Methods for time series network analysis

Michael Hellstern

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2025

Reading Committee:
Ali Shojaie, Chair
Zaid Harchaoui
Eardi Lila

Program Authorized to Offer Degree:
Biostatistics

University of Washington

Abstract

Methods for time series network analysis

Michael Hellstern

Chair of the Supervisory Committee:
Ali Shojaie
Department of Biostatistics

Statistical networks can encode arbitrary relationships between variables in a system. Due to this flexibility, scientific hypotheses about interactions between variables can typically be formulated as a statistical network analysis. In addition to analyzing static networks, studying how statistical networks change in response to experimental or environmental conditions is often of scientific interest. A network is typically defined as a set of vertices and edges. Specifically a network or graph, G, can be written as $G = (V, E)$ where $V = \{1, \ldots, k\}$ are the vertices or variables and E is the edge set that encodes the relationship between variables. A common example of a statistical network is the correlation matrix, where an edge represents the correlation between variables.

While analysis of networks and their changes are ubiquitous across many domains, our work is motivated specifically by applications in which networks are derived from time series data. In contrast to independent data, statistical analysis of time series data is complicated by the inherent serial correlation. In practice, the degree of this correlation is unknown and network analysis methods that can flexibly handle varying degrees of dependence are needed. We approach this problem from two angles. The first angle, used in the first two portions of this thesis, focuses on developing methods with minimal assumptions on temporal dependence. In the third portion of this thesis we approach this problem from the second angle which attempts to leverage the flexibility of deep learning methods to analyze statistical networks.

In the first chapter, we propose a novel order selection method in vector autoregressive (VAR)

models. Order selection is an essential step in fitting VAR models and while many order selection methods exist, all come with weaknesses. Our proposed order selection method is based on the observation that the expected squared error loss is flat once the fitted order reaches or exceeds the true order. We show that under mild assumptions on the underlying process our new order selection method consistently estimates the true order.

Motivated by applications in neuroscience, the second chapter of this thesis develops a novel estimation and inference procedure for a difference in the inverse spectral densities. In neuroscience, it is often of interest to study how brain networks change in response to electrical stimulation with the hopes of developing stimulation-based treatments for neurodegenerative diseases. Furthermore, it is essential to study networks in the frequency domain as higher frequencies contain key brain connectivity information. With this in mind, we develop methods to directly estimate and perform statistical inference on a difference in inverse spectral densities. Crucially, our method relies on minimal assumptions and can flexibly handle a large range of data dependence.

The last chapter of this thesis proposes a new deep learning-based change-point detection framework. The core idea behind this method is a continuous approximation of the indicator function. With this approximation, change-points can be specified as parameters of a deep learning model. Thus, change-points and model parameters can be jointly learned using stochastic optimization techniques. The proposed framework is general and can be applied to both independent and dependent data, such as time series data. Furthermore, the framework is model-agnostic and thus can be used to encode networks and study their changes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Above all, I would like to thank my thesis advisor, Ali Shojaie. For the better part of a decade - from the start of my master's degree to the culmination of my doctoral degree - thank you for igniting my intellectual curiosity and for your unwavering support and guidance. I will always be grateful for your mentorship and encouragement to pursue independent projects.

I would also like to thank the many others who have been part of my academic journey. The late Stephen Sheppard at Williams College guided me through my first steps into academic research and sparked my interest in pursuing a doctoral degree. To my collaborators and committee members, Byol Kim, Zaid Harchaoui, Eardi Lila, Daniela Witten, and Steve Mooney, thank you for your time and support.

To all the UW statistics and biostatistics students I've met along the way, thank you for the study groups and countless intellectual discussions.

Lastly, to my parents, siblings, friends, and other family, I can't express enough gratitude for your emotional support and encouragement throughout my entire educational journey. I would not have been able to do this without you.

DEDICATION

To my parents

Richard Hellstern and Kathleen Meier-Hellstern

Chapter 1

INTRODUCTION

Studying how variables interact as well as how these interactions change over time or in response to external conditions is of scientific interest across a wide range of disciplines from neuroscience to genomics, economics, and oceanography (Bloch et al., 2022; Shojaie and Michailidis, 2009; Chiou-Wei et al., 2008; Laurindo et al., 2019). While statistical network analysis is commonplace in applications involving both independent and dependent data, we focus on the analysis of dependent data, and more specifically time series data, due to the additional challenges that arise and the ubiquity of time series data. Notably, compared to independent data, analysis of time series data requires statistical tools that account for the inherent data dependence present. Within the time series domain, the breadth of areas in which statistical network analysis is used give rise to an equally broad set of data applications. In neuroscience, for example, electrical voltage on dozens of sub-millimeter sized portions of the brain are captured thousands of times per second (Yazdan-Shahmorad et al., 2018). Contrastingly, yearly GDP and energy consumption data is used in economics to study the relationship between the two (Chiou-Wei et al., 2008). This array of data applications in the time series network analysis domain necessitate tools that are designed to handle a wide variety of temporal dependence.

Due to the known temporal dependence in time series data, it is common to model data at time t as a function of its prior lags. When the data is multivariate and a linear relationship between the current time t and prior lags is used, this results in the well-known vector autoregression (VAR) model (Sims, 1980). However, a key component of VAR models is determining the number of prior lags to use, a problem known as order selection. Order selection in VAR models is critical as the chosen order can have direct impacts on scientific conclusions when using, for example,

VAR models for Granger causality analysis (Shojaie and Fox, 2022). Order selection has been studied dating back to the 1970s and 1980s and many of these methods are still popular today (Akaike, 1973, 1974; Hannan and Quinn, 1979; Schwarz, 1978). In Chapter 2, we propose a new order selection criteria, mean square information criteria (MIC), based on the observation that the population squared error loss is flat when the fitted order reaches or exceeds the true model order. We show MIC consistently estimates the true order under mild assumptions. In particular our results make no assumptions on the likelihood. We also connect our MIC procedure to existing procedures such as AIC (Akaike, 1973) and BIC (Schwarz, 1978) through each methods use of the residual error matrix. In particular, we show that all methods rely on an estimate of the residual error matrix and only differ in how they use and penalize this information.

