}) - g(e_{j,t})$ is 0. Here $g(\cdot)$ is the loss function, $g(\widehat{e}_{i,t}) = g(y_t, \widehat{y}_{i,t})$ and $g(\widehat{e}_{j,t}) = g(y_t, \widehat{y}_{j,t})$ are prediction errors. In Diebold and Mariano (2002), they constructed a parametric test, with the test statistic designed as follow:

$$S_1 = \frac{\bar{d}}{\sqrt{\frac{2\pi \widehat{f}_d(0)}{T}}}, \tag{4.34}$$

where $\bar{d}$ is the sample mean loss differential. $\widehat{f}_d(0)$ is an estimation of $f_d(0)$, which is the spectral density of the loss differential at frequency 0. The test statistic S_1 follows a standard normal distribution $N(0,1)$.

The DM test is robust, allowing forecast errors to be non-Gaussian, have nonzero mean, and exhibit both serial and contemporaneous correlation, making it applicable across a wide range of scenarios. We apply the DM test on the proposed methods using the R function `dm.test`, which is a modified DM Test proposed in Harvey, Leybourne, and Newbold (1997). Below we present our findings in Table 4.2:

Method 1	Method 2	Test Statistic S_1	p-value
GAM	Dynamic KF	19.13	0.00
GAM	Factor Model (Permutation Testing)	31.63	0.00
Dynamic KF	Factor Model (Permutation Testing)	0.44	0.66

Table 4.2 p-value of DM test on selected methods, based on national electricity load data. Method 1 generates $\widehat{y}_{i,t}$, and Method 2 generates $\widehat{y}_{i,t}$.

Table 4.2 shows that both the dynamic Kalman filter and the Factor Model with Permutation Testing yield forecasts that are statistically different from those of GAM, and positive test statistic S_1 indicating that the errors from GAM forecasts are larger. This suggests that both the dynamic Kalman filter and the Factor Model with Permutation Testing improve forecast accuracy based on GAM. However, when comparing the dynamic Kalman filter to the factor model with permutation testing, the DM test does not reject the null hypothesis, indicating insufficient evidence to conclude that the forecasts from the factor model with permutation testing are statistically better than those from the dynamic Kalman filter. Thus, while both methods outperform GAM, their relative performance difference is not statistically significant based on the DM test results.

The dataset contains 10 years' electricity consumption, and there might be changes in electricity consumption patterns. If such changes exist, using all historical data could reduce model credibility. To address this, we experimented with training models on more recent data, selected through the sliding window approach discussed earlier. In Table 4.3, we compare the prediction performance of selected methods when trained on the full dataset versus only recent data. This comparison allows us to evaluate whether focusing on more relevant, recent data improves forecast accuracy, as well as inspecting the presence of evolving electrycity consumption patterns.

	Method	Training Data Set	
		All Time	Sliding Window(2Y)
1	Uni-AR	1310	1320
2	VAR	1270	5451
3	TS-PCA (Ratio-Based)	1359	1369
4	TS-PCA (Trimmed Grouping)	1301	1347
5	Factor Model (1-Step Ratio)	1340	1308
6	Factor Model (2-Step Ratio)	1237	1301
7	Factor Model (Permutation Testing)	1197	1286

Table 4.3 RMSE of predictions from each listed methods using sliding window with length = 2 years as training data. Predictions are made from January 2023 to September 2023. All models are trained and used following the One-Step-Ahead approach.

Based on RMSEs in Table 4.3, we discover that using more recent data does not improve, but rather weakens the predictions, possibly due to reduced number of observations. In VAR models, there is a notable increase in RMSE, highlighting the negative impact of over-parameterization when training data is limited. This suggests either there is no change in pattern, or there are multiple changes in both training data sets. The relative performance ranking among all methods remains roughly the same, except that VAR gets worse by a larger margin. Such result may also support the effectiveness of factor models with permutation testing. LSTM is not included in this comparison, as it is well-suited for capturing both long-term and short-term dependencies, and the issue of over-parameterization will only get worse with limited observations.

To further compare the characteristics of each method, we breakdown the prediction from selected methods (dynamic Kalman filter, TS-PCA with trimmed grouping, Factor Model with Permutation Testing, LSTM) by month of year and time of day, and we present the results in Figure 4.7 and 4.8:

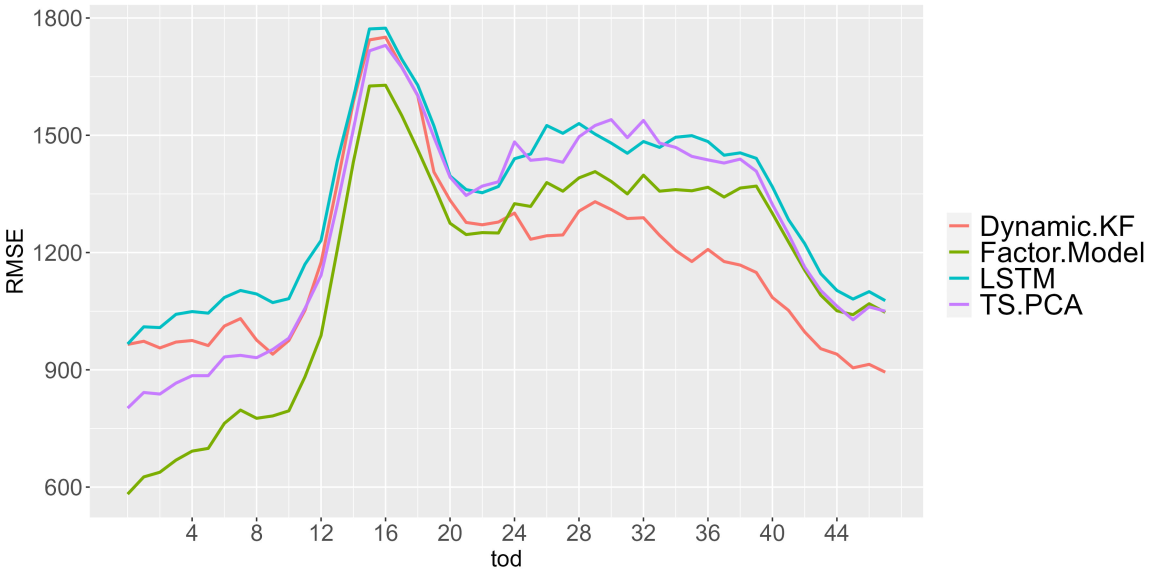

Fig. 4.7 RMSE of predictions from each model by time of day.

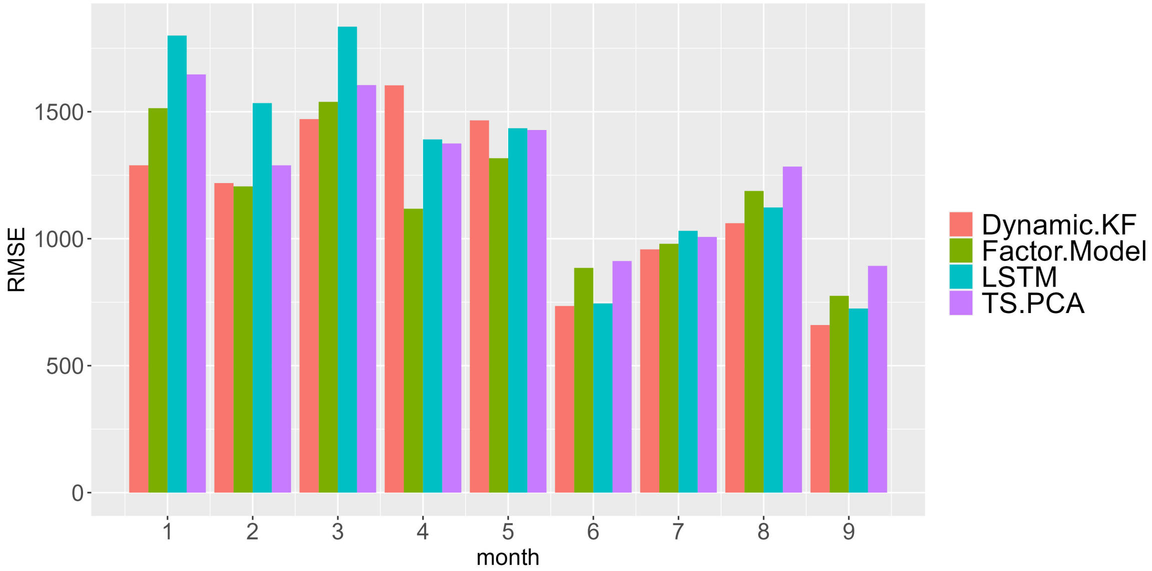

Fig. 4.8 RMSE of predictions from each model by month of year.

Figure 4.7 presents the average RMSE of different methods for each time of day. The average is calculated based on 273 predictions. We observe a similar pattern across all methods: RMSE tends to be higher during morning hours (from tod=12 to 20), and lowest during late night (tod$\geq$44 and tod$\leq$10). The Factor Model performs particularly well in the first half of the day, while the dynamic Kalman filter excels in the second half of the day.

Similarly, Figure 4.8 plots the average RMSE of different methods for each month of the year.Based on our observations, summer months (June to September) appear more predictable than winter months (January to March). When examining each method individually, either the dynamic Kalman filter or the Factor Model consistently performs best across different months.

4.5.1.3 Details in Factor Models: Estimating $\hat{r}$

In this section, we further investigate the prediction from factor models. Recall that factor models assume the existence of underlying latent structure, and the key is to estimate the number of factors within the latent structure. Following the One-Step-Ahead prediction approach, we generated 273 estimations for $\hat{r}$ over the testing period, corresponding to each prediction step. Figure 4.9 plots the 273 estimations in time order, allowing us to observe any trends or fluctuations in the estimated number of factors over time.

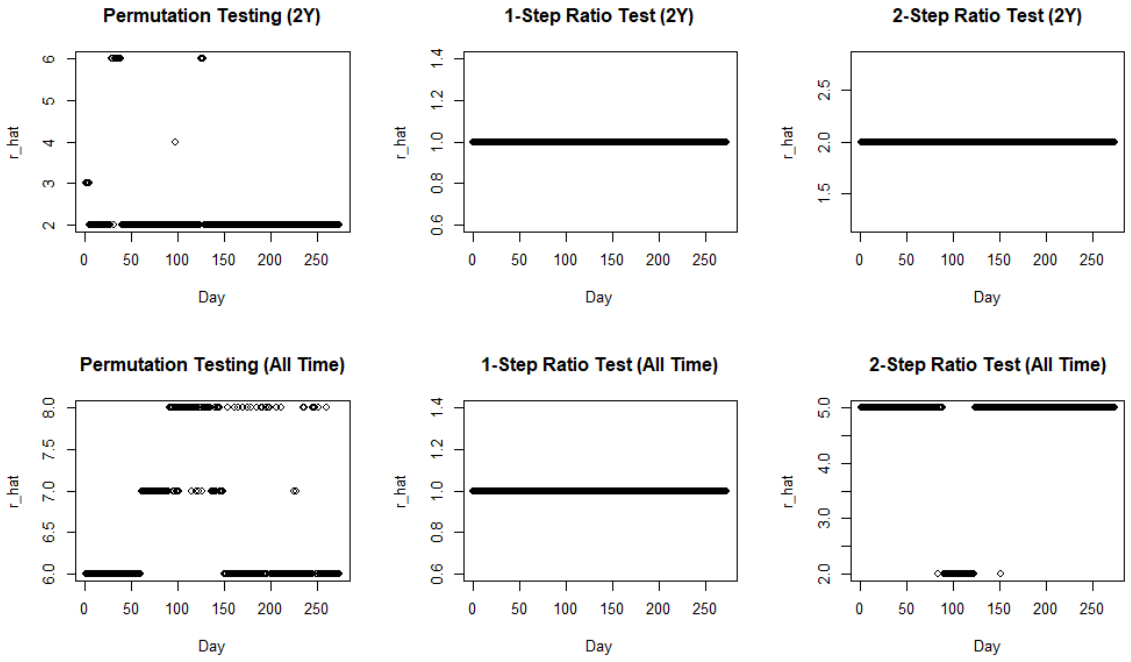

Fig. 4.9 Estimation of $\hat{r}$ from all 3 methods on both training datasets.

From Figure 4.9, we observe that permutation testing generally provides higher $\hat{r}$, with estimated values varying within a small interval of 6 to 8 with the all time training dataset. Given that the permutation testing approach achieves a lower RMSE, we can infer that the true number of factors is likely closer to $\hat{r}_{PT}$, while the ratio-test based methods under-estimates true r. This discrepancy in estimation from the two ratio test estimators is likely due to varying factor strengths. The issue of underestimation of $\hat{r}$ from ratio tests is probably caused by the varying factor strength of the underlying latent process. If each factor has a different strength $\delta_i \neq \delta_j$, $i \neq j$, the ratio test may fail, as it only identifies factors with same factor strength at each step.

Assuming that the underlying latent structure is invariant over time, we may search for the true number of factors by testing a sequence of candidate $\hat{r}$ values, and look for the one that yields the lowest RMSE. Below, we present our findings for the optimal $\hat{r}$ based on this RMSE-minimization search.