In neuroscience, changes in the brain connectivity network can be induced by spike timing dependent plasticity (STDP) informed neural stimulation (Bloch et al., 2022; Bi and Poo, 1998). While network changes can be induced by stimulation, its effect on the entire brain network is not immediately clear. Analyses are further complicated by the fact that the underlying brain network may mediate the effect of stimulation (Bloch et al., 2022). However, understanding how brain networks change in response to stimulation may be a key breakthrough in the development of treatments for diseases such as schizophrenia and Alzheimer's disease (Garrity et al., 2007; Stam et al., 2007). The inverse spectral density is an appealing choice of network as edges encode the coherence, the frequency domain analog of correlation, between two nodes after removing the linear effects of all other nodes (Dahlhaus, 2000). In this way the inverse spectral density more closely resembles the direct relationship between two nodes.

Modern data collection techniques have allowed for datasets where the number of variables is much larger than the number of samples $p \gg n$. Classical asymptotic statistics assumes that the dimension p is fixed and studies statistical properties as $n \to \infty$. In a high-dimensional data regime, such tools are no longer valid and new statistical and computational tools are needed (Johnstone and Titterington, 2009). Popular computational tools for the analysis of high-dimensional data include the LASSO, for linear models, and its graphical equivalent, for sparse inverse covariance estimation (Tibshirani, 1996; Friedman et al., 2008). On the theoretical front, Negahban et al.

(2012) has developed a framework to establish the consistency and convergence rates of penalized high-dimensional M-estimators. This is expanded upon further in Neykov et al. (2018) where the authors develop a new statistical inference framework for high-dimensional estimating equations. While these high-dimensional statistical tools exist, they often assume the data is independent across observations and are not immediately applicable to the dependent case. Motivated by the differential network applications in neuroscience we develop a new method to directly estimate and perform inference on the difference in high-dimensional inverse spectral densities that allows for flexible levels of data dependence in Chapter 3.

In the final Chapter, Chapter 4, we develop a deep learning-based change-point detection framework motivated by temporally dynamic time series networks. Change-point detection aims to detect abrupt structural breaks in a data generating process with the first works dating back to the 1950s (Page, 1954). Since then, the change-point detection problem has been studied extensively. See Truong et al. (2020) for a recent review. Nonparametric approaches to the change-point detection problem such as those in Arlot et al. (2019) and Matteson and James (2014) aim to detect changes in the entire probability distribution. While general, such methods are often designed for independent data. With the advent and rapid rise of deep learning, it is natural to consider how such techniques can be adapted to the change-point detection problem with dependent data. Indeed, Chang et al. (2019) and De Ryck et al. (2021) have recently proposed such models for time series data. However, these deep learning-based change-point detection approaches are limited to a specific architecture which may not be suitable for all applications.

We formulate the change-point detection problem as a piecewise optimization problem where the prediction functions in each regime can have an arbitrary architecture. Moreover, while we only discuss detection of changes in the regression function via mean square error loss in Chapter 4, our method is applicable to any loss function. At its core, our method hinges on a continuous relaxation of the indicator function to make the change-points differentiable parameters. With this relaxation we can jointly optimize the loss function over the change-points and prediction functions. We conclude this thesis with a dicussion of avenues for future research in Chapter 5.

Chapter 2

ORDER SELECTION IN VECTOR AUTOREGRESSION BY MEAN SQUARE INFORMATION CRITERION

2.1 Introduction

In vector autoregressive (VAR) models, each variable is modeled as a linear function of the multivariate time series over prior lags. VARs were first introduced in macroeconometrics by Sims (1980) and have since become standard for macroeconomic forecasting. They have also become essential tools in a range of other fields. For instance, VARs have been used in biomedical applications; in neuroscience to analyze functional connectivity in the brain (Seth et al., 2015) and in epidemiology to predict COVID-19 cases (Kitaoka and Takahashi, 2023). A fundamental problem in fitting VAR models is how to choose the lag order. In forecasting analyses using too few lags may result in underfitting, while too many lags can lead to overfitting, both decreasing the accuracy of forecasts. Incorrect selection of the lag order can also impact the selection of the relevant variables in the VAR model, resulting in ambiguous interpretations. Unfortunately, the lag order is typically unknown and must be chosen either by prior knowledge or in a data-dependent way.

Perhaps the most popular VAR order selection method is to choose the order that minimizes an information theoretic criteria, commonly referred to as Akaike's Information Criterion (AIC). Model selection by minimizing AIC was proposed in Akaike (1973, 1974). Minimizing the AIC selects the model with the lowest negative log likelihood plus a penalty term on the number of independently adjustable parameters. Despite its popularity, AIC has several drawbacks for use in VAR models. The first stems from AIC's inherent reliance on the likelihood. In the context of VAR models, this often amounts to assuming a Gaussian likelihood, which results in simplifying the log likelihood term to the log determinant of the prediction error matrix. When the errors are not Gaussian, this simplification of the likelihood is no longer valid. The second is that AIC may

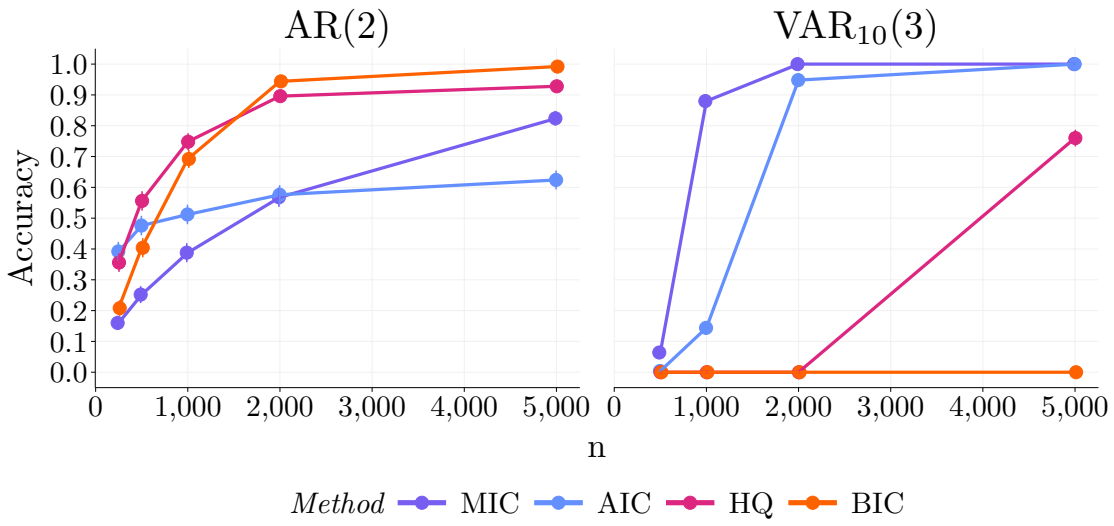


Figure 2.1: Accuracy of various order selection methods in detecting the order of simulated $\mathrm{AR}(2)$ (*left*) and $\mathrm{VAR}_{10}(3)$ (*right*) processes. Vertical lines represent standard errors. Accuracy is measured as the proportion of simulations where the correct order was chosen. See Section 2.5 for more details.

not provide a consistent estimate of the VAR order (Lütkepohl, 2005, Corollary 4.2.1). Although this limitation improves as the dimension of the process increases and is negligible for VAR models of dimension greater than 5 (Paulsen and Tjøstheim, 1985), it nonetheless limits the applicability of AIC. This lack of consistency is highlighted in the simulation results in the left panel of Figure 2.1. The plot shows that in the univariate case, AIC reaches its peak accuracy of 0.6 at around $n = 2000$ and does not improve substantially as n increases to $5,000$. These results are in contrast to those presented in the right panel of Figure 2.1 for a 10 dimensional VAR model: in this case, AIC has nearly perfect accuracy for $n \geq 2000$. Shortly after AIC was introduced, additional VAR order selection criteria were proposed by Schwarz (1978) (BIC) and Hannan and Quinn (1979), Quinn (1980) (Hannan-Quinn, HQ). Similar to AIC, the BIC criterion relies on the likelihood, which in the case of Gaussian errors amounts to the log determinant of the prediction error, plus a penalty on

the number of model parameters. The HQ criteria is not likelihood based but also relies on the log determinant of the prediction error plus a penalty. While all three criteria use the log determinant of the prediction error, they differ in the penalty used. If n is the sample size, AIC uses a penalty of $(2/n)(\#\text{parameters})$, while HQ and BIC use penalties of $((2\log\log n)/n)(\#\text{parameters})$ and $((\log n)/n)(\#\text{parameters})$, respectively. This change of penalty is essential: if the underlying process is a stationary stable VAR with standard white noise, it can be shown that both HQ and BIC consistently estimate the true order of the process (Lütkepohl, 2005, Corollary 4.2.2). However, while HQ and BIC are consistent, simulation results in Figure 2.1 show they perform poorly for small sample sizes when the dimension of the process increases to a moderate size.

To address the above limitations of existing order selection methods, we propose a new method, mean squared information criterion (MIC). MIC is likelihood-free, consistent, and performs well in a variety of simulation settings. Our criterion leverages the novel observation that the expected squared error loss is constant when the fitted VAR order is at least as large as the true model order. We establish the consistency of MIC under mild assumptions and show that, compared with AIC, HQ, and BIC, it performs well in a variety of simulation settings and forecasting on real data.

The rest of the chapter is structured as follows. In Section 2.2, we motivate our new information criterion and show that the expected squared error loss is constant when the fitted order is at least as large as the true model order. We extend this observation to the multivariate case and present theoretical results in Section 2.3, introduce our estimator in Section 2.4 and compare its performance to AIC, HQ and BIC using simulated data in Section 2.5. We apply our method to a financial application and COVID-19 forecasting in Section 2.6 and end with some concluding remarks and a discussion in Section 2.7.

Notation: We use uppercase letters X, Y, Z to denote random variables. We will use subscript t to denote the time component as in Z_t. When Z or Z_t are random vectors, the components can be accessed by Z_j, $Z_{t,j}$, respectively. Observed values of random variables are written in lowercase as in z_t. Additionally, observed vectors and matrices are denoted using bold lowercase and uppercase letters as in $\mathbf{z}$ and $\mathbf{Z}$, respectively. Hats will be used to specify sample estimates of population quantities, e.g. $\hat{\Gamma}_0$ represents the sample estimate of Γ_0.

2.2 A new idea

We begin with the univariate case. Let Z_t be a stationary univariate time series. Without loss of generality, we assume Z_t is a mean zero process, as in practice, we can subtract the sample mean from the data. We denote the h^{th} autocovariance as $\gamma_h = \mathbb{E}(Z_t Z_{t-h})$ and let $Y_t = Z_t, X_p = \begin{bmatrix} Z_{t-1} \dots Z_{t-p} \end{bmatrix}$. We wish to study the behavior of the expected squared error loss for different orders p up to a prespecified maximum order, $p_{\max}$. Specifically, we study

$$L_{\text{AR}}(p, \beta) := \mathbb{E}\left[(Y_t - X_p\beta)^2\right] .$$

In this case, β is a nuisance parameter. For fixed p, $L_{\text{AR}}(p, \beta)$ is the usual least squares problem and we can solve for β to get $\beta_p^* = \mathbb{E}\left(X_p^T X_p\right)^{-1} \mathbb{E}\left(X_p^T Y_t\right)$. Plugging β_p^* back in to the expected loss and simplifying, we get a loss that only depends on p, for which we use the shorthand notation $L_{\text{AR}}(p)$ and get that

$$L_{\text{AR}}(p) = \mathbb{E}\left(Y_t^2\right) - \mathbb{E}\left(X_p^T Y_t\right)^T \mathbb{E}\left(X_p^T X_p\right)^{-1} \mathbb{E}\left(X_p^T Y_t\right) .$$

By stationarity of Z_t, this simplifies to

$$L_{\text{AR}}(p) = \gamma_0 - \begin{bmatrix} \gamma_1 & \dots & \gamma_p \end{bmatrix} \begin{bmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \cdots & \\ \gamma_1 & \gamma_0 & \gamma_1 & \cdots & \\ \gamma_2 & \gamma_1 & \gamma_0 & \cdots & \\ \vdots & \vdots & \vdots & \ddots & \\ \gamma_{p-1} & \gamma_{p-2} & \gamma_{p-3} & \cdots & \gamma_0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_p \end{bmatrix} .$$

Note that up to this point we have only assumed Z_t is stationary and have made no assumptions on the structure of the data generating process. Now suppose Z_t is not only stationary but is also an $\text{AR}(p_0)$ process, that is,

$$Z_t = a_1 Z_{t-1} + \cdots + a_{p_0} Z_{t-p_0} + \epsilon_t,$$

where $\mathbb{E}(\epsilon_t) = 0$, $\mathbb{E}(\epsilon_t^2) = \sigma_\epsilon^2$ and $\mathbb{E}(\epsilon_s \epsilon_t) = 0$ for $s \neq t$. Then γ_h has a known form and we can calculate the expected squared error loss for each $p = 1, \dots, p_{\max}$.