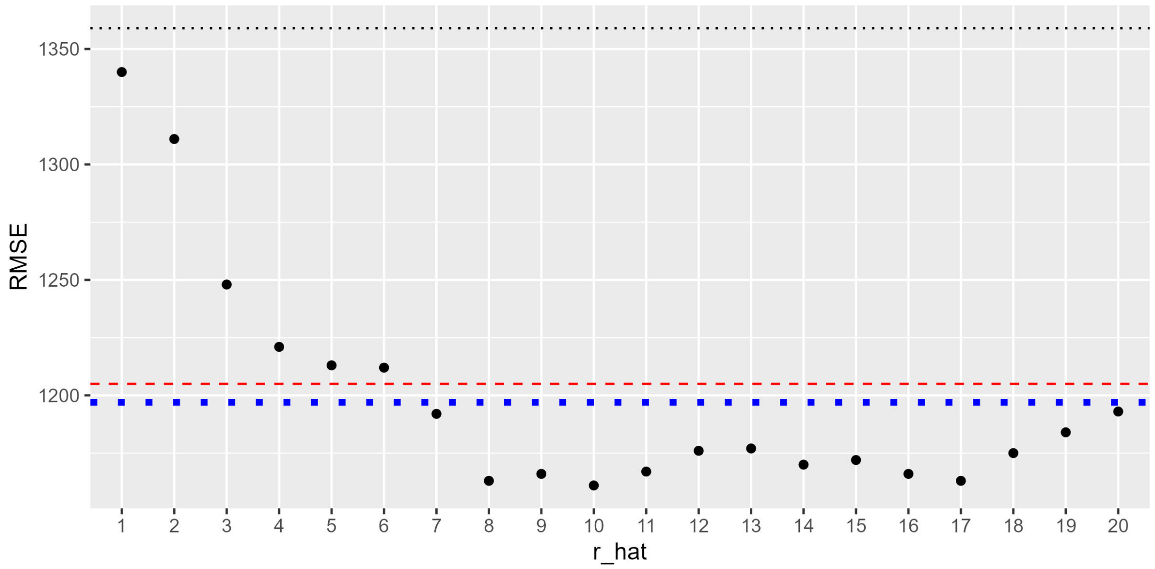

Fig. 4.10 Prediction RMSE with fixing $\hat{r} \in 1,\ldots,20$. The black/red/blue lines represent the RMSE from TS-PCA/ Dynamic Kalman Filter/ Factor Model with Permutation Testing respectively.

In Figure 4.10, we observe lowest RMSE=1161 at $\hat{r} = 10$. The three dotted line represents the RMSE from TS-PCA/dynamic Kalman filter/Factor Model with Permutation Testing, and we can see that any $\hat{r} \geq 7$ gives better RMSE than the best porforming Permutation Testing (RMSE=1197). As the true r should result in lowest RMSE, the figure suggests that r might vary between 8 and 17.

4.5.1.4 Details in TS-PCA

In this section, we present the details of TS-PCA in predicting electricity load on national level. Recall that our objective for deploying TS-PCA and factor model is to reduce the number of parameters required for VAR modeling on the residual series. TS-PCA helps via segmenting data into smaller groups, and reduce inter-group cross correlation through a linear transformation. Below, we present the maximum cross-correlation in (4.8) between components of the vectorized residual series from national data, before and after applying the TS-PCA linear transformation. This comparison illustrates the effectiveness of TS-PCA in simplifying the structure and parameter requirements of the VAR models.

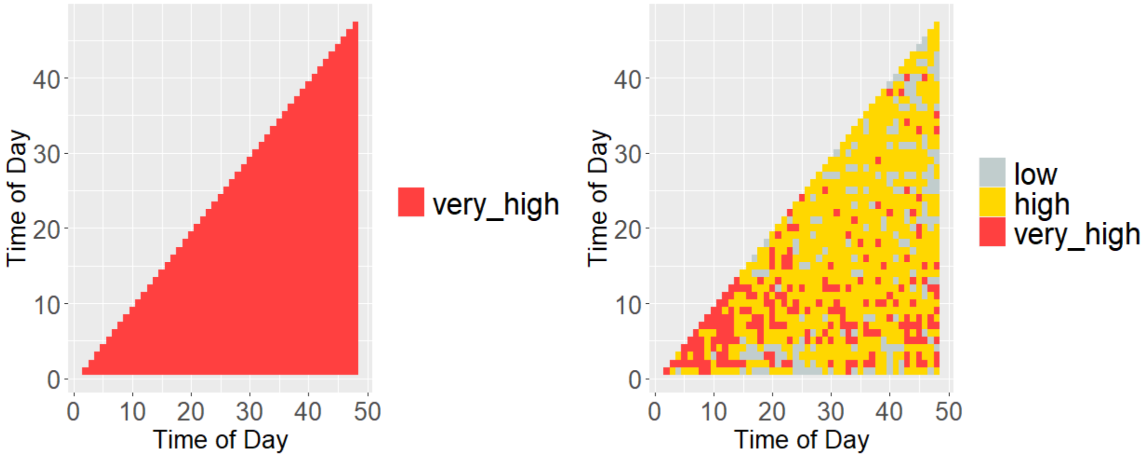

Fig. 4.11 Heatmap of Maximum Cross Correlation among 48 tod for all-time residual data (left), and TS-PCA transformed data (right). $MCC < cv$ as "low", $cv \leq MCC < 2 \times cv$ as high, and $MCC > 2 \times cv$ as "very high". $cv = 0.033$.

Here in Figure 4.11, we classified the numerical values of MCC into 3 levels: low, high and very high, which corresponds with their relationship with the critical value. The critical value is set at $1.96/\sqrt{n} = 0.033$, which is the 95% confidence level for sample cross-correlation of two time series, as commonly used in statistical analysis. The heatmap on the right in Figure 4.11, representing the TS-PCA transformed residual series, shows lower MCC values across all component pairs, indicating reduced inter-component dependencies. This figure also helps explain why a simple VAR model can still perform reasonably well, as many pairs retained significant correlations, even though the TS-PCA transformation reduces cross-correlation overall.

Next, we examine the segmentation of transformed variates in the TS-PCA model. In Table 4.1, we present 2 different grouping approach on the transformed series $z_t = \mathbf{\Gamma}_y^\top y_t$. The Ratio-Based approach was originally proposed in Chang, Guo, and Yao (2018a), and the trimmed grouping method is proposed in section 4.4.1.2. As Table 4.1 shows, the Ratio-Based method is not as good as the trimmed grouping method. Below we compare the segmentation result of the Ratio-Based method, threshold grouping method, and our proposed trimmed grouping method. Threshold is set at 0.08, as stated in model specification section.

Method	Number of Groups	Largest Group Size
Ratio-Based Grouping	47	2
Threshold Grouping	12	36
Trimmed Grouping	18	14

Table 4.4 Comparison of grouping result from different approach in grouping the components of z_t.

From Table 4.4, we see that the segmentation of 47 groups from the Ratio-Based method is very similar to the univariate AR model setting, where we have 48 groups (each component modeled independently). In contrast, when applying a hard threshold to determine the significance of pairwise cross-correlation, the number of groups decreases, although still resulting in groups of unbalanced sizes. Below we present the RMSE of two methods with manually controlled threshold value, allowing us to evaluate the impact of group size balance on prediction accuracy:

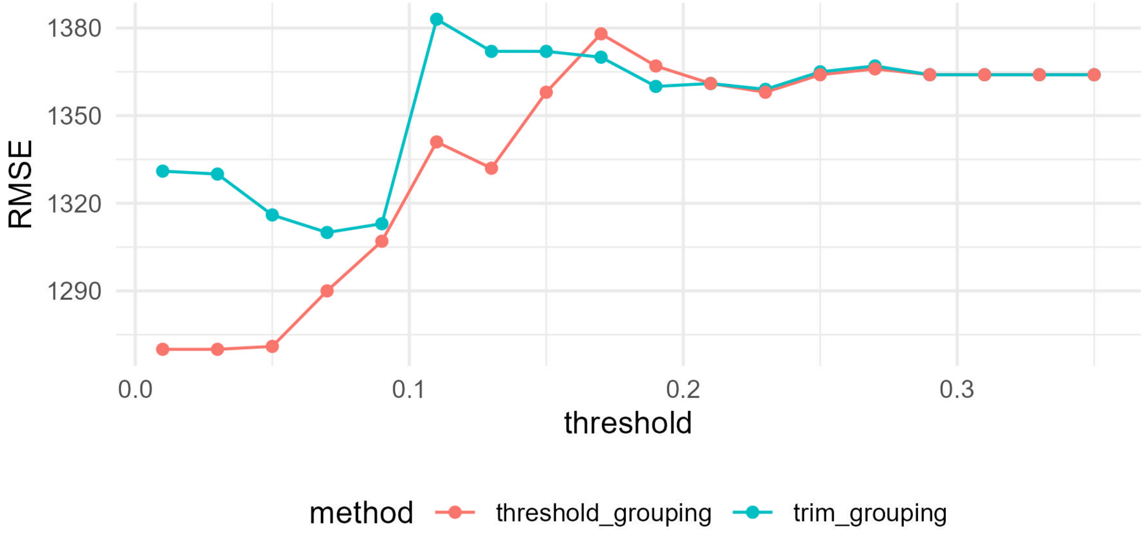

Fig. 4.12 Comparison of RMSE between thresold grouping method and trimmed grouping method.

In Figure 4.12, we discover that the proposed trimmed grouping method does not outperform the thresold grouping method. More specifically, when the threshold is low, the threshold grouping method performs significantly better; when the threshold is high, both methods yield similar results. With a low threshold, the threshold grouping method combines all components into one large group, making the segmentation identical to the VAR model. As observed from the plot, the RMSE is nearly the same to RMSE from VAR in Table 4.1. Conversely, with a high threshold, none of the pairwise correlation is significant enough to be grouped together, thus the segmentation will be similar to the univariate-AR like segmentation from ratio-based grouping. The RMSE of both methods in Figure 4.12 is indeed similar to RMSE of the threshold grouping TS-PCA in Table 4.1. This comparison highlights the sensitivity of the grouping methods to the choice of threshold, as well as the difference between different methods.

The primary motivation for using TS-PCA is to address the over-parameterization issue encountered in VAR modeling. When training on the all-time dataset, the VAR model achieves a reasonable RMSE despite slight over-parameterization. However, if the number of observations is further reduced, the performance of the VAR model deteriorates, and the problem of imbalanced segmentation with the default thresholding method in TS-PCA becomes more obvious. To illustrate this issue, we reduce the training data to 730 observations using the sliding-window approach and generate predictions using the One-Step-Ahead method. Below, we present the RMSE results for TS-PCA under both the trimmed grouping and threshold grouping methods, using a range of threshold values. For the trimmed grouping method, we set an upper limit of $s = 10$ for group size, allowing us to assess how balanced grouping affects prediction performance in a reduced data setting.

Threshold	0.01	0.05	0.09	0.13	0.17	0.21	0.25	0.29	0.33
Trimmed Grouping	1357	1355	1339	1379	1371	1364	1369	1369	1370
Threshold Grouping	6438	6438	13493	15196	1372	1364	1368	1369	1370

Table 4.5 Comparison of RMSE from different approach in grouping the components of z_t, training data is selected via sliding window, window length is 2 years.

In Table 4.5, we found that when threshold is low, threshold grouping has extremely high RMSE, showing similar behavior to VAR model in Table 4.3 under limited observations.

This issue is alleviated with increasing threshold, preventing variables from being grouped together. On the other hand, TS-PCA with trimmed grouping showed stable performance all time, validating its robustness against less-appropriate threshold value and limited sample size. The optimal choice of group size should depends on the number of observations. Since we use AR/VAR models to make prediction for each segmented group, we need to be careful in selecting the upper limit s for groups to ensure that in the VAR model, $s+k\times s^2 \leq n_{obs}$. This condition helps maintain a balance between capturing interactions within groups and avoiding over-parameterization, especially when number of observations are limited

4.5.2 Prediction on Regional Level

In this section, we focus on predicting electricity load at the regional level. The 12 metropolitan regions have the same independent and dependent variables as the national data, enabling us to apply the methods from Table 4.1 to each region individually. However, treating the regions separately would have missed the interactions among them.

To process regional data from all 12 regions at the same time, we first fit a separate GAM model for each region, obtaining a regional residual vector time series with a dimension of $p = 48$ for each of the 12 regions. We then combine all 12 regional residual series and reshape these residual data into a 12×48 matrix time series with 3865 observations. To model and predict residuals as matrix time series, we use MAR models as benchmark, and Matrix Time Series Decorrelation from Section 4.4.3 as the more efficient model that reduces model complexity. Additionally, we also vectorized the 12×48 matrix time series into a $p = 576$ vector and used VAR, TS-PCA, and factor models for predicting the residual series as well.

The performance of each method is evaluated using RMSE. For methods that model each region separately, this forecasting and evaluation process is identical to the national level forecasting evaluation, where we add the forecast residual from proposed models back to GAM estimates, and compute the RMSE between estimate and actual electricity load. For methods applied to the matrix residual series and the vectorized matrix residual series, predictions require an additional step. Once forecasting are generated, we identify the region where each forecasting corresponds to, reconstruct each region-specific residual series, and then calculate RMSE for each region separately. The only exception is for dynamic Kalman filter, which applies on regions separately and directly predicts the regional electricity load instead.

4.5.2.1 Prediction Performance Comparison

Here we present the predictions from selected methods on each region separately. Table 4.6 illustrates the RMSE of predictions from each method. The first two methods, MAR model and Matrix Time Series Decorrelation, are applied directly on the 12×18 variate residual matrix time series, and they are marked with (m). The subsequent TS-PCA and VAR model are based on vectorized residual matrix time series, maked with (v), and the rest are modeled for each region separately, marked with (r). We also ranked the performance of each method within every region, and the rank of summation on RMSE over 12 regions. We present the results in Table 4.7.

Method	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	Sum
Mat-Decorr(m)	486	174	156	129	397	529	491	282	321	260	245	214	3684
MAR(m)	532	184	159	138	412	539	504	272	336	272	268	219	3835
TS-PCA(v)	493	178	148	129	412	553	519	307	317	271	247	221	3795
VAR(v)	486	175	158	133	401	541	494	281	323	265	243	215	3715
Factor_perm(r)	621	192	154	129	500	663	524	314	362	267	248	217	4191
Factor_fixed(r)	556	191	144	124	453	560	555	298	335	280	276	216	3988
DKF(r)	449	164	137	115	388	500	398	255	293	228	203	202	3332
GAM(r)	489	175	162	135	406	551	494	282	327	266	243	217	3747

Table 4.6 RMSE of prediction on 12 regions' electricity load prediction from different methods.

Method	Avg Rank	Sum Rank
Mat-Decorr(m)	2.92	2
MAR(m)	5.88	6
TS-PCA(v)	5.21	5
VAR(v)	3.67	3
Factor_perm(r)	6.62	8
Factor_fixed(r)	6.00	7
DKF(r)	1.00	1
GAM(r)	4.71	4

Table 4.7 Average ranking and ranking of summed RMSE over 12 regions' electricity load prediction from different methods.