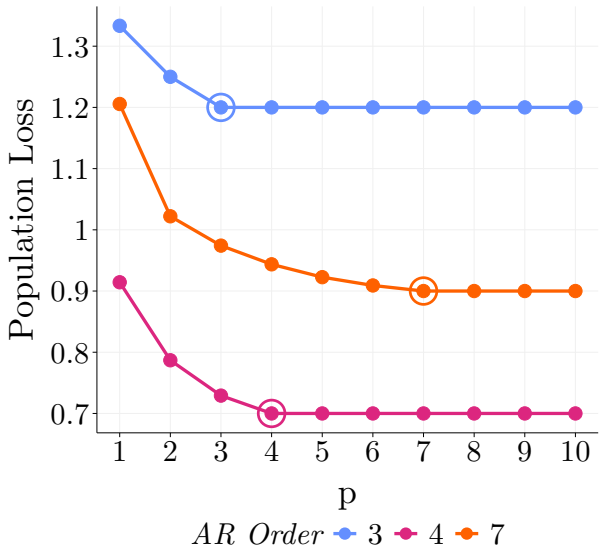


Figure 2.2: Population loss for AR(3), AR(4), and AR(7) processes with $\sigma_\epsilon^2 = 1.2, 0.7, 0.9$ respectively. As the fitted order increases to the true order the expected loss decreases monotonically to σ_ϵ^2. The true order for each line is denoted by a hollow circle.

Intuitively, the expected squared error loss should be flat after the true order p_0 as the first p_0 lags should contain all the information needed for prediction. Furthermore, the error of the process is white noise and thus unpredictable. Therefore, the lowest achievable expected squared error should be the variance of the error. In Figure 2.2, we compute the expected squared error loss for several AR processes and see that our intuition holds. It can be seen that the expected squared error loss is indeed flat once the fitted order is at least as large as the true order p_0 and the eventual value of the expected loss is the variance of the error, σ_ϵ^2. The behavior of the loss observed in Figure 2.2 can be proven mathematically and follows immediately from the multivariate case proved in Theorem 1. Given this behavior, if we add an appropriately sized penalty to the fitted order p and consider orders large enough ($p_{\max} > p_0$), we should be able to design a correct order selection procedure in the sense that when the process is AR(p_0) we will recover the true order p_0. Specifically the true order

can be found as

$$p_0 = \underset{p \in \{0,\ldots,p_{\max}\}}{\arg\min} \; L_{\mathrm{AR}}(p) + \lambda p \,.$$

If λ is too large, we would underestimate the order. However, we must also have $\lambda > 0$ to avoid multiple solutions and an undefined parameter. In practice, we rarely deal with univariate time series so in the next section we extend these concepts to the multivariate case.

2.3 Extension to the multivariate case

In this section, we extend the concepts of Section 2.2 to the multivariate case and show a similar behavior of the expected squared error loss. Suppose $Z_t = \left[Z_{t,1}, \ldots, Z_{t,k}\right]^T$ is a k-dimensional column vector. We assume that Z_t is a stable process, as formalized in Assumption 2 (stability) in Appendix A.1. The h^{th} autocovariance matrix is defined as $\Gamma_h = \mathbb{E}\left(Z_t Z_{t-h}^T\right) \in \mathbb{R}^{k \times k}$. Note that Γ_h is not symmetric in general, but $\Gamma_h = \Gamma_{-h}^T$. Similar to the univariate case, we define $Y_t = Z_t, X_p = \left[Z_{t-1}^T \ldots Z_{t-p}^T\right]^T$ where $Y_t \in \mathbb{R}^{k \times 1}$ and $X_p \in \mathbb{R}^{kp \times 1}$. We study the expected squared error loss at different values of p,

$$L_{\mathrm{VAR}}(p, A_p) := \mathbb{E}\left[\left(Y_t - A_p X_p\right)^T \left(Y_t - A_p X_p\right)\right] .$$

We proceed by profiling out A_p. For fixed p, we have $A_p^* = \mathbb{E}\left(Y_t X_p^T\right) \mathbb{E}\left(X_p X_p^T\right)^{-1}$. Similar to the univariate case we plug A_p^* back into the expected loss and simplify which we denote as $L_{\mathrm{VAR}}(p)$. It is shown in Appendix A.2 that $L_{\mathrm{VAR}}(p)$ is

$$L_{\mathrm{VAR}}(p) = \mathrm{Tr}\left(\Gamma_0\right) - \mathrm{Tr}\left(\begin{bmatrix}\Gamma_1 & \ldots & \Gamma_p\end{bmatrix} \begin{bmatrix}\Gamma_0 & \ldots & \Gamma_{p-1} \\ \vdots & & \vdots \\ \Gamma_{p-1}^T & \ldots & \Gamma_0\end{bmatrix}^{-1} \begin{bmatrix}\Gamma_1^T \\ \vdots \\ \Gamma_p^T\end{bmatrix}\right) .$$

For $p = 0$ there are no prior lags as predictors so we get that $L_{\mathrm{VAR}}(0) = \mathrm{Tr}\left(\Gamma_0\right)$. Similar to the univariate case, if Z_t is a stationary VAR(p_0) process, the multivariate loss decreases until the fitted order is equal to the true order at which point the population loss is constant with a value of $\mathrm{Tr}(\Sigma_\epsilon)$. This behavior of the loss is formally stated in Theorem 1.

Theorem 1 (Flat Loss)**.** *Suppose Z_t is a* $\mathrm{VAR}(p_0)$ *process with standard white noise. That is,* $Z_t = \sum_{i=1}^{p_0} A_i Z_{t-i} + \epsilon_t$ *where* $\mathbb{E}(\epsilon_t) = 0, \mathbb{E}\left(\epsilon_t \epsilon_t^T\right) = \Sigma_\epsilon$ *and* $\mathbb{E}\left(\epsilon_t \epsilon_s^T\right) = 0$ *for* $t \neq s$*. If Assumption 2 (stability), Assumption 3 (invertibility), and Assumption 4 (irreducibility) hold, then*

$$\begin{cases} L_{\mathrm{VAR}}(p) < L_{\mathrm{VAR}}(p-1) & \text{if } p \leq p_0 \\ L_{\mathrm{VAR}}(p) = \mathrm{Tr}(\Sigma_\epsilon) & \text{if } p \geq p_0 \,. \end{cases}$$

For $L_{\mathrm{VAR}}(p_0)$, Theorem 1 implies both $L_{\mathrm{VAR}}(p_0) < L_{\mathrm{VAR}}(p_0 - 1)$ and $L_{\mathrm{VAR}}(p_0) = \mathrm{Tr}(\Sigma_\epsilon)$. If we penalize the fitted order p by an appropriate amount, and use $p_{\max} > p_0$, we would obtain a procedure that recovers the true order, p_0. This is formally stated in Corollary 1.