From Table 4.6 and 4.7, we see that dynamic Kalman filter has the best performance in summed RMSE over all regions. The basic GAM prediction is only about 12.5% less accurate than dynamic Kalman filter, which is a smaller gap than on the national level (25.5% less accurate). The only 2 methods that showed improvement based on GAM is the Matrix Time Series Decorrelation, and the vectorized VAR model. The simple MAR model, despite not having the over-parameterization issue, did not perform well. On the other hand, we tried factor model with 2 different settings: using permutation testing to identify the number of factors $\hat{r}$ or inherit $\hat{r}$ from national level. As the table showed, both factor models did not perform well, although inheriting $\hat{r}$ from national data performs slightly better. This result suggests that the estimated $\hat{r}$ from permutation testing method was less accurate on regional data, which could be caused by more noisy observations.

Matrix TS Decorrelation segmented the transformed series into several smaller groups, both row-wise (1 group of 7) and column-wise (1 group of 4, 3 groups of 2), and has shown improvement over GAM, while outperforming the MAR model as well. We believe that this enhancement comes from the bilinear transformation that led to efficient decorrelation of the regional residual matrix series. We vectorized the 12×48 residual series, and plotted the heatmap based on MCC before and after matrix TS decorrelation:

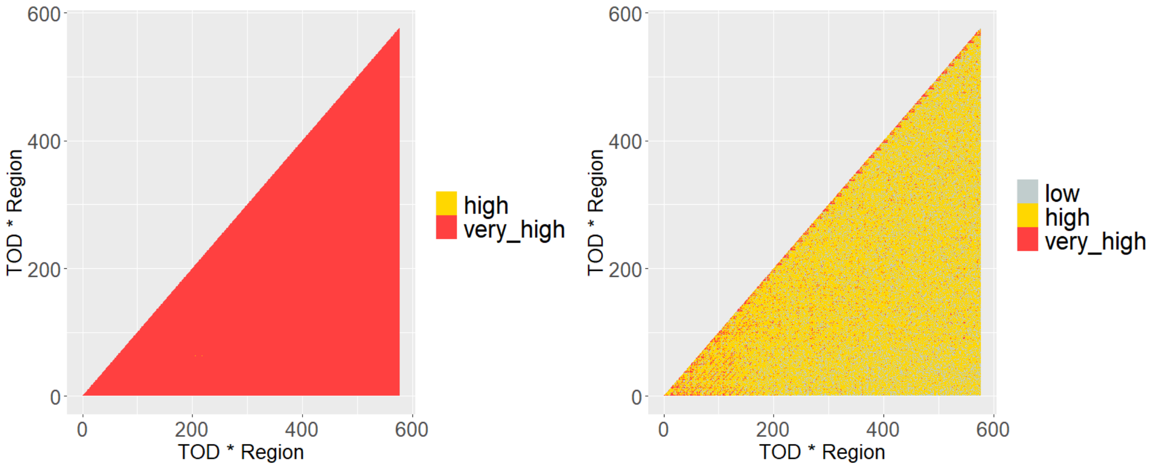

Fig. 4.13 Heatmap of Maximum Cross Correlation among vectorized 12×48 variates for all-time residual residual data (left), and Matrix TS Decorrelation transformed data (right).$MCC < cv$ as "low", $cv \leq MCC < 2 \times cv$ as high, and $MCC > 2 \times cv$ as "very high". $cv = 0.033$.

In Figure 4.13, we observe that MCC has significantly decreased after the bilinear transformation. This reduction justifies the improvement, as unnecessary connection among less-related components are no longer included in the model, reducing noise overall.

4.6 Conclusion

In this section, we attempted to enhance the performance of electricity load forecasts from Generalized Additive Model (GAM). We examined the residual series ε_t^{GAM} from GAM, and has found significance in autocorrelation within each component, as well as cross correlation among the components. To extract the information remained in residual time series ε_t^{GAM}, we proposed to recover the residual series' latent structure, and make predictions $\widehat{\varepsilon}_{t+1}^{GAM}$ using multivariate time series models based on the latent process. Then, forecast on electricity load are corrected using predicted residual via $\tilde{y}_{t+1} = \widehat{y}_{t+1}^{GAM} + \widehat{\varepsilon}_{t+1}^{GAM}$.

Evaluation has been performed based on the corrected electricity load prediction, using RMSE as the measure. We apply the methods mentioned above on electricity load data in France, and compared RMSE of predictions from multivariate time series models, operating model from EDF, and other miscellaneous methods such as Long Short Term Memory.

On national level forecasting, factor model recovered the underlying latent structure of residual series from residuals, and the proposed permutation testing procedure was able to estimate the number of factors better than the ratio test proposed by Lam and Yao (2012a). Factor model with permutation testing outperformed both the time series benchmark model VAR, as well as the EDF operating model dynamic Kalman filter in RMSE. As for TS-PCA, facing challenges due to the strong cross-correlation among components of ε_t^{GAM}, did not find a clear latent segmentation among transformed components. To address this, we introduced a new method for segmenting the components of the vector residual time series into smaller groups with more balanced sizes, which helped alleviate the segmentation issue. This balanced grouping approach enhances TS-PCA's ability to manage cross-correlations effectively, improving its applicability in scenarios with complex dependencies.

For regional-level forecasting, the matrix time series decorrelation method—an adaptation of TS-PCA for matrix time series—identified a latent segmentation within the regional matrix residual series. This method enhanced GAM predictions and outperformed the MAR benchmark model. In contrast, the factor model with permutation testing, applied separately to each region, did not improve the GAM predictions. Furthermore, the number of factors identified in the national-level data did not translate effectively to regional data. Since the permutation testing procedure is designed to handle factors of varying strengths, this outcome suggests that the regional residual series may lack a distinct latent structure at the regional level, making factor models less beneficial in this context.

To summarize the characteristics of various methods on electricity load forecasting, while the GAM framework provides a robust baseline for electricity load forecasting, it has room for improvement because it does not fully capture the time-varying features of the data. Simple time series methods, such as the vector autoregressive (VAR) model, also fall short as they require more parameters than the available number of observations can support. Similarly, although machine learning approaches like LSTM hold potential for improving forecasts through their ability to model complex nonlinear relationships, they too suffer from the issue of over-parameterization. On the other hand, considering the time series nature of electricity load data, incorporating time-related models should be able to significantly enhance forecast quality. For instance, compared to the static Kalman filter, the dynamic Kalman filter adapts more promptly to changes in the data, leading to better performance. In a similar vein, our proposed method effectively links GAM with time series techniques through dimension reduction and data segmentation, thereby overcoming the limitations associated with high-dimensional parameterization encountered by traditional time series methods and neural-network based methods, while capturing essential temporal dynamics.

For future work, further investigation into alternative time series approaches for modeling the residual data obtained after applying TS-PCA or factor models may offer additional potential for improvement. Specifically, these residuals can be handled by other time series methods, potentially capturing further hidden structures. Moreover, while the variable selection process was carried out by the EDF team, an alternative strategy might involve directly utilizing all available variables when applying TS-PCA or factor models, thereby revealing new relationships that were previously overlooked. For instance, incorporating new variables—such as electricity price—could enhance forecast accuracy by integrating market dynamics into the model. Finally, a deeper focus on regional load forecasting is warranted. A more detailed examination of the latent structure of the regional matrix residual series could uncover additional insights into inter-regional dependencies, thereby refining our understanding of regional load patterns. Additionally, treating the regional data as a network with 12 nodes and developing network-based time series models, such as the Generalized Network Autoregressive (GNAR) model proposed by Knight et al. (2019), could offer a fresh perspective on capturing spatial and temporal interactions, ultimately leading to improved forecast performance.

References

Antoniadis, Anestis, Solenne Gaucher, and Yannig Goude (2023). "Hierarchical transfer learning with applications to electricity load forecasting". In: *International Journal of Forecasting* 39.4, pp. 1–14. DOI: 10.1016/j.ijforecast.2023.04.006. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169207023000420.

Bachoc, François et al. (2020). "Spatial blind source separation". In: *Biometrika* 107.3, pp. 627–646.

Bai, Jushan and Serena Ng (2002). "Determining the number of factors in approximate factor models". In.

Benjamini, Yoav and Yosef Hochberg (1995). "Controlling the false discovery rate: A practical and powerful approach to multiple testing". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 57.1, pp. 289–300.

Bickel, Peter J. and Elizaveta Levina (2008). "Covariance regularization by thresholding". In: *The Annals of Statistics* 36.6, pp. 2577–2604. DOI: 10.1214/08-AOS600. URL: https://projecteuclid.org/journals/annals-of-statistics/volume-36/issue-6/Covariance-regularization-by-thresholding/10.1214/08-AOS600.full.

Box, George EP and David A Pierce (1970). "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models". In: *Journal of the American statistical Association* 65.332, pp. 1509–1526.

Brégère, Margaux and Malo Huard (2022). "Online hierarchical forecasting for power consumption data". In: *International Journal of Forecasting* 38.1, pp. 339–351. DOI: 10.1016/j.ijforecast.2021.05.011. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169207021000947.

Canatar, Abdulkadir, Blake Bordelon, and Cengiz Pehlevan (2021). "Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks". In: *Nature communications* 12.1, p. 2914.

Cardoso, J-F (1998). "Multidimensional independent component analysis". In: *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181).* Vol. 4. IEEE, pp. 1941–1944.

Chang, Jinyuan, Bin Guo, and Qiwei Yao (2018a). "Principal component analysis for second-order stationary vector time series". In: *The Annals of Statistics* 46.5, pp. 2045–2083. DOI: 10.1214/17-AOS1613. URL: https://projecteuclid.org/journals/annals-of-statistics/volume-46/issue-5/Principal-component-analysis-for-second-order-stationary-vector-time-series/10.1214/17-AOS1613.full.

— (2018b). "Principal component analysis for second-order stationary vector time series". In: *The Annals of Statistics* 46.5, pp. 2094–2124.

Cho, Haeran et al. (2013). "Modeling and Forecasting Daily Electricity Load Curves: A Hybrid Approach". In: *Journal of the American Statistical Association* 108.501, pp. 7–21.

DOI: 10.1080/01621459.2012.722900. URL: http://www.tandfonline.com/doi/abs/10.1080/01621459.2012.722900.

Clarkson, Douglas B (1988). "Remark AS R71: A remark on algorithm AS 211. The FG diagonalization algorithm". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 37.1, pp. 147–151.

Diebold, Francis X and Robert S Mariano (2002). "Comparing predictive accuracy". In: *Journal of Business & economic statistics* 20.1, pp. 134–144.

Fama, Eugene F and Kenneth R French (2015). "A five-factor asset pricing model". In: *Journal of financial economics* 116.1, pp. 1–22.

— (1993). "Common risk factors in the returns on stocks and bonds". In: *Journal of financial economics* 33.1, pp. 3–56.

Ferréol, Anne, Laurent Albera, and Pascal Chevalier (2005). "Fourth-order blind identification of underdetermined mixtures of sources (FOBIUM)". In: *IEEE Transactions on Signal processing* 53.5, pp. 1640–1653.

Filzmoser, Peter and Maintainer Peter Filzmoser (2015). "Package 'StatDA'". In: *R Package Version* 1.

Fisher, Thomas J and Colin M Gallagher (2012). "New weighted portmanteau statistics for time series goodness of fit testing". In: *Journal of the American Statistical Association* 107.498, pp. 777–787.

Han, Yuefeng et al. (2023). "Simultaneous Decorrelation of Matrix Time Series". In: *Journal of the American Statistical Association* 118.1, pp. 1–13. DOI: 10.1080/01621459.2022.2151448. URL: https://www.tandfonline.com/doi/full/10.1080/01621459.2022.2151448.

Harvey, David, Stephen Leybourne, and Paul Newbold (1997). "Testing the equality of prediction mean squared errors". In: *International Journal of forecasting* 13.2, pp. 281–291.

Hastie, Trevor and Robert Tibshirani (1987). "Generalized additive models: some applications". In: *Journal of the American Statistical Association* 82.398, pp. 371–386.

Hochreiter, S (1997). "Long Short-term Memory". In: *Neural Computation MIT-Press*.

Hyvarinen, Aapo, J Karhunen, and E Oja (2001). *Independent component analysis and blind source separation*.

Kalman, Rudolph Emil (1960). "A new approach to linear filtering and prediction problems". In.

Knight, Marina et al. (2019). "Generalised network autoregressive processes and the GNAR package". In: *arXiv preprint arXiv:1912.04758*.

Kuster, Corentin, Yacine Rezgui, and Monjur Mourshed (2017). "Electrical load forecasting models: A critical systematic review". In: *Sustainable Cities and Society* 35, pp. 257–270. DOI: 10.1016/j.scs.2017.08.009. URL: https://linkinghub.elsevier.com/retrieve/pii/S2210670717305899.

Lam, Clifford and Qiwei Yao (2012a). "Factor modeling for high-dimensional time series: Inference for the number of factors". In: *The Annals of Statistics* 40.2, pp. 694–726. DOI: 10.1214/12-AOS970. URL: https://projecteuclid.org/journals/annals-of-statistics/volume-40/issue-2/Factor-modeling-for-high-dimensional-time-series--Inference-for/10.1214/12-AOS970.full.

— (2012b). "Factor modeling for high-dimensional time series: inference for the number of factors". In: *The Annals of Statistics*, pp. 694–726.

Ljung, Greta M and George EP Box (1978). "On a measure of lack of fit in time series models". In: *Biometrika* 65.2, pp. 297–303.

McLeod, AI (1978). “On the distribution of residual autocorrelations in Box–Jenkins models”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 40.3, pp. 296–302.

Nordhausen, Klaus et al. (2015). “Blind source separation for spatial compositional data”. In: *Mathematical Geosciences* 47, pp. 753–770.

Nti, Isaac Kofi et al. (2020). “Electricity load forecasting: a systematic review”. In: *Journal of Electrical Systems and Information Technology* 7.1, p. 13. DOI: 10.1186/s43067-020-00021-8. URL: https://jesit.springeropen.com/articles/10.1186/s43067-020-00021-8.

Obst, David, Joseph De Vilmarest, and Yannig Goude (2021). “Adaptive Methods for Short-Term Electricity Load Forecasting During COVID-19 Lockdown in France”. In: *IEEE Transactions on Power Systems* 36.5, pp. 4754–4763. DOI: 10.1109/TPWRS.2021.3067551. URL: https://ieeexplore.ieee.org/document/9382417/.

Pierrot, Amandine and Yannig Goude (2011). “Short-term electricity load forecasting with generalized additive models”. In: *Proceedings of ISAP power* 2011.

Reimann, Clemens et al. (2011). *Statistical data analysis explained: applied environmental statistics with R*. John Wiley & Sons.

Romano, Joseph P. and Marius A. Tirlea (2022). “Permutation testing for dependence in time series”. In: *Journal of Time Series Analysis* 43.5, pp. 781–807. DOI: 10.1111/jtsa.12638. URL: https://onlinelibrary.wiley.com/doi/10.1111/jtsa.12638.

Vilmarest, Joseph de et al. (2023). *Adaptive Probabilistic Forecasting of Electricity (Net)-Load*. arXiv:2301.10090 [stat]. URL: http://arxiv.org/abs/2301.10090.

Vilmarest, Joseph de and Olivier Wintenberger (2024). “Viking: variational Bayesian variance tracking”. In: *Statistical Inference for Stochastic Processes*, pp. 1–22.

Wang, Chen et al. (2022). “A Transformer-Based Method of Multienergy Load Forecasting in Integrated Energy System”. In: *IEEE Transactions on Smart Grid* 13.4, pp. 2703–2714. DOI: 10.1109/TSG.2022.3166600. URL: https://ieeexplore.ieee.org/document/9756020/.

Wold, Svante, Kim Esbensen, and Paul Geladi (1987). “Principal component analysis”. In: *Chemometrics and intelligent laboratory systems* 2.1-3, pp. 37–52.

Wood, Simon N (2017). *Generalized additive models: an introduction with R*. chapman and hall/CRC.

Xu, Xiuqin et al. (2020). *Probabilistic Forecasting for Daily Electricity Loads and Quantiles for Curve-to-Curve Regression*. arXiv:2009.01595 [stat]. URL: http://arxiv.org/abs/2009.01595.

Ziehe, Andreas and Klaus-Robert Müller (1998). “TDSEP—an efficient algorithm for blind separation using time structure”. In: *ICANN 98: Proceedings of the 8th International Conference on Artificial Neural Networks, Skövde, Sweden, 2–4 September 1998 8*. Springer, pp. 675–680.

Appendix A

Supplementary Materials for Chapter 2

A.1 Some Useful Lemmas

C_0 which is defined in Assumption A1 and A which is defined in Assumption A2 are two important notations in our proofs. Without loss of generality, we assume that $C_0 \leq A$. It means that

$$\sup_{\beta \geq 1, 1 \leq i \leq p} \beta^{-1/2} \{E|Z_i(s)|^{\beta}\}^{1/\beta} \leq A. \tag{A.1}$$

Thus, any fixed moment of $Z_g(s)$ can be bounded by a constant only depending on A.

Let Z be the $p \times n$ matrix with $(Z_i(s_1), \cdots, Z_i(s_n)) = Z^i$ as its i-th row.

Lemma A.1.1. *Let Assumptions 2.3.1 and 2.3.2 hold, and $p = o(n)$. Then there exists λ_{max} depending only on A such that*

$$\max_{1 \leq g \leq p} \lambda_g \leq \lambda_{max} < \infty. \tag{A.2}$$

Proof. For any $g = 1, \cdots, p$, (2.6) implies that

$$\lambda_g = \frac{1}{k} \sum_{h=1}^{k} \sum_{u=1}^{p} E[\frac{1}{n} \sum_{i,j=1}^{n} f_h(s_i - s_j) \tilde{Z}_g(s_i) \tilde{Z}_u(s_j)]^2 \tag{A.3}$$

$$= \quad \frac{1}{k} \sum_{h=1}^{k} \sum_{u \neq g} E[\frac{1}{n} \sum_{i,j=1}^{n} f_h(s_i - s_j) \tilde{Z}_g(s_i) \tilde{Z}_u(s_j)]^2 + \frac{1}{k} \sum_{h=1}^{k} E[\frac{1}{n} \sum_{i,j=1}^{n} f_h(s_i - s_j) \tilde{Z}_g(s_i) \tilde{Z}_g(s_j)]^2.$$

We consider the first part $u \neq g$ for each h,

$$\begin{aligned}
& \sum_{u\neq g} E[\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_g(s_i)\tilde{Z}_u(s_j)]^2 \\
= & \sum_{u\neq g} \frac{1}{n^2} \sum_{i,j,\tilde{i},\tilde{j}=1}^{n} f_h(s_i - s_j) f_h(s_{\tilde{i}} - s_{\tilde{j}}) E[\tilde{Z}_g(s_i)\tilde{Z}_u(s_j)\tilde{Z}_g(s_{\tilde{i}})\tilde{Z}_u(s_{\tilde{j}})] \\
= & \sum_{u\neq g} \frac{1}{n^2} \sum_{i,j,\tilde{i},\tilde{j}=1}^{n} f_h(s_i - s_j) f_h(s_{\tilde{i}} - s_{\tilde{j}}) E[\tilde{Z}_g(s_i)\tilde{Z}_g(s_{\tilde{i}})] E[\tilde{Z}_u(s_j)\tilde{Z}_u(s_{\tilde{j}})] \\
\leq & \sum_{u\neq g} \frac{1}{n^2} \sum_{i,j,\tilde{i},\tilde{j}=1}^{n} \frac{A}{1+\|s_i - s_j\|^{d+\alpha}} \frac{A}{1+\|s_{\tilde{i}} - s_{\tilde{j}}\|^{d+\alpha}} \frac{A}{1+\|s_i - s_{\tilde{i}}\|^{d+\alpha}} \frac{A}{1+\|s_j - s_{\tilde{j}}\|^{d+\alpha}}.
\end{aligned}$$

The last inequality is from (2.12) and (2.13). This, together with $p = o(n)$ and $\|s_i - s_j\| \geq \triangle$ for all $n \geq 2$ and $1 \leq i \neq j \leq n$, implies that

$$\frac{1}{k}\sum_{h=1}^{k}\sum_{u\neq g} E[\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_g(s_i)\tilde{Z}_u(s_j)]^2 = O(A^4 n^{-1} p) = o(1). \tag{A.4}$$

Thus we only need to consider $E[\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_g(s_i)\tilde{Z}_g(s_j)]^2$. Since $Z^g = (Z_g(s_1), \cdots, Z_g(s_n))$ and

$$(\tilde{Z}_g(s_1), \cdots, \tilde{Z}_g(s_n)) = Z^g[I_n - n^{-1}1_{n\times n}]. \tag{A.5}$$

We can rewrite it as $E(\frac{1}{n}Z^g[I_n - n^{-1}1_{n\times n}]T_h[I_n - n^{-1}1_{n\times n}](Z^g)^\top)^2$, where T_h is a $n \times n$ matrix with the (i,j)th entry $f_h(s_i - s_j)/2 + f_h(s_j - s_i)/2$. Note that $\frac{1}{n}Z^g[I_n - n^{-1}1_{n\times n}]T_h[I_n - n^{-1}1_{n\times n}](Z^g)^\top$ is a quadratic form and $Z_g(s)$ is a sub-Gaussian process. (2.13) implies that $\|T_h\| \leq \tilde{C}$, where $\tilde{C}$ only depends on A. These, together with (2.12), imply that there exists a positive constant $\tilde{C}_1$ depending only on A such that

$$\frac{1}{k}\sum_{h=1}^{k} E[\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_g(s_i)\tilde{Z}_g(s_j)]^2 \leq \tilde{C}_1.$$

This, together with (A.3)- (A.4), implies that $\lambda_g \leq 2\tilde{C}_1$, for any $1 \leq g \leq p$. We complete the proof. □

Lemma A.1.2. *Let Assumptions A1 and A2 hold. For any $n \times n$ non-random symmetric matrix Q with bounded $\|Q\|$, there exists a constant $C > 0$ depending only on A and λ_{max} for which*

$$\max_{1 \le g,u \le p} var[\frac{1}{n} \sum_{i,j=1}^{n} Q_{ij} Z_g(s_i) Z_u(s_j)] \le C\|Q\|^2 n^{-1}. \tag{A.6}$$

Here Q_{ij} is the $(i,j)-th$ entry of Q.

Proof. When $g \neq u$, from the independence between $Z_g(s_i)$ and $Z_u(s_j)$ we have

$$\begin{aligned}
& var[\frac{1}{n} \sum_{i,j=1}^{n} Q_{ij} Z_g(s_i) Z_u(s_j)] \\
= \quad & n^{-2} \sum_{i_1,j_1,i_2,j_2=1}^{n} Q_{i_1 j_1} Q_{i_2 j_2} E[Z_g(s_{i_1}) Z_u(s_{j_1}) Z_g(s_{i_2}) Z_u(s_{j_2})] \\
= \quad & n^{-2} \sum_{i_1,j_1,i_2,j_2=1}^{n} Q_{i_1 j_1} Q_{i_2 j_2} E[Z_g(s_{i_1}) Z_g(s_{i_2})] E[Z_u(s_{j_1}) Z_u(s_{j_2})] \\
\le \quad & n^{-2} \sum_{i_1,j_1,i_2,j_2=1}^{n} Q_{i_1 j_1} Q_{i_2 j_2} \frac{A}{1+\|s_{i_1} - s_{i_2}\|^{d+\alpha}} \frac{A}{1+\|s_{j_1} - s_{j_2}\|^{d+\alpha}} \\
\le \quad & C\|Q\|^2 n^{-1}.
\end{aligned}$$

The first inequality is from (2.12) and (2.13). The second inequality is from $\|s_i - s_j\| \ge \triangle$ for all $n \ge 2$ and $1 \le i \neq j \le n$. When $g = u$, we note that $\frac{1}{n} \sum_{i,j=1}^{n} Q_{ij} Z_g(s_i) Z_g(s_j)$ is a quadratic form and $Z_g(s)$ is a sub-Gaussian process. This completes the proof. □

Lemma A.1.3. *Let Assumptions A1 and A2 hold, and $p = o(n)$. Then there exists a positive constant C_A depending only on A such that*

$$\lim_{n \to \infty} P(n^{-1}\|Z\|^2 \le C_A) = 1. \tag{A.7}$$

Proof. For any fixed $1 \times n$ unit vector $x = (x_1, \cdots, x_n)$, we denote $xZ^\top$ by $z(x) = \Big(z_1(x), \cdots, z_p(x)\Big)$. Since $Z_1(\cdot), \cdots, Z_p(\cdot)$ are independent, the elements of $z(x)$ are independent. (2.12) implies that $\max_{1 \le j \le p} Ez_j^2(x) \le \tilde{C}_A$ where $\tilde{C}_A$ only depends on A.

$$xZ^\top Z x^\top = \sum_{j=1}^{p} [z_j^2(x) - Ez_j^2(x)] + \sum_{j=1}^{p} Ez_j^2(x) \le \sum_{j=1}^{p} [z_j^2(x) - Ez_j^2(x)] + p\tilde{C}_A.$$

By the sub-Gaussian property of $Z(s)$, we can conclude that for any fixed $1 \times p$ unit vector x and any $c > 0$ there exists $\tilde{C}_{A,1}$ depending only on A and c such that

$$P\Big(\|xZ^{\top}\|^2 > \tilde{C}_{A,1}(n+p)\Big) \leq c\exp(-5(n+p)). \tag{A.8}$$

As we know, the unit Euclidean sphere S^{n-1} consists of all n-dimensional unit vectors x. Unfortunately the cardinality of S^{n-1} is uncountable cardinal number. We can't use (A.8) to derive an upper bound of $\|Z\|^2$ directly. Thus we introduce a method based on nets to control $\|Z\|^2$. The basic idea is as follows. We define a subset of S^{n-1} as S_{ε} satisfying $\max_{x\in S^{n-1}} \min_{y\in S_{\varepsilon}} \|x-y\| \leq \varepsilon$. S_{ε} is a so-called net of S^{n-1} and the cardinality of S_{ε} is bounded by $(1+2\varepsilon^{-1})^n$. Thus we can control $\max_{y\in S_{\varepsilon}} \|Zy^{\top}\|$ in probability by (A.8). Finally, we can control the difference between $\max_{y\in S_{\varepsilon}} \|Zy^{\top}\|$ and $\max_{x\in S^{n-1}} \|Zx^{\top}\|$.