Corollary 1. *For a VAR*(p_0) *process that satisfies the conditions of Theorem 1 and for* $\lambda \in (0, M)$ *and* $p_{\max} > p_0$ *we have*

$$p_0 = \underset{p \in \{0, \ldots, p_{\max}\}}{\arg\min} \; L_{\mathrm{VAR}}(p) + \lambda p \,,$$

where $M = \min\left(L_{\mathrm{VAR}}(p_0 - 1) - L_{\mathrm{VAR}}(p_0), \left[L_{\mathrm{VAR}}(p_0 - 2) - L_{\mathrm{VAR}}(p_0)\right]/2, \ldots, \left[L_{\mathrm{VAR}}(0) - L_{\mathrm{VAR}}(p_0)\right]/p_0\right)$.

It is worth noting that Assumptions 2 to 4 are mild and standard assumptions which hold in many applications. Further discussion on the assumptions is provided in Appendix A.1.

To estimate p_0, we need to estimate $L_{\mathrm{VAR}}(p) + \lambda p$ for each p. While we use the form $L_{\mathrm{VAR}}(p)$ with the autocovariance matrices to establish the theoretical results, $L_{\mathrm{VAR}}(p)$ can be expressed as the expected squared error loss: $L_{\mathrm{VAR}}(p) = \mathbb{E}\left[\left(Y_t - A_p^* X_p\right)^T \left(Y_t - A_p^* X_p\right)\right]$. Therefore, a natural estimator of $L_{\mathrm{VAR}}(p)$ is the sample squared error loss, as defined formally below.

Let $\{\mathbf{z}_t\}_{t=1}^{n+p_{\max}}$ denote the observed k-dimensional time series of length $n + p_{\max}$. We assume the sample is length $n + p_{\max}$ as fitting a $\mathrm{VAR}(p_{\max})$ model requires the $p_{\max}$ prior data points as covariates. As a result for a $\mathrm{VAR}(p_{\max})$ model we only have n usable data points. Denoting $\mathbf{Y} = \begin{bmatrix}\mathbf{z}_1, \ldots, \mathbf{z}_n\end{bmatrix} \in \mathbb{R}^{k \times n}$, $\mathbf{x}_{i,p} = \begin{bmatrix}\mathbf{z}_{i-1}^T & \ldots & \mathbf{z}_{i-p}^T\end{bmatrix}^T \in \mathbb{R}^{kp \times 1}$ and $\mathbf{X}_p = \begin{bmatrix}\mathbf{x}_{1,p} & \ldots & \mathbf{x}_{n,p}\end{bmatrix} \in \mathbb{R}^{kp \times n}$, an estimate of the expected squared error loss at a fitted order p is given by

$$\begin{aligned} \hat{L}_{\mathrm{VAR}}(p) &= \mathrm{Tr}\left(\frac{1}{n}\left(\mathbf{Y} - \hat{A}_p \mathbf{X}_p\right)^T \left(\mathbf{Y} - \hat{A}_p \mathbf{X}_p\right)\right) \\ &= \mathrm{Tr}\left(\frac{1}{n}\left(\mathbf{Y} - \hat{A}_p \mathbf{X}_p\right) \left(\mathbf{Y} - \hat{A}_p \mathbf{X}_p\right)^T\right), \end{aligned}$$

where $\hat{A}_p = \mathbf{Y}\,\mathbf{X}_p^T\left(\mathbf{X}_p\,\mathbf{X}_p^T\right)^{-1}$ and we used $\mathrm{Tr}(A^TB) = \mathrm{Tr}(AB^T)$. We can then define p^*_{MIC} which relies on known $\lambda \in (0, M)$ as

$$p^*_{\mathrm{MIC}} = \underset{p\in\{0,\ldots,p_{\max}\}}{\arg\min}\ \hat{L}_{\mathrm{VAR}}(p) + \lambda p\,. \tag{2.1}$$

Note that it is not possible to use this estimator in practice as λ is unknown. The MIC estimator we use in practice is shown in Eq. (2.2). To solve the minimization problem in Eq. (2.1), we can simply compute $\hat{L}_{\mathrm{VAR}}(p)$ for each $p = 0, \ldots, p_{\max}$. This amounts to solving $p_{\max}$ multivariate least squares regression problems and computing the trace of the residual matrix. Theorem 2 establishes the consistency of $\hat{L}_{\mathrm{VAR}}(p)$.

Theorem 2 (Consistency of Loss)**.** *Under the assumptions of Theorem 1, we have that*

$$\left|\hat{L}_{\mathrm{VAR}}(p) - L_{\mathrm{VAR}}(p)\right| = o_p(n^{-1/2+\delta})\,,$$

for all $\delta > 0$.

As shown in the proof of Theorem 2, the consistency of $\hat{L}_{\mathrm{VAR}}(p)$ relies on the consistency of the sample autocovariances and the rate of convergence of $\hat{L}_{\mathrm{VAR}}(p)$ is the same as the rate of convergence of the sample autocovariance. One benefit of this information criterion is that it does not rely on a likelihood. In fact, this result will hold as long as the sample autocovariance is consistent. In that case, the rate of convergence will change to that of the sample autocovariance. An immediate consequence of Theorem 2 is consistency of the p^*_{MIC}.

Corollary 2 (Consistency of order estimate)**.** *Under the assumptions of Theorem 1 we have that*

$$p^*_{\mathrm{MIC}} \to_p p_0,$$

where $\to_p$ *denotes convergence in probability.*

2.4 MIC estimator of VAR order

While our theoretical analyses assume $\lambda \in (0, M)$ is known, in practice λ is unknown and needs to be selected. The flat loss concept from Theorem 1 tells us that once the fitted order exceeds the true

order, the loss should be constant. In practice there is sampling variability so the loss will never completely stabilize. We generate an estimate of this variability using a "self-tuning" approach.