Let S_{ε} be a subset of S^{n-1}. For any $x \in S^{n-1}$, there exists $\tilde{x} \in S_{\varepsilon}$ such that $\|\tilde{x}-x\| \leq \varepsilon$. This, together with (A.8) and $|S_{\varepsilon}| \leq (1+2\varepsilon^{-1})^n$, implies that

$$P\Big(\max_{\tilde{x}\in S_{1/2}} \|Z\tilde{x}^{\top}\|^2 > \tilde{C}_{A,1}(n+p)\Big) \leq c|S_{1/2}|\exp(-5n-5p) \leq c5^n\exp(-5n-5p). \tag{A.9}$$

Then if $\|Zx^{\top}\| = \|Z\|$, there exists $\tilde{x} \in S_{\varepsilon}$ such that

$$\|Z\tilde{x}^{\top}\| \geq \|Zx^{\top}\| - \|Z(\tilde{x}-x)^{\top}\| \geq \|Z\| - \varepsilon\|Z\| = (1-\varepsilon)\|Z\|.$$

Let $\varepsilon = 1/2$,

$$\|Z\|^2 \leq 4\max_{\tilde{x}\in S_{1/2}} \|Z\tilde{x}^{\top}\|^2.$$

This, together with (A.9), implies that

$$P\Big(\|Z\|^2 > 4\tilde{C}_{A,1}(n+p)\Big) \leq c|S_{1/2}|\exp(-5n-5p) \leq c5^n\exp(-5n-5p). \tag{A.10}$$

Then (A.7) is implied by (A.10) and $p = o(n)$. □

Definition A.1.1.

$$\widehat{N} = \frac{1}{k}\sum_{h=1}^{k}\{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}(s_i)\tilde{Z}(s_j)^{\top}\}\{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}(s_i)\tilde{Z}(s_j)^{\top}\}^{\top}. \tag{A.11}$$

Lemma A.1.4. *Let Assumptions A1 and A2 hold, and $p = o(n)$. Let M_{gu} be the (g,u)-th entry of $\hat{N} - N$. There exists a positive constant C_1 depending only on A such that*

$$\max_{1\le g,u\le p} EM_{gu}^2 \le C_1 n^{-1}. \tag{A.12}$$

Proof. Since N is diagonal, when $g \neq u$,

$$M_{gu} = \frac{1}{k}\sum_{h=1}^{k}\sum_{\tilde{u}=1}^{p}[\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_g(s_i)\tilde{Z}_{\tilde{u}}(s_j)][\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_u(s_i)\tilde{Z}_{\tilde{u}}(s_j)].$$

Divide the term on the RHS of the above equation into three terms: (i)$\tilde{u} = g$, (ii)$\tilde{u} = u$ and (iii) $\tilde{u} \neq g,u$. We control each term as follows. When $\tilde{u} = g$,

$$E\Big([\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_g(s_i)\tilde{Z}_g(s_j)][\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_u(s_i)\tilde{Z}_g(s_j)]\Big) = 0.$$

$$\begin{aligned}
& var\Big([\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_g(s_i)\tilde{Z}_g(s_j)][\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_u(s_i)\tilde{Z}_g(s_j)]\Big) \\
= \quad & E\Big([\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_g(s_i)\tilde{Z}_g(s_j)][\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_u(s_i)\tilde{Z}_g(s_j)]\Big)^2 \\
= \quad & E\Big(n^{-4}\sum_{i_1,i_2,i_3,i_4,j_1,j_2,j_3,j_4=1}^{n} f_h(s_{i_1} - s_{j_1})f_h(s_{i_2} - s_{j_2})f_h(s_{i_3} - s_{j_3})f_h(s_{i_4} - s_{j_4}) \\
& \tilde{Z}_g(s_{i_1})\tilde{Z}_g(s_{i_1})\tilde{Z}_g(s_{i_1})\tilde{Z}_g(s_{i_1})\tilde{Z}_g(s_{j_1})\tilde{Z}_g(s_{j_3})\tilde{Z}_u(s_{j_2})\tilde{Z}_u(s_{j_4})\Big) \\
\le \quad & n^{-4}\sum_{i_1,i_2,i_3,i_4,j_1,j_2,j_3,j_4=1}^{n}[\prod_{v=1}^{4}\frac{A}{1+\|s_{i_v} - s_{j_v}\|^{d+\alpha}}]\frac{A}{1+\|s_{j_2} - s_{j_4}\|^{d+\alpha}}EZ_g^6(s) \\
\le \quad & \tilde{C}_1 n^{-1},
\end{aligned}$$

where $\tilde{C}_1$ only depends on A. The first inequality is from (2.12)-(2.13) and the independence between $Z_g(\cdot)$ and $Z_u(\cdot)$. The second inequality is from (2.11), $C_0 \leq A$ and $\|s_i - s_j\| \geq \triangle$ for all $n \geq 2$ and $1 \leq i \neq j \leq n$.

Thus we can control

$$(\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_g(s_i)\tilde{Z}_g(s_j))(\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_u(s_i)\tilde{Z}_g(s_j)).$$

When $\tilde{u} = u$, we can repeat the above method to control

$$(\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_g(s_i)\tilde{Z}_u(s_j))(\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_u(s_i)\tilde{Z}_u(s_j)).$$

Let's consider the third term

$$\sum_{\tilde{u}\neq g,u}(\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_g(s_i)\tilde{Z}_{\tilde{u}}(s_j))(\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i - s_j)\tilde{Z}_u(s_i)\tilde{Z}_{\tilde{u}}(s_j)).$$

We can rewrite it as

$$\begin{aligned} & \frac{1}{n^2}\sum_{\tilde{u}\neq g,u}\sum_{i,j,\tilde{i},\tilde{j}=1}^{n} f_h(s_i - s_j)f_h(s_{\tilde{i}} - s_{\tilde{j}})\tilde{Z}_g(s_i)\tilde{Z}_u(s_{\tilde{i}})\tilde{Z}_{\tilde{u}}(s_j)\tilde{Z}_{\tilde{u}}(s_{\tilde{j}}) \\ = \quad & \frac{1}{n}\sum_{i,\tilde{i}=1}^{n}\Big(\frac{1}{n}\sum_{j,\tilde{j}=1}^{n} f_h(s_i - s_j)f_h(s_{\tilde{i}} - s_{\tilde{j}})\sum_{\tilde{u}\neq g,u}\tilde{Z}_{\tilde{u}}(s_j)\tilde{Z}_{\tilde{u}}(s_{\tilde{j}})\Big)\tilde{Z}_g(s_i)\tilde{Z}_u(s_{\tilde{i}}). \end{aligned}$$

Let $\tilde{H}$ be a $n \times n$ symmetric matrix with $(i,\tilde{i})$th entry

$$\frac{1}{n}\sum_{j,\tilde{j}=1}^{n} f_h(s_i - s_j)f_h(s_{\tilde{i}} - s_{\tilde{j}})\sum_{\tilde{u}\neq g,u}\tilde{Z}_{\tilde{u}}(s_j)\tilde{Z}_{\tilde{u}}(s_{\tilde{j}}).$$

Recalling (A.5) and (A.6), we define $Q = (I_n - n^{-1}1_{n\times n})\tilde{H}(I_n - n^{-1}1_{n\times n})$. Although Q is random, we can find that Q is independent of $Z_g(s)$ and $Z_u(s)$. It's easy to see

$$E\frac{1}{n}\sum_{i,j=1}^{n} Q_{i,j}Z_g(s_i)Z_u(s_j) = 0.$$

$$
\begin{aligned}
& var[\frac{1}{n}\sum_{i,j=1}^{n} Q_{i,j}Z_g(s_i)Z_u(s_j)] = E[\frac{1}{n}\sum_{i,j=1}^{n} Q_{i,j}Z_g(s_i)Z_u(s_j)]^2 \\
= \quad & \frac{1}{n^2}\sum_{i,j,\tilde{i},\tilde{j}=1}^{n} E(Q_{i,j}Q_{\tilde{i},\tilde{j}})E[Z_g(s_i)Z_g(s_{\tilde{i}})]E[Z_u(s_j)Z_u(s_{\tilde{j}})] \\
\leq \quad & \frac{1}{n^2}\sum_{i,j,\tilde{i},\tilde{j}=1}^{n} (EQ_{i,j}^2)^{1/2}(EQ_{\tilde{i},\tilde{j}}^2)^{1/2}\frac{A}{1+(s_i-s_{\tilde{i}})^{d+\alpha}}\frac{A}{1+(s_j-s_{\tilde{j}})^{d+\alpha}} \\
\leq \quad & \frac{\tilde{C}_2}{n^2}\sum_{i,j=1}^{n} EQ_{i,j}^2 = \frac{\tilde{C}_2}{n^2}E\|Q\|_F^2,
\end{aligned}
$$

where $\tilde{C}_2$ only depends on A and the first inequality is from (2.12). The second inequality is from $\|s_i - s_j\| \geq \triangle$ for all $n \geq 2$ and $1 \leq i \neq j \leq n$. Recalling the definition of Q, we can rewrite it as

$$
Q = \frac{1}{n}(I_n - n^{-1}1_{n\times n})V_h(I_n - n^{-1}1_{n\times n})Z_{-g,-u}^{\top}Z_{-g,-u}(I_n - n^{-1}1_{n\times n})V_h^{\top}(I_n - n^{-1}1_{n\times n}),
$$

where V_h has the (i,j)th entry $f_h(s_i - s_j)$ and $Z_{-g,-u}$ is a $(p-2)\times n$ matrix without Z^g and Z^u. Then

$$
\|Q\|_F^2 \leq \|V_h\|^4\|\frac{1}{n}Z_{-g,-u}^{\top}Z_{-g,-u}\|_F^2 \leq \tilde{C}_3\|\frac{1}{n}Z^{\top}Z\|_F^2,
$$

where $\tilde{C}_3$ only depends on A and the last inequality is from (2.13). Moreover,

$$
\begin{aligned}
& E\|\frac{1}{n}Z^\top Z\|_F^2 = E\|\frac{1}{n}ZZ^\top\|_F^2 \\
= \quad & E\sum_{g,u=1}^{p}[n^{-1}\sum_{i=1}^{n} Z_g(s_i)Z_u(s_i)]^2 \\
= \quad & E\sum_{1\le g\ne u\le p}[n^{-1}\sum_{i=1}^{n} Z_g(s_i)Z_u(s_i)]^2 + E\sum_{g=1}^{p}[n^{-1}\sum_{i=1}^{n} Z_g^2(s_i)]^2 \\
= \quad & \sum_{1\le g\ne u\le p} n^{-2}\sum_{i,j=1}^{n} E[Z_g(s_i)Z_g(s_j)]E[Z_u(s_i)Z_u(s_j)] + \sum_{g=1}^{p} n^{-2}\sum_{i,j=1}^{n} E[Z_g^2(s_i)Z_g^2(s_j)] \\
\le \quad & \sum_{1\le g\ne u\le p} n^{-2}\sum_{i,j=1}^{n}(\frac{A}{1+\|s_i-s_j\|^{d+\alpha}})^2 + \sum_{g=1}^{p} EZ_g^4(s) \\
\le \quad & \tilde{C}_4 p,
\end{aligned}
$$

where $\tilde{C}_4$ only depends on A. The first inequality is from (2.12). The second equation is from (2.11), $C_0 \le A$, $p = o(n)$ and $\|s_i - s_j\| \ge \triangle$ for all $n \ge 2$ and $1 \le i \ne j \le n$. Then we can conclude that

$$E\|Q\|_F^2 \le \tilde{C}_5 p,$$

where $\tilde{C}_5$ only depends on A. From $p = o(n)$,

$$var[\frac{1}{n}\sum_{i,j=1}^{n} Q_{i,j}Z_g(s_i)Z_u(s_j)] \le \frac{\tilde{C}_2\tilde{C}_5}{n^2}p = o(n^{-1}).$$

Thus we control the third term and prove (A.12) for $g \ne u$. When $g = u$, the proof is similar. □

Definition A.1.2. *Let J_1 and J_2 be two subsets of $\{1,\cdots,p\}$. Let $\hat{N}_{J_1,J_2}$ be the sub-matrix of $\hat{N}$ consisting of the rows with the indices in J_1 and the columns with the indices in J_2. Write $\hat{N}_{J_1} = \hat{N}_{J_1,J_1}$.*

Lemma A.1.5. *Under the conditions of Lemma A.1.3 and $J_1 \cap J_2 = \emptyset$, we define the event $B_Z = \{n^{-1}\|Z\|^2 \le C_A\}$.*

Then there exists a positive constant C_2 depending only on A, c and v such that

$$P\Big(\|\hat{N}_{J_1,J_2}\|^2 > C_2 n^{-1} v(|J_1|+|J_2|)\Big| B_Z\Big) \le c(5^{|J_1|}+5^{|J_2|})\exp(-5|J_1|v - 5|J_2|v). \quad \text{(A.13)}$$

Here $v > 0$ can be finite or tending to infinite.

Proof. Since k is finite, it's sufficient to prove (A.13) on

$$n^{-2} Z_{J_1}(I_n - n^{-1}1_{n\times n})V_h(I_n - n^{-1}1_{n\times n})Z^\top Z(I_n - n^{-1}1_{n\times n})V_h^\top(I_n - n^{-1}1_{n\times n})Z_{J_2}^\top,$$

where Z_{J_1} is a sub-matrix of Z with ith row if and only if $i \in J_1$. V_h is a $n\times n$ matrix with the (i,j)th entry $f_h(s_i - s_j)$. We define $\tilde{V}_h = (I_n - n^{-1}1_{n\times n})V_h(I_n - n^{-1}1_{n\times n})$.

$$\begin{aligned} & Z_{J_1}\tilde{V}_h Z^\top Z \tilde{V}_h^\top Z_{J_2}^\top \\ = \quad & Z_{J_1}\tilde{V}_h Z_{J_1}^\top Z_{J_1}\tilde{V}_h^\top Z_{J_2}^\top + Z_{J_1}\tilde{V}_h Z_{J_2}^\top Z_{J_2}\tilde{V}_h^\top Z_{J_2}^\top \\ + \quad & Z_{J_1}\tilde{V}_h Z_J^\top Z_J \tilde{V}_h^\top Z_{J_2}^\top, \end{aligned} \quad \text{(A.14)}$$

where J is the complementary set of $J_1 \cup J_2$. At first we deal with $Z_{J_1}\tilde{V}_h Z_{J_1}^\top Z_{J_1}\tilde{V}_h^\top Z_{J_2}^\top$.

$$\begin{aligned} & \|Z_{J_1}\tilde{V}_h Z_{J_1}^\top Z_{J_1}\tilde{V}_h^\top Z_{J_2}^\top\|^2 \\ = \quad & \|Z_{J_2}\tilde{V}_h Z_{J_1}^\top Z_{J_1}\tilde{V}_h^\top Z_{J_1}^\top Z_{J_1}\tilde{V}_h Z_{J_1}^\top Z_{J_1}\tilde{V}_h^\top Z_{J_2}^\top\| \\ = \quad & \|Z_{J_2}H_{h,J_1}Z_{J_2}^\top\|, \end{aligned}$$

where

$$H_{h,J_1} = \tilde{V}_h Z_{J_1}^\top Z_{J_1}\tilde{V}_h^\top Z_{J_1}^\top Z_{J_1}\tilde{V}_h Z_{J_1}^\top Z_{J_1}\tilde{V}_h^\top$$

is a $n \times n$ symmetric matrix with rank $|J_1|$ at most. Since $J_1 \cap J_2 = \emptyset$, H_{h,J_1} and $Z_{J_2}^\top$ are independent. Moreover, under the event $B_Z = \{n^{-1}\|Z\|^2 \leq C_A\}$,

$$\|H_{h,J_1}\| \leq \|\tilde{V}_h\|^4 \|Z_{J_1}^\top Z_{J_1}\|^3 \leq \|V_h\|^4 \|Z^\top Z\|^3 \leq \|V_h\|^4 n^3 C_A^3.$$

It follows that

$$\lim_{n \to \infty} P(\|H_{h,J_1}\| \leq n^3 \tilde{C}_A | B_Z) = 1, \tag{A.15}$$

where $\tilde{C}_A$ only depends on A. Now we recall the rank of H_{h,J_1} is not larger than $|J_1|$. For given H_{h,J_1}, we can do eigen-decomposition on it as follows.

$$H_{h,J_1} = U_{h,J_1} \Lambda_{h,J_1} U_{h,J_1}^\top, \tag{A.16}$$

where U_{h,J_1} is a $n \times |J_1|$ matrix and Λ_{h,J_1} is a $|J_1| \times |J_1|$ diagonal matrix. $U_{h,J_1}^\top U_{h,J_1} = I_{|J_1|}$. Then

$$\|Z_{J_1} \tilde{V}_h Z_{J_1}^\top Z_{J_1} \tilde{V}_h^\top Z_{J_2}^\top\|^2 \leq \|Z_{J_2} U_{h,J_1}\|^2 \|\Lambda_{h,J_1}\|.$$

Since $\|\Lambda_{h,J_1}\|$ can be controlled by (A.15), we only need to consider $\|Z_{J_2} U_{h,J_1}\|^2$. Let $Y = Z_{J_2} U_{h,J_1}$ be a $|J_2| \times |J_1|$ matrix with the (i,j)th entry Y_{ij}. The independence between the rows of Z_{J_2} implies the independence between the rows of Y.

For any fixed $1 \times |J_1|$ unit vector $x = (x_1, \cdots, x_{|J_1|})$, we define $xY^\top$ as $Y(x) = (y_1(x), \cdots, y_{|J_2|}(x))$. Then the elements of $Y(x)$ are independent.

$$xY^\top Y x^\top = \sum_{j=1}^{|J_2|} [y_j^2(x) - Ey_j^2(x)] + \sum_{j=1}^{|J_2|} Ey_j^2(x).$$

$Yx^\top = Z_{J_2}U_{h,J_1}x^\top$ and $U_{h,J_1}x^\top$ is an unit vector independent of Z_{J_2}. By the sub-Gaussian property of $Z(s)$, we have

$$xY^\top Yx^\top \leq \sum_{j=1}^{|J_2|}[y_j^2(x) - Ey_j^2(x)] + |J_2|\tilde{C}_{A,2},$$

where $\tilde{C}_{A,2}$ only depends on A. Moreover, we can also deal with $\sum_{j=1}^{|J_2|}[y_j^2(x) - Ey_j^2(x)]$ with the sub-Gaussian property of $Z(s)$. Thus, for any fixed $1 \times |J_1|$ unit vector x, any $c > 0$ and $v > 0$, there exists $C_{A,3}$ depending only on A, c and v such that

$$P\Big(||xY^\top||^2 > C_{A,3}v(|J_1| + |J_2|)\Big|B_Z\Big) \leq c\exp(-5|J_1|v - 5|J_2|v). \tag{A.17}$$

As we know, the unit Euclidean sphere $S^{|J_1|-1}$ consists of all $|J_1|$-dimensional unit vectors x. Unfortunately, the cardinality of $S^{|J_1|-1}$ are uncountable cardinal number. We can't use (A.17) to conclude the upper bound of $||Y||^2$ directly. Thus we use the method based on Nets to control $||Y||^2$. Let S_ε be a subset of $S^{|J_1|-1}$. For any $x \in S^{|J_1|-1}$, there exists $\tilde{x} \in S_\varepsilon$ such that $||\tilde{x} - x|| \leq \varepsilon$. Then if $||Yx^\top|| = ||Y||$, there exists $\tilde{x} \in S_\varepsilon$ such that

$$||Y\tilde{x}^\top|| \geq ||Yx^\top|| - ||Y(\tilde{x} - x)^\top|| \geq ||Y|| - \varepsilon||Y|| = (1-\varepsilon)||Y||.$$

Let $\varepsilon = 1/2$,

$$||Y||^2 \leq 4\max_{\tilde{x}\in S_{1/2}} ||Y\tilde{x}^\top||^2.$$

This, together with (A.17) and $|S_\varepsilon| \leq (1+2\varepsilon^{-1})^{|J_1|}$, implies that

$$P\Big(||Y||^2 > 4C_{A,3}v(|J_1| + |J_2|)\Big|B_Z\Big) \leq c5^{|J_1|}\exp(-5|J_1|v - 5|J_2|v). \tag{A.18}$$

Recalling (A.15), one can conclude that for any $c > 0$, there exists $C_{A,4}$ only depending on A and c such that

$$\begin{aligned}
&P\Big(\|n^{-2}Z_{J_1}\tilde{V}_hZ_{J_1}^\top Z_{J_1}\tilde{V}_h^\top Z_{J_2}^\top\|^2 > 4C_{A,4}vn^{-1}(|J_1|+|J_2|)\Big|B_Z\Big)\\
&\leq c5^{|J_1|}\exp(-5|J_1|v-5|J_2|v). \qquad \text{(A.19)}
\end{aligned}$$

Others term in (A.14) can be controlled by the same method. This completes the proof. □

Lemma A.1.6. *Under Assumptions 2.3.1-2.3.3 and $p = o(n)$,*

$$\|\hat{N}_{J_i} - \Lambda_i\| = O_p(n^{-1/2}q_i^{1/2}), \qquad \text{(A.20)}$$

where $J_i = \{j \in \mathscr{Z} : p_{i-1} < j \leq p_i\}$, $\Lambda_i = \operatorname{diag}(\lambda_{p_{i-1}+1}, \cdots, \lambda_{p_i})$, and λ_i are specified in Assumption 2.3.3.

Proof. We divide $\hat{N}_{J_i}$ into two terms: (i) the diagonal term $\hat{N}_{J_i,d}$ and (ii) the off-diagonal term $\hat{N}_{J_i,o}$. Lemma A.1.4 ensures $\|\hat{N}_{J_i,d} - \Lambda_i\| = O_p(n^{-1/2}q_i^{1/2})$. Thus we only need to show $\|\hat{N}_{J_i,o}\| = O_p(n^{-1/2}q_i^{1/2})$. If q_i is finite, Lemma A.1.4 can also ensure it. So we only need to consider the case q_i tends to infinity.

We can rewrite $\hat{N}_{J_i,o}$ and control $\|\hat{N}_{J_i,o}\|$ with the following idea.

$$\hat{N}_{J_i,o} = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix} = \begin{pmatrix} V_{11} & 0 \\ 0 & V_{22} \end{pmatrix} + \begin{pmatrix} 0 & V_{12} \\ V_{21} & 0 \end{pmatrix} = D_1 + V_{o,1}.$$

Each block is a $q_i/2 \times q_i/2$ matrix. Note that $V_{12} = V_{21}^\top$ and the norm of the second term $V_{o,1}$ (off-diagonal block) can be controlled by $\|V_{12}\|$. Moreover, we can control $\|V_{12}\|$ by Lemmas A.1.3 and A.1.5. In details, Lemma A.1.5 implies that

$$P\Big(\|V_{o,1}\|^2 > C_2vn^{-1}q_i\Big|B_Z\Big) \leq c(5^{q_i/2} + 5^{q_i/2})\exp(-5q_iv). \qquad \text{(A.21)}$$

For the first term, we can repeat the step on V_{11} and V_{22} to get a new matrix with off-diagonal blocks as follows:

$$V_{o,2} = diag\Big[\begin{pmatrix} 0 & V_{11,12} \\ V_{11,21} & 0 \end{pmatrix}, \begin{pmatrix} 0 & V_{22,12} \\ V_{22,21} & 0 \end{pmatrix}\Big].$$

Lemma A.1.5 implies that

$$P\Big(\|V_{o,2}\|^2 > C_2 v n^{-1} q_i/2 \Big| B_Z\Big) \leq 2c(5^{q_i/4} + 5^{q_i/4})\exp(-5q_i v/2). \tag{A.22}$$

Repeat the steps, we can find that $V_{o,j}$ has 2^{j-1} diagonal blocks and each diagonal block has two $2^{-j}q_i \times 2^{-j}q_i$ off-diagonal blocks. Lemma A.1.5 implies that

$$P\Big(\|V_{o,j}\|^2 > 2^{1-j} C_2 v n^{-1} q_i \Big| B_Z\Big) \leq 2^{j-1} c(5^{2^{-j}q_i} + 5^{2^{-j}q_i})\exp(-5q_i v \times 2^{1-j}). \tag{A.23}$$

We divide it into j_0 matrices: $\hat{N}_{J_i,o} = \sum_{j=1}^{j_0} V_{o,j}$, $2^{j_0-1} \leq q_i$ and $j_0 = O(\log q_i)$. For different j, we choose different v to control (A.23). When $\log q_i = o(2^{1-j}q_i)$, we choose $v = 1$. It follows that

$$P\Big(\|V_{o,j}\|^2 > 2^{1-j} C_2 n^{-1} q_i \Big| B_Z\Big) \leq 2^{j-1} c(5^{2^{-j}q_i} + 5^{2^{-j}q_i})\exp(-5q_i \times 2^{1-j}) = o(\log^{-1} q_i). \tag{A.24}$$

Otherwise, we choose $v = q_i^{4/5}\log^{-1} q_i$. It follows that

$$\begin{aligned} & P\Big(\|V_{o,j}\|^2 > C_2 n^{-1} q_i \log^{-2} q_i \Big| B_Z\Big) \qquad \text{(A.25)} \\ \leq \quad & P\Big(\|V_{o,j}\|^2 > 2^{1-j} C_2 q_i^{4/5} n^{-1} q_i \log^{-1} q_i \Big| B_Z\Big) \\ \leq \quad & 2^{j-1} c(5^{2^{-j}q_i} + 5^{2^{-j}q_i})\exp(-5q_i^{9/5}\log^{-1} q_i \times 2^{1-j}) = o(\log^{-1} q_i). \end{aligned}$$

(A.24)-(A.25) and $\|\hat{N}_{J_i,o}\| \leq \sum_{j=1}^{j_0} \|V_{o,j}\|$ imply that

$$P\Big(\|\hat{N}_{J_i,o}\| > 5C_2^{1/2} n^{-1/2} q_i^{1/2} \Big| B_Z\Big) = o(1). \tag{A.26}$$

Lemma A.1.3 implies that $\lim_{n\to\infty} P(B_Z) = 1$. This, together with (A.26) and $\|\hat{N}_{J_i,d} - \Lambda_i\| = O_p(n^{-1/2}q_i^{1/2})$, completes the proof.

□

Lemma A.1.7. *Under Assumptions 2.3.1-2.3.2 and $p = o(n)$,*

$$\|\Omega^\top \hat{\Sigma}^{-1}\Omega - I_p\| = O_p(n^{-1/2}p^{1/2}). \tag{A.27}$$

Proof. Since $\tilde{X}(s_j) = \Omega\tilde{Z}(s_j)$,

$$\begin{aligned}\Omega^\top\hat{\Sigma}^{-1}\Omega - I_p = &\quad \Omega^\top[n^{-1}\sum_{1\le j\le n}\tilde{X}(s_j)\tilde{X}(s_j)^\top]^{-1}\Omega - I_p\\ = &\quad [n^{-1}\sum_{1\le j\le n}\tilde{Z}(s_j)\tilde{Z}(s_j)^\top]^{-1} - I_p.\end{aligned}$$

It suffices to prove

$$\|n^{-1}\sum_{1\le j\le n}\tilde{Z}(s_j)\tilde{Z}(s_j)^\top - I_p\| = O_p(n^{-1/2}p^{1/2}).$$

Following the proof of Lemma A.1.6, one can verify the above equation. □

A.2 Proofs of Theorems

Recalling (A.11), write $\widehat{N} = \widehat{\Gamma}\widehat{\lambda}\widehat{\Gamma}^\top$ as its spectral decomposition, i.e.

$$\widehat{\lambda} = \mathrm{diag}(\widehat{\lambda}_1, \cdots, \widehat{\lambda}_p),$$

where $\widehat{\lambda}_1 \ge \cdots \ge \widehat{\lambda}_p \ge 0$ are the eigenvalues of $\widehat{N}$, and the columns of the orthogonal matrix $\widehat{\Gamma}$ are the corresponding eigenvectors. Recalling the definition of $\widehat{w}$ in (2.9)-(2.10), we can find that

$$\begin{aligned}
\widehat{w} = \quad & \frac{1}{k}\sum_{h=1}^{k}\hat{M}(f_h)\hat{M}(f_h)^\top \\
= \quad & \frac{1}{k}\sum_{h=1}^{k}\{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i-s_j)\widehat{\Sigma}^{-1/2}\tilde{X}(s_i)\tilde{X}(s_j)^\top\widehat{\Sigma}^{-1/2}\} \\
& \{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i-s_j)\widehat{\Sigma}^{-1/2}\tilde{X}(s_i)\tilde{X}(s_j)^\top\widehat{\Sigma}^{-1/2}\}^\top \\
= \quad & \frac{1}{k}\widehat{\Sigma}^{-1/2}\Omega\sum_{h=1}^{k}\{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i-s_j)\tilde{Z}(s_i)\tilde{Z}(s_j)^\top\}\Omega^\top\widehat{\Sigma}^{-1}\Omega \\
& \{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i-s_j)\tilde{Z}(s_i)\tilde{Z}(s_j)^\top\}^\top\Omega^\top\widehat{\Sigma}^{-1/2} \\
= \quad & \frac{1}{k}\widehat{\Sigma}^{-1/2}\Omega\sum_{h=1}^{k}\{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i-s_j)\tilde{Z}(s_i)\tilde{Z}(s_j)^\top\}(\Omega^\top\widehat{\Sigma}^{-1}\Omega-I_p) \\
& \{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i-s_j)\tilde{Z}(s_i)\tilde{Z}(s_j)^\top\}^\top\Omega^\top\widehat{\Sigma}^{-1/2}+\widehat{\Sigma}^{-1/2}\Omega\widehat{N}\Omega^\top\widehat{\Sigma}^{-1/2}.
\end{aligned}$$