In our "self-tuning" approach, we fit models from lag order $p_{\max} + 1$ to $2p_{\max}$ and take the absolute value of the mean of the difference between each loss and the subsequent loss. While it is possible to use orders larger than $2p_{\max}$, the trade-off is fitting larger order models consumes more prior data points as covariates and reduces usable sample size. We find $2p_{\max}$ works well in practice. Our "self-tuning" approach computes

$$\mathrm{MD} = \left|\mathrm{mean}\left(\hat{L}_{\mathrm{VAR}}(p_{\max}) - \hat{L}_{\mathrm{VAR}}(p_{\max}+1)), \ldots, \hat{L}_{\mathrm{VAR}}(2p_{\max}-1) - \hat{L}_{\mathrm{VAR}}(2p_{\max})\right)\right| .$$

We then scale MD by $\sqrt{n/(k^2\log(n))}$ to get

$$\lambda_{\mathrm{ST}} = \mathrm{MD}\sqrt{\frac{n}{k^2\log(n)}} .$$

With this choice of λ, our estimator is defined as

$$\hat{p}_{\mathrm{MIC}} = \underset{p\in\{0,\ldots,p_{\max}\}}{\arg\min}\ \hat{L}_{\mathrm{VAR}}(p) + \lambda_{\mathrm{ST}} p . \tag{2.2}$$

We next discuss the choice of MD as well as the scaling $\sqrt{n/(k^2\log(n))}$.

Due to the flat loss concept, each $\hat{L}_{\mathrm{VAR}}(p_{\max}+i) - \hat{L}_{\mathrm{VAR}}(p_{\max}+i+1)$ should represent an estimate of how the sample loss changes when we have exceeded the true order and increase the fitted order by 1. We average over $p_{\max}$ of these to reduce the variance in this estimate. When computing the mean, subsequent differences cancel and this quantity can be simplified as

$$\mathrm{MD} = \left|\frac{\hat{L}_{\mathrm{VAR}}(p_{\max}) - \hat{L}_{\mathrm{VAR}}(2p_{\max})}{p_{\max}}\right| .$$

Thus, MD can be computed efficiently as it only requires fitting one additional regression of order $2p_{\max}$. It is also worth noting that $\hat{L}_{\mathrm{VAR}}(p_{\max}), \ldots, \hat{L}_{\mathrm{VAR}}(2p_{\max})$ converge to the same asymptotic distribution and are asymptotically perfectly correlated. In this instance, the correlation is beneficial as it further reduces the variance of our estimate since for two random variables, X, Y, $\mathrm{Var}(X-Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) - 2\mathrm{Cov}(X,Y)$.

To understand why it is necessary to scale MD, consider the case where we instead use $\lambda_{\mathrm{ST}} = \mathrm{MD}$. To simplify notation and provide a more concrete setting, consider $p_{\max} = 10$. With these, the score for order $2p_{\max} := 20$ based on Eq. (2.2) becomes

$$\begin{aligned}\hat{L}_{\mathrm{VAR}}(20) + \frac{\hat{L}_{\mathrm{VAR}}(10) - \hat{L}_{\mathrm{VAR}}(20)}{10}20 &= \frac{20}{10}\hat{L}_{\mathrm{VAR}}(10) - \frac{10}{10}\hat{L}_{\mathrm{VAR}}(20) \\ &= \hat{L}_{\mathrm{VAR}}(10) + \frac{\hat{L}_{\mathrm{VAR}}(10) - \hat{L}_{\mathrm{VAR}}(20)}{10}10\,,\end{aligned}$$

where we have assumed that $\hat{L}_{\mathrm{VAR}}(10) > \hat{L}_{\mathrm{VAR}}(20)$ so we can ignore the absolute value in MD. This will always be true if using the same dataset to compute $\hat{L}_{\mathrm{VAR}}(10)$ and $\hat{L}_{\mathrm{VAR}}(20)$. The last equation on the right hand side is exactly the value of the penalized loss for order $p_{\max} := 10$. In other words, setting $\lambda = \mathrm{MD}$ treats models $p_{\max}$ and $2p_{\max}$ as equally viable. However, these models are not equally viable and we want to enforce a belief that higher orders are worse. Thus we need to choose a penalty that is larger than MD. One way to do this is to scale MD by a factor > 1. From Theorem 2 we know that $\hat{L}_{\mathrm{VAR}} = L_{\mathrm{VAR}} + o_P(n^{-1/2+\delta}) \quad \forall\, \delta > 0$ so scaling by $n^{1/2}$ is too fast. In practice we find that $\sqrt{n/\log(n)}$ works well. Lastly we scale by $1/k$ since each additional order fitted requires estimating k more parameters than the prior order (see e.g. the proof of Theorem 1).

We now compare our method, MIC, to AIC, BIC, and HQ. For ease of comparison, we write the prediction error matrix as

$$\hat{\Sigma}_p = \frac{1}{n}\left(\mathbf{Y} - \hat{A}_p\,\mathbf{X}_p\right)\left(\mathbf{Y} - \hat{A}_p\,\mathbf{X}_p\right)^T.$$

Note that $\mathrm{Tr}(\hat{\Sigma}_p) = \hat{L}_{\mathrm{VAR}}(p)$. The criteria considered can be written as follows

$$\begin{aligned}\mathrm{MIC}(p) &= \mathrm{Tr}(\hat{\Sigma}_p) + \lambda_{\mathrm{ST}}p & \mathrm{AIC}(p) &= \log\left|\hat{\Sigma}_p\right| + \frac{2}{n}k^2p\,; \\ \mathrm{BIC}(p) &= \log\left|\hat{\Sigma}_p\right| + \frac{\log n}{n}k^2p & \mathrm{HQ}(p) &= \log\left|\hat{\Sigma}_p\right| + \frac{2\log\log n}{n}k^2p\,.\end{aligned}$$

Note that $\log$ refers to the natural logarithm. The above formulations show that all criteria rely on the same estimate of the error matrix $\hat{\Sigma}_p$ and differ only in how they use it — $\mathrm{Tr}()$ or $\log|\cdot|$ —- and penalize that information.