Let $\hat{\Sigma}^{-1/2}\Omega=\widehat{V}_\Omega\widehat{\lambda}_\Omega\widehat{U}_\Omega$ where $\widehat{V}_\Omega\widehat{V}_\Omega^\top=\widehat{U}_\Omega\widehat{U}_\Omega^\top=I_p$ and $\widehat{\lambda}_\Omega$ is a diagonal matrix. Then

$$\begin{aligned}
\widehat{w} = \quad & \widehat{V}_\Omega\widehat{U}_\Omega\widehat{\Gamma}\widehat{\lambda}\widehat{\Gamma}^\top\widehat{U}_\Omega^\top\widehat{V}_\Omega^\top+\frac{1}{k}\widehat{\Sigma}^{-1/2}\Omega\sum_{h=1}^{k}\{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i-s_j)\tilde{Z}(s_i)\tilde{Z}(s_j)^\top\} \\
& \widehat{U}_\Omega^\top(\widehat{\lambda}_\Omega^2-I_p)\widehat{U}_\Omega\{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i-s_j)\tilde{Z}(s_i)\tilde{Z}(s_j)^\top\}^\top\Omega^\top\widehat{\Sigma}^{-1/2} \\
& +\widehat{V}_\Omega(\widehat{\lambda}_\Omega-I_p)\widehat{U}_\Omega\widehat{\Gamma}\widehat{\lambda}\widehat{\Gamma}^\top\widehat{U}_\Omega^\top\widehat{V}_\Omega^\top+\widehat{V}_\Omega\widehat{\lambda}_\Omega\widehat{U}_\Omega\widehat{\Gamma}\widehat{\lambda}\widehat{\Gamma}^\top\widehat{U}_\Omega^\top(\widehat{\lambda}_\Omega-I_p)\widehat{V}_\Omega^\top.
\end{aligned}$$

It follows that

$$\begin{aligned}
\widehat{U}_\Omega^\top\widehat{V}_\Omega^\top\widehat{w}\widehat{V}_\Omega\widehat{U}_\Omega = \quad & \widehat{\Gamma}\widehat{\lambda}\widehat{\Gamma}^\top+\frac{1}{k}\widehat{U}_\Omega^\top\widehat{\lambda}_\Omega\widehat{U}_\Omega\sum_{h=1}^{k}\{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i-s_j)\tilde{Z}(s_i)\tilde{Z}(s_j)^\top\} \\
& \widehat{U}_\Omega^\top(\widehat{\lambda}_\Omega^2-I_p)\widehat{U}_\Omega\{\frac{1}{n}\sum_{i,j=1}^{n} f_h(s_i-s_j)\tilde{Z}(s_i)\tilde{Z}(s_j)^\top\}^\top\widehat{U}_\Omega^\top\widehat{\lambda}_\Omega\widehat{U}_\Omega \\
& +\widehat{U}_\Omega^\top(\widehat{\lambda}_\Omega-I_p)\widehat{U}_\Omega\widehat{\Gamma}\widehat{\lambda}\widehat{\Gamma}^\top+\widehat{U}_\Omega^\top\widehat{\lambda}_\Omega\widehat{U}_\Omega\widehat{\Gamma}\widehat{\lambda}\widehat{\Gamma}^\top\widehat{U}_\Omega^\top(\widehat{\lambda}_\Omega-I_p)\widehat{U}_\Omega.
\end{aligned}$$

Then

$$\|\widehat{U}_{\Omega}^{\top}\widehat{V}_{\Omega}^{\top}\widehat{w}\widehat{V}_{\Omega}\widehat{U}_{\Omega}-\widehat{\Gamma}\widehat{\lambda}\widehat{\Gamma}^{\top}\|=O\{\|\widehat{\lambda}_{\Omega}-I_p\|\|\widehat{\lambda}\|(1+\|\widehat{\lambda}_{\Omega}\|)^3\}. \tag{A.28}$$

(A.27) implies that $\|\widehat{\lambda}_{\Omega}-I_p\|=O_p(n^{-1/2}p^{1/2})$ and $\|\widehat{\lambda}_{\Omega}\|=O_p(1)$.

Recalling $\hat{\Sigma}^{-1/2}\Omega=\widehat{V}_{\Omega}\widehat{\lambda}_{\Omega}\widehat{U}_{\Omega}$,

$$\|\widehat{U}_{W}^{\top}\widehat{\Sigma}^{-1/2}\Omega-\widehat{U}_{W}^{\top}\widehat{V}_{\Omega}\widehat{U}_{\Omega}\|\leq\|\widehat{U}_{W}^{\top}\widehat{V}_{\Omega}^{\top}(\widehat{\lambda}_{\Omega}-I_p)U_{\Omega}\|=O_p(n^{-1/2}p^{1/2}). \tag{A.29}$$

(A.29) implies that the leading term of $\widehat{\Gamma}_{\Omega}=\widehat{U}_{W}^{\top}\widehat{\Sigma}^{-1/2}\Omega$ is $\widehat{U}_{W}^{\top}\widehat{V}_{\Omega}\widehat{U}_{\Omega}$. (A.28) implies that $\widehat{U}_{W}^{\top}\widehat{\Sigma}^{-1/2}\Omega$ is close to $\widehat{\Gamma}^{\top}$.

Thus, the asymptotic properties of $\widehat{\Gamma}^{\top}$ is the key point. We will prove the following theorem for $\widehat{\Gamma}$ and $\widehat{\lambda}$.

Put $q_i=p_i-p_{i-1}$ for $i=1,\cdots,m$ (see Assumption 2.3.3), and

$$\hat{\Gamma}=\begin{pmatrix}\hat{\Gamma}_{11} & \cdots & \hat{\Gamma}_{1m}\\ \cdots & \cdots & \cdots\\ \hat{\Gamma}_{m1} & \cdots & \hat{\Gamma}_{mm}\end{pmatrix}, \qquad \hat{\Lambda}=\mathrm{diag}(\hat{\Lambda}_1,\cdots,\hat{\Lambda}_m), \tag{A.30}$$

where submatrix $\widehat{\Gamma}_{ij}$ is of the size $q_i\times q_j$, and $\hat{\Lambda}_i$ is a $q_i\times q_i$ diagonal matrix.

Theorem A.2.1. *Let Assumptions 2.3.1-2.3.1 hold. As $n\to\infty$ and $p=o(n)$, it holds that*

$$\|\hat{\Gamma}_{ij}\|=O_p\{n^{-1/2}(q_i+q_j)^{1/2}+n^{-1}p\}, \quad 1\leq i\neq j\leq m, \quad \text{and} \tag{A.31}$$

$$\|\hat{\Lambda}_i-\Lambda_i\|=O_p(n^{-1/2}q_i^{1/2}+n^{-1}p), \quad 1\leq i\leq m, \tag{A.32}$$

where $\Lambda_i=\mathrm{diag}(\lambda_{p_{i-1}+1},\cdots,\lambda_{p_i})$, and λ_i are specified in Assumption2.3.3.

(A.28), (A.29), (A.27) and Theorem A.2.1 can conclude Theorem 2.3.1. Thus, we now need to prove Theorem A.2.1.

Proof of Theorem A.2.1. (2.15) and (A.2) show that m is bounded. Let $J_i=\{j\in\mathscr{X}:p_{i-1}<j\leq p_i\}$. At first we prove (A.32). We only need to prove it when $i=1$ and other cases can be

concluded by a permutation. Define J_1^c be the complementary set of J_1, then we can rewrite $\det(\lambda I_p - \hat{N}) = 0$ as follows.

$$0 = \det(\lambda I_p - \hat{N}) = \det \begin{pmatrix} \lambda I_{p_1} - \hat{N}_{J_1} & -\hat{N}_{J_1,J_1^c} \\ -\hat{N}_{J_1^c,J_1} & \lambda I_{p-p_1} - \hat{N}_{J_1^c} \end{pmatrix}. \tag{A.33}$$

Lemmas A.1.3 and A.1.5 conclude $\|\hat{N}_{J_1^c,J_1}\| = O_p(n^{-1/2}p^{1/2}) = o_p(1)$. Lemmas A.1.3-A.1.6 and the Assumption 2.3.3 imply that there exists a positive constant $\tilde{C}_N$ such that

$$\lim_{n\to\infty} P(\|\lambda_l I_{p-p_1} - \hat{N}_{J_1^c}\|_{min} > \tilde{C}_N) = 1 \tag{A.34}$$

for any $1 \le l \le p_1$. Lemma A.1.6 also implies that

$$\lim_{n\to\infty} P\Big(\lambda_{p_1} - \tilde{C}_N/2 < \|\hat{N}_{J_1}\|_{min} \le \|\hat{N}_{J_1}\| < \lambda_1 + \tilde{C}_N/2\Big) = 1. \tag{A.35}$$

If $\lambda \in (\lambda_{p_1} - \tilde{C}_N/2, \lambda_1 + \tilde{C}_N/2)$ is a solution of (A.33), it is also (with probability 1) a solution of

$$0 = \det\Big(\lambda I_{p_1} - \hat{N}_{J_1} - \hat{N}_{J_1,J_1^c}(\lambda I_{p-p_1} - \hat{N}_{J_1^c})^{-1}\hat{N}_{J_1^c,J_1}\Big). \tag{A.36}$$

Lemma A.1.5 and (A.34) imply that

$$\|\hat{N}_{J_1,J_1^c}(\lambda I_{p-p_1} - \hat{N}_{J_1^c})^{-1}\hat{N}_{J_1^c,J_1}\| = O_p(n^{-1}p). \tag{A.37}$$

Let $\tilde{\lambda}_1 \ge \cdots \ge \tilde{\lambda}_{p_1}$ be the eigenvalues of $\hat{N}_{J_1}$, (A.36)-(A.37) conclude that

$$\tilde{\lambda}_l - \hat{\lambda}_l = O_p(n^{-1}p) \tag{A.38}$$

for any $1 \le l \le p_1$. This, together with (A.20), concludes (A.32).

Now we consider (A.31). We only need to prove it when $j = 1$ and $i > 1$. Other cases can be concluded by a permutation. From $\widehat{N} = \widehat{\Gamma}\widehat{\lambda}\widehat{\Gamma}^\top$ and (A.30), we can find that

$$\begin{pmatrix} \sum_{i=1}^m \hat{N}_{J_1,J_i}\hat{\Gamma}_{i1} \\ \cdots \\ \sum_{i=1}^m \hat{N}_{J_m,J_i}\hat{\Gamma}_{i1} \end{pmatrix} = \hat{N} \begin{pmatrix} \hat{\Gamma}_{11} \\ \cdots \\ \hat{\Gamma}_{m1} \end{pmatrix} = \begin{pmatrix} \hat{\Gamma}_{11}\hat{\Lambda}_1 \\ \cdots \\ \hat{\Gamma}_{m1}\hat{\Lambda}_1 \end{pmatrix}. \tag{A.39}$$

Define $U_{11} = \hat{N}_{J_1,J_1}$, $U_{12} = \hat{N}_{J_1,J_1^c}$, $U_{21} = \hat{N}_{J_1^c,J_1}$ and $U_{22} = \hat{N}_{J_1^c,J_1^c}$. Similarly, define $\tilde{\Gamma}_{21}^\top = (\hat{\Gamma}_{21}^\top, \cdots, \hat{\Gamma}_{m1}^\top)^\top$. Then we can rewrite (A.39) as

$$\begin{pmatrix} U_{11}\hat{\Gamma}_{11} + U_{12}\tilde{\Gamma}_{21} \\ U_{21}\hat{\Gamma}_{11} + U_{22}\tilde{\Gamma}_{21} \end{pmatrix} = \begin{pmatrix} \hat{\Gamma}_{11}\hat{\Lambda}_1 \\ \tilde{\Gamma}_{21}\hat{\Lambda}_1 \end{pmatrix}. \tag{A.40}$$

$$\tilde{\Gamma}_{21}\hat{\Lambda}_1 = \tilde{\Gamma}_{21}(\hat{\Lambda}_1 - \lambda_1 I_{p_1}) + \lambda_1\tilde{\Gamma}_{21}.$$

Then the second line of (A.40) is equivalent to

$$(U_{22} - \lambda_1 I_{p-p_1})\tilde{\Gamma}_{21} = \tilde{\Gamma}_{21}(\hat{\Lambda}_1 - \lambda_1 I_{p_1}) - U_{21}\hat{\Gamma}_{11}.$$

Recal that (A.34), $U_{22} - \lambda_1 I_{p-p_1}$ is invertible with probability 1 as n tends to infinity.

$$\tilde{\Gamma}_{21} = (U_{22} - \lambda_1 I_{p-p_1})^{-1}\tilde{\Gamma}_{21}(\hat{\Lambda}_1 - \lambda_1 I_{p_1}) - (U_{22} - \lambda_1 I_{p-p_1})^{-1}U_{21}\hat{\Gamma}_{11}.$$

(2.14)-(2.15) and Lemmas A.1.3-A.1.6 imply that $\|\hat{\Lambda}_1 - \lambda_1 I_{p_1}\| = o_p(1)$ and $\|(U_{22} - \lambda_1 I_{p-p_1})^{-1}\| = O_p(1)$. Then $(\lambda_1 I_{p-p_1} - U_{22})^{-1}U_{21}\hat{\Gamma}_{11}$ is the leading term of $\tilde{\Gamma}_{21}$. Moreover, $\|\hat{\Gamma}_{11}\| = O(1)$. Thus we only need to consider $(\lambda_1 I_{p-p_1} - U_{22})^{-1}U_{21}$. We rewrite $(\lambda_1 I_{p-p_1} - U_{22})^{-1}$ as

$$\begin{pmatrix} \lambda_1 I_{p_2} - \hat{N}_{J_2,J_2} & \cdots & -\hat{N}_{J_2,J_m} \\ \cdots & \cdots & \cdots \\ -\hat{N}_{J_m,J_2} & \cdots & \lambda_1 I_{p_m} - \hat{N}_{J_m,J_m} \end{pmatrix}^{-1} = (\lambda_1 I_{p-p_1} - U_{22})^{-1} = \begin{pmatrix} V_{22} & \cdots & V_{2m} \\ \cdots & \cdots & \cdots \\ V_{m2} & \cdots & V_{mm} \end{pmatrix}.$$

(2.14)-(2.15) and Lemma A.1.6 ensure $\|(\lambda_1 I_{p_i} - \hat{N}_{J_i,J_i})^{-1}\| = O_p(1)$ for $2 \le i \le m$. Lemma A.1.5 ensures $\|\hat{N}_{J_i,J_t}\| = O_p(n^{-1/2}p^{1/2}) = o_p(1)$ for $2 \le i \ne t \le m$. Since m is finite, we can find $\|V_{ii}\| = O_p(1)$ and $\|V_{it}\| = O_p(n^{-1/2}p^{1/2})$ for $2 \le i \ne t \le m$. Recall that $\|\hat{N}_{J_i,J_1}\| = O_p(n^{-1/2}(q_1+q_i)^{1/2})$ for $2 \le i \le m$ and

$$(\lambda_1 I_{p-p_1} - U_{22})^{-1}U_{21} = \begin{pmatrix} V_{22} & \cdots & V_{2m} \\ \cdots & \cdots & \cdots \\ V_{m2} & \cdots & V_{mm} \end{pmatrix} \begin{pmatrix} \hat{N}_{J_2,J_1} \\ \cdots \\ \hat{N}_{J_m,J_1} \end{pmatrix}.$$

It follows that $\|V_{ii}\hat{N}_{J_i,J_1}\| = O_p(n^{-1/2}(q_1+q_i)^{1/2})$ and $\|\sum_{t\ne i} V_{it}\hat{N}_{J_t,J_1}\| = O_p(n^{-1}p)$.

We complete the proof of (A.31).

□

Now we prove Theorem 2.3.2. By the same idea, we give the following result for $\widehat{N}$.

Theorem A.2.2. *Let Assumptions 2.3.1, 2.3.2 and 2.3.4 hold. Denote by $\hat{\gamma}_{ij}$ the (i,j)-th entry of matrix $\widehat{\Gamma}$ in (A.30). Then as $n,p \to \infty$, it holds that*

$$\widehat{\Gamma}_{ij} = O_p(n^{-1/2}v_{\text{gap}}^{-1}|j-i|^{-1}) \quad \text{for } 1 \le i \ne j \le p, \quad \text{and} \tag{A.41}$$

$$\widehat{\Gamma}_{ii} = 1 + O_p(n^{-1}v_{\text{gap}}^{-2}) \quad \text{for } i = 1,\cdots,p. \tag{A.42}$$

Moreover,

$$\|\hat{\Lambda} - \Lambda\| = O_p(n^{-1/2}p^{1/2}). \tag{A.43}$$

Proof of Theorem A.2.2. Following the proof of Lemma A.1.6, one can verify that $\|\widehat{\lambda} - N\| = O_p(n^{-1/2}p^{1/2})$. This, together with A4, implies (A.43).

From $\widehat{N}\widehat{\Gamma} = \widehat{\Gamma}\widehat{\lambda}$, we can find that

$$\widehat{\Gamma}\widehat{\lambda} - N\widehat{\Gamma} = (\widehat{N} - N)\widehat{\Gamma}. \tag{A.44}$$

(A.44) implies that

$$\widehat{\Gamma}_{ij}(\widehat{\lambda}_j - \lambda_i) = \sum_{s=1}^{p} M_{is}\widehat{\Gamma}_{sj}, \tag{A.45}$$

where M_{is} is defined in Lemma A.1.4. The Assumption 2.3.4 and $\|\widehat{\lambda} - N\| = O_p(n^{-1/2}p^{1/2})$ can control $(\widehat{\lambda}_j - \lambda_i)$. Then we can divide the right hand of the above equation into two part.

$$\sum_{s=1}^{p} M_{is}\widehat{\Gamma}_{sj} = \sum_{s\neq j} M_{is}\widehat{\Gamma}_{sj} + M_{ij}\widehat{\Gamma}_{jj}. \tag{A.46}$$

(A.12) implies that $E|M_{ij}\widehat{\Gamma}_{jj}|^2 \leq E|M_{ij}|^2 \leq C_1 n^{-1}$. Thus we only need to consider the order of $\sum_{s\neq j} M_{is}\widehat{\Gamma}_{sj}$. Define $v = \max_{1\leq i\leq p}\max_{j\neq i}|\sum_{s\neq j} M_{is}\widehat{\Gamma}_{sj}|$. Then for any $j \neq i$, (A.45) implies that

$$|\widehat{\Gamma}_{ij}| \leq (|i-j|v_{gap} - \|\widehat{\lambda} - N\|)^{-1}(v + |M_{ij}|)$$

and

$$\begin{aligned} & |\sum_{s\neq j} M_{is}\widehat{\Gamma}_{sj}| \leq \sum_{s\neq j} |M_{is}||\widehat{\Gamma}_{sj}| \\ \leq \quad & \sum_{s\neq j} |M_{is}|(|s-j|v_{gap} - \|\widehat{\lambda} - N\|)^{-1}(v + |M_{sj}|) \\ \leq \quad & v\sum_{s\neq j} |M_{is}|(|s-j|v_{gap} - \|\widehat{\lambda} - N\|)^{-1} + \sum_{s\neq j} |M_{is}||M_{sj}|(|s-j|v_{gap} - \|\widehat{\lambda} - N\|)^{-1}. \end{aligned}$$

The Assumption 2.3.4, $\|\widehat{\lambda} - N\| = O_p(n^{-1/2}p^{1/2})$ and (A.12) conclude that

$$\sum_{s\neq j} |M_{is}|(|s-j|v_{gap} - \|\widehat{\lambda} - N\|)^{-1} = O(v_{gap}^{-1}\log p \max_{1\leq i,s\leq p}|M_{is}|) = o_p(1)$$

and

$$\sum_{s\neq j} |M_{is}||M_{sj}|(|s-j|v_{gap} - \|\widehat{\lambda} - N\|)^{-1} = o_p(n^{-1/2}).$$

This, together with the definition of v, implies that $v = o_p(n^{-1/2})$.

$$|\widehat{\Gamma}_{ij}| \leq (|i-j|v_{gap} - \|\widehat{\lambda} - N\|)^{-1}[o_p(n^{-1/2}) + |M_{ij}|].$$

This, together with (A.12), concludes (A.41).

$$\widehat{\Gamma}_{ii}^2 = 1 - \sum_{j \neq i} \widehat{\Gamma}_{ij}^2 \geq 1 - \sum_{j \neq i} (|i-j|v_{gap} - \|\widehat{\lambda} - N\|)^{-2}(v + |M_{ij}|)^2 = 1 + O_p(n^{-1}v_{gap}^{-2}).$$

We complete the proof. □

(A.28) and (A.27) imply that

$$\|\widehat{\Gamma}^\top \widehat{U}_\Omega^\top \widehat{V}_\Omega^\top \widehat{w} \widehat{V}_\Omega \widehat{U}_\Omega \widehat{\Gamma} - \widehat{\lambda}\| = O_p(n^{-1/2}p^{1/2}).$$

This and Theorem A.2.2 can conclude the asymptotic properties of $\widehat{U}_W^\top \widehat{V}_\Omega \widehat{U}_\Omega \widehat{\Gamma}$. Then we can prove Theorem 2.3.2 by (A.29) and Theorem A.2.2.

A.3 An Additional Example for Numerical Results

In this section, we further present the usefulness of Multiple Ring Kernels by constructing a special example. In this example, Ring Kernel 1 is no longer the best single kernel. We achieve this goal by generating latent fields in a mixing way. To generate data, we split the map of sample locations into 10 rows according to their y coordinates, and all rows have equal width. For each row, let the sample points within be independent from adjacent rows. In order to achieve this, for each of the p latent fields, we generate 3 independent candidate random fields using same set of coordinates and covariance function parameters. The process for generating each candidate random field is the same as described before. The coordinates belong to the $1^{st}, 4^{th}, 7^{th}$ and 10^{th} row would take values from the first candidate random field, those belong to the $2^{nd}, 5^{th}, 8^{th}$ row would take values from the second candidate random field, and the rest of the sample points will take values from the third candidate random field. In this way, the samples from most adjacent rows are independent to each other, and the effectiveness of Ring Kernel 1 is weakened.

We performed simulation using latent random fields constructed from the method above. Dimension of latent field $p = 3$. The sample size, sampling method of coordinates, setting of mixing matrix, and use of matern covariance function is identical to the description of simulation setting in numerical illustration section. The boxplot of $D(\Omega, \hat{\Omega})$ obtained from 1000 replications is presented in figure, and median of $D(\Omega, \hat{\Omega})$ is presented in table.

As the Figure shows, kernel 1 is no longer the best-performing single kernel, while multiple kernel remains very close to the best single kernel, and outperforming most other single kernels. Yet as sample size increases, $D(\Omega, \hat{\Omega})$ did not improve, which might due to the artificial nature of this special example. More detailed data is presented in Table A.1.

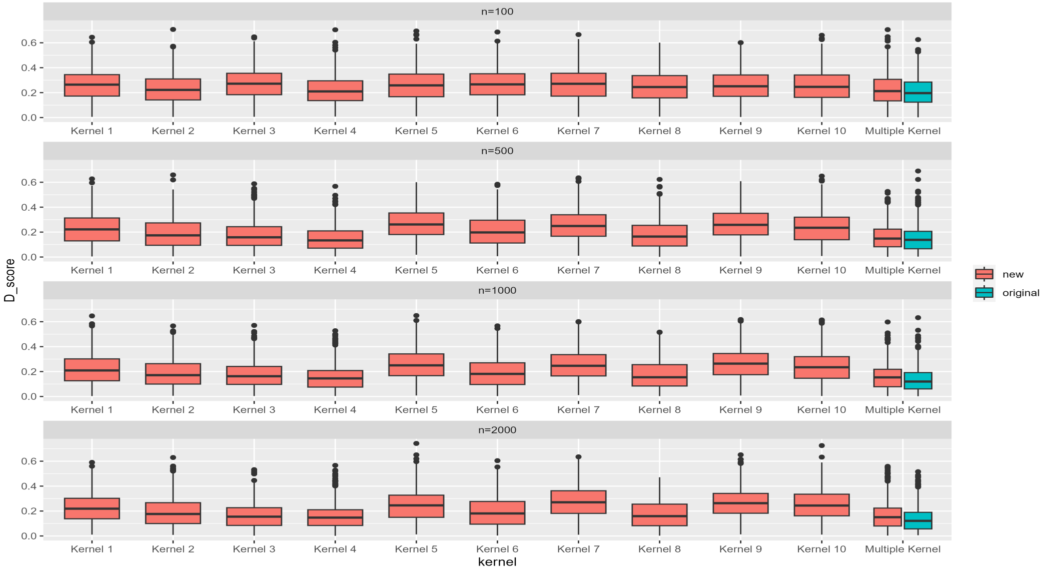

Fig. A.1 Boxplots of $D(\Omega, \hat{\Omega})$ for the proposed method using the 10 single kernels, or multiple kernel(including all 10 ring kernels), and the method of Bachol et el. using the multiple kernel (original) in a simulation with 1000 replications for the mix Gaussian random fields. The number of observations n is 100, 500, 1000 or 2000 (from top to bottom), and the dimension of random fields is $p = 3$. Latent field is generated in a mixing approach as described.

Kernel	1	2	3	4	5	6	7	8	9	10	Multiple	Multiple Original
n=100	0.2642	0.2218	0.2718	0.2098	0.2583	0.2663	0.2710	0.2448	0.2510	0.2462	0.2123	0.2242
n=500	0.2218	0.1739	0.1583	0.1335	0.2614	0.1974	0.2489	0.1642	0.2582	0.2348	0.1480	0.1045
n=1000	0.2091	0.1712	0.1627	0.1452	0.2500	0.1813	0.2463	0.1544	0.2638	0.2346	0.1535	0.8800
n=2000	0.2190	0.1763	0.1548	0.1474	0.2455	0.1807	0.2703	0.1590	0.2631	0.2442	0.1506	0.0752

Table A.1 Median of $D(\Omega, \hat{\Omega})$ from the proposed method using the 10 single kernels, or multiple kernel(including all 10 ring kernels), and the method of Bachol et el. using the multiple kernel (original) in a simulation with 1000 replications for the mixed random fields. The number of observations n is 100, 500, 1000 or 2000 , and the dimension of random fields is $p = 3$.

Appendix B

List of Variables in Dataset for Chapter 4

B.1 Name of 12 Metropolitan Regions:

1. Auvergne-Rhône-Alpes
2. Bourgogne-Franche-Comté
3. Bretagne
4. Centre-Val de Loire
5. Grand Est
6. Hauts-de-France
7. Île-de-France
8. Normandie
9. Nouvelle-Aquitaine
10. Occitanie
11. Provence-Alpes-Côte d'Azur (PACA)
12. Pays de la Loire

B.2 List of National Level Variables:

Category	**Variables**
Calendar	date, Date, tod, month, year, toy, day_type_jf, day_type_ljf, day_type_vjf, day_type_week, day_type_week_jf, day_type_hc, period_hour_changed, period_holiday, period_holiday_zone_a, period_holiday_zone_b, period_holiday_zone_c, period_christmas, period_summer, day_type_week_period_hour_changed, day_type_week_jf_period_holiday, week_number
Electricity Load(MW)	Load, Load_d1, Load_d7, Wind_power, Solar_power, DayValidity
Meteorological	temperature, temperature_lisse_990, temperature_lisse_950, wind, nebulosity, wind_by_wind_power_weights.x, nebulosity_by_solar_power_weights.x

Table B.1 Classification of Variables by Category, National Level

B.3 List of Region-Specific Variables:

Category	**Variables for Each Region**
Electricity Load (MW)	$Load_{(region)}$,$Load_{(region)}\_d1$, $Load_{(region)}\_d7$
Meteorological	$temperature_{(region)}$, $temperature_{(region)}\_lisse\_990$, $temperature_{(region)}\_lisse\_950$, $Wind\_power_{(region)}$, $wind_{(region)}$, $Solar\_power_{(region)}$, $nebulosity_{(region)}$

Table B.2 Region-Specific Variables by Category