2.4.1 Alternative choices of λ

We also considered two other methods to select order based on the flat loss concept discussed. The first method, which we refer to as MIC-sp, uses MIC with a penalty λ_{sp} that is chosen by splitting the data into train and test sets. Due to the time dependent nature of our data we use the first 70% of observations as training and the remainder as test data. Models from $p_{\max}$ to $2p_{\max}$ are fit on the training data to estimate $\hat{A}_p$ and their prediction errors computed on the test data are denoted as e.g. $\hat{\Sigma}_{\text{test},p_{\max}}$. We set

$$\lambda_{\text{sp}} = \text{mean}\left(\left|\text{Tr}(\hat{\Sigma}_{\text{test},p_{\max}}) - \text{Tr}(\hat{\Sigma}_{\text{test},p_{\max}+1})\right|, \ldots, \left|\text{Tr}(\hat{\Sigma}_{\text{test},2p_{\max}-1}) - \text{Tr}(\hat{\Sigma}_{\text{test},2p_{\max}})\right|\right) .$$

Due to the flat loss property, each of these differences should be 0 and any sample variability should be captured in λ_{sp}. Lastly, we consider another procedure which we denote as MIC-mt. We again use a 70-30 train-test split and fit VAR models of order $0, \ldots, p_{\max}$ on the train dataset. Similarly, we compute the errors of each fitted model on the test data. MIC-mt then chooses the order $0, \ldots, p_{\max}$ that minimizes the test error. Simulations comparing all three methods in the case of a diagonal covariance matrix with Gaussian errors are shown in Figure A.1. The results show that MIC, which indicates the MIC method with self-tuned λ, performs the best across a variety of sample sizes and dimensions. It is only consistently outperformed by MIC-sp in the AR(2) case. Results for other simulations in Section 2.5 are not shown but are qualitatively similar. Thus, we proceed with out self-tuning approach for selecting λ.

2.5 Simulations

2.5.1 Order selection accuracy

In this section, we compare the accuracy of MIC, AIC, BIC, and HQ order selection methods using simulated data. In general, we will use $\text{VAR}_k(p)$ to denote the dimension k and order p of the process. We will also use $\text{U}(a, b)$ to denote a $\text{Unif}(a, b)$ distribution. We consider VAR models with 4 different dimensions and 3 different error structures. The first is an autoregressive process of order 2, AR(2), with parameters (0.3, 0.1). The second is a $\text{VAR}_2(2)$ process. The entries of the first lag

coefficient matrix are 25% sparse and randomly drawn from either a $\mathrm{U}(0.1, 0.3)$ or a $\mathrm{U}(-0.3, -0.1)$ each with 50% probability. The entries of the second lag coefficient matrix are 50% sparse and randomly drawn from either a $\mathrm{U}(0.07, 0.2)$ or a $\mathrm{U}(-0.2, -0.07)$ each with 50% probability. The third simulation setting is a $\mathrm{VAR}_5(3)$. All lag coefficient matrices have 60% sparsity. In the first lag coefficient matrix, the non-zero entries are drawn from a $\mathrm{U}(0.1, 0.3)$ or a $\mathrm{U}(-0.3, -0.1)$ each with equal probability. The second lag coefficient matrix uses a $\mathrm{U}(0.1, 0.2)$ or a $\mathrm{U}(-0.2, -0.1)$ while the third uses a $\mathrm{U}(0.05, 0.1)$ or a $\mathrm{U}(-0.1, -0.05)$. The fourth simulation setting is a $\mathrm{VAR}_{10}(3)$ process where the first lag coefficient matrix has 40% sparsity and the non-zero entries are drawn from a $\mathrm{U}(0.1, 0.3)$ or a $\mathrm{U}(-0.3, -0.1)$.The second lag coefficient matrix has 80% sparsity, but the remaining entries are drawn from a $\mathrm{U}(-0.2, 0.2)$. The final lag coefficient matrix has 80% sparsity with remaining entries drawn from a $\mathrm{U}(-0.1, 0.1)$. All coefficient matrices are generated once and the same matrices are used throughout the simulations. Stability for each setting is verified using the method of Lütkepohl (2005, pp. 14 - 17).

All datasets are simulated using three error structures. The first is mean-zero Gaussian errors with an identity covariance matrix while the second uses a randomly generated covariance matrix. The covariance matrix is generated by computing a $k \times k$ matrix with entries drawn from a $\mathrm{U}(-3, 3)$. The matrix is then symmetrized by left multiplying by its transpose. We enforce a maximum condition number of 100 for each matrix by adding 0.001 to all diagonal until the condition number is met. After the covariance matrix is reconditioned, it is scaled to have unit variances. The third error structure is a Gaussian mixture model with 5 components. Each component is Gaussian where the mean vector is generated from a $\mathrm{U}(-5, 5)$. For each k we then subtract the mean across all 5 components so that the mean of the component means is 0. The covariances are $k \times k$ matrices with entries drawn from a $\mathrm{U}(-3, 3)$. The matrices are subsequently symmetrized, reconditioned, and rescaled as explained above. We simulate $n = 250, 500, 1000, 2000, 5000$ observations for each setting except for $\mathrm{VAR}_{10}(3)$ where $n = 250$ is excluded as the number of parameters exceeds the number of data points.

Lastly, we consider a $\mathrm{VAR}_3(2)$ process that switches between one of two regimes as in Kalli and Griffin (2018). Both regimes are order 2 and thus the true order is 2 but they use different coefficient

and error covariance matrices. Due to the switching nature of this process, it is not stationary and we present these results to study how the methods perform under misspecification.

We compute each criteria, MIC, AIC, BIC, and HQ, for $p = 0, \ldots, 10 := p_{\max}$. The estimated orders for each criteria are those that achieve the minimum value. That is,

$$\hat{p}_{\text{CRITERION}} = \underset{p \in \{0,\ldots,p_{\max}\}}{\arg\min} \ \text{CRITERION}(p)\,.$$

For each error structure and VAR model, we simulate $b = 250$ times and compute the proportion of times the correct order is estimated. That is, we use

$$\text{Accuracy} = \frac{1}{b}\sum_{i=1}^{b} I(\hat{p}_{\text{CRITERION}} = p_0)\,.$$

The simulation results are summarized in Figures 2.3 to 2.6. When errors are diagonal Gaussian and the dimension is large, MIC outperforms AIC, HQ, and BIC as shown in Figure 2.3. This makes sense as in this setting AIC, HQ, and BIC all use the entire error matrix, including the off-diagonals, through the determinant. When the true errors are diagonal, the estimated off-diagonals can contain incorrect information. On the other hand, MIC only uses the diagonals and so discards the potentially misleading off-diagonal information. For diagonal errors with small sample sizes and small dimension, HQ, and BIC perform slightly better although MIC is still competitive. As previously noted, AIC is not consistent for the $\mathrm{AR}(2)$ and $\mathrm{VAR}_2(2)$ processes. Results for non-diagonal Gaussian errors in Figure 2.4 show much better performance for AIC, BIC, and HQ. MIC still appears to consistently estimate the order as sample size increases, but the small sample performance is now worse relative to the other methods. This makes sense as there is useful information contained in the off-diagonals of the error matrix that AIC, BIC, and HQ can leverage while MIC cannot. We see similar trends for Gaussian mixture errors in Figure 2.5. Note that performance is sometimes better and sometimes worse than the corresponding results in Figure 2.4. This is likely due to variation in the simulated error covariances and means.

For the misspecification simulation, the $\mathrm{VAR}_3(2)$ switching process (Figure 2.6), the performance of AIC and HQ deteriorates as sample size increases. This is much more pronounced for AIC with estimated order accuracy reducing from 0.91 when $n = 250$ to 0.13 when $n = 5,000$. For HQ

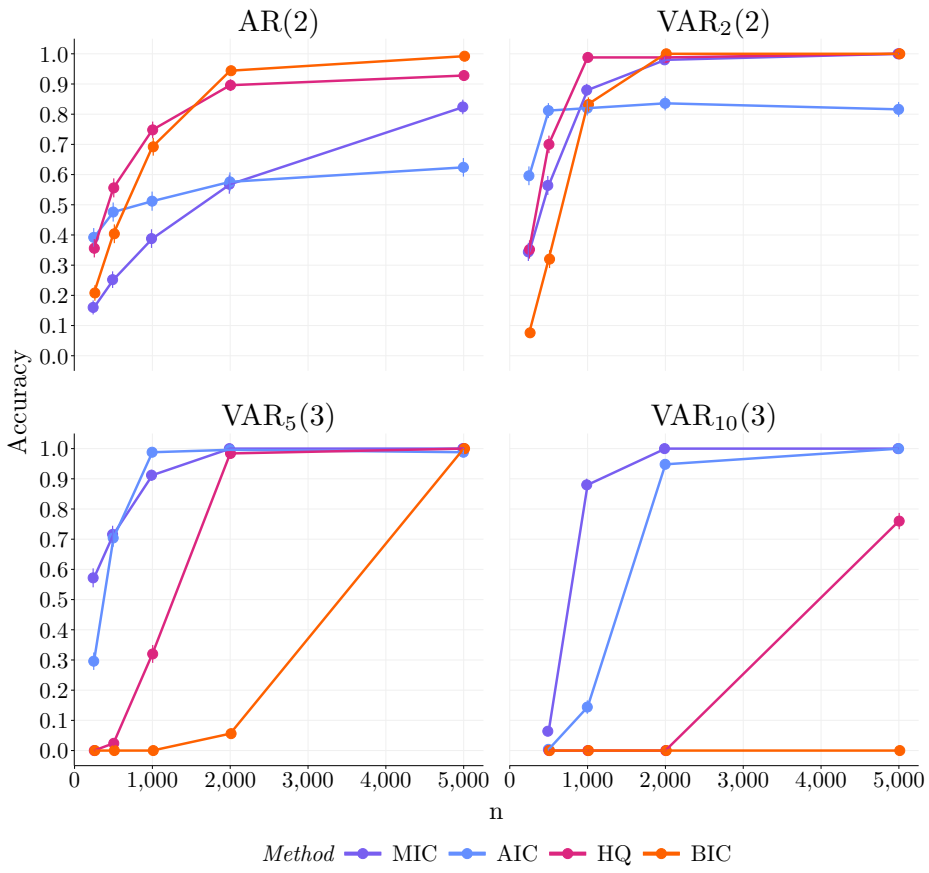


Figure 2.3: **Diagonal Gaussian errors**. Simulation results for accuracy of specific order selection method and simulation setting with diagonal Gaussian errors. Vertical lines indicate standard errors.

the estimated order accuracy only reduces slightly to 0.95 when $n = 5,000$. Given that accuracy is perfect for $n = 250, 500, 1000$ this is likely not due to sampling variability. This does not appear to hold for BIC, but since BIC converges much slower than AIC and HQ, as shown in Figures 2.3 to 2.5, it is likely that the sample size is not large enough. We explore the deteriorating performance with increasing sample size in Figure A.2 and find that this is indeed the case with AIC, BIC, and HQ all having 0 accuracy when $n = 100,000$. In contrast, MIC shows robust performance and perfectly estimates the order regardless of the sample size.

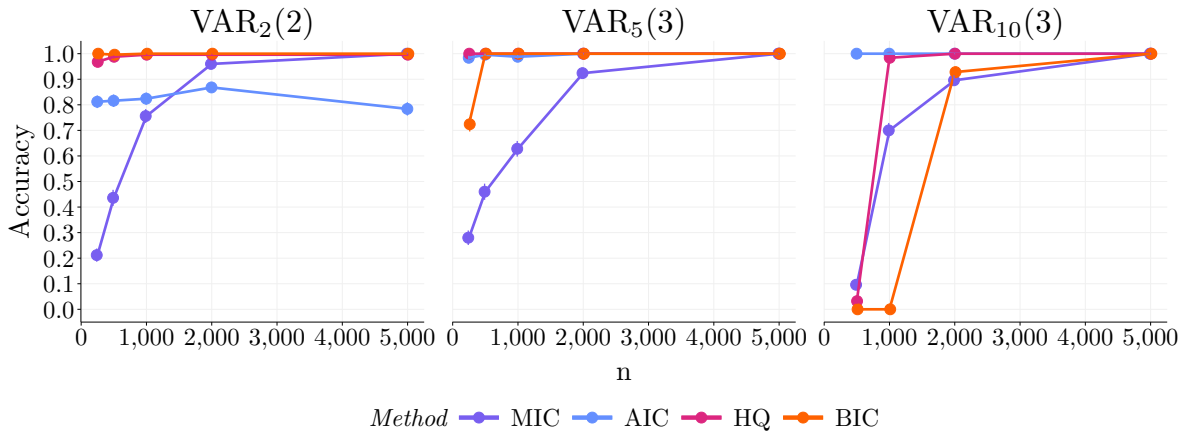


Figure 2.4: **Non-diagonal Gaussian errors**. Simulation results for accuracy of specific order selection method and simulation setting with non-diagonal Gaussian errors. Vertical lines indicate standard errors.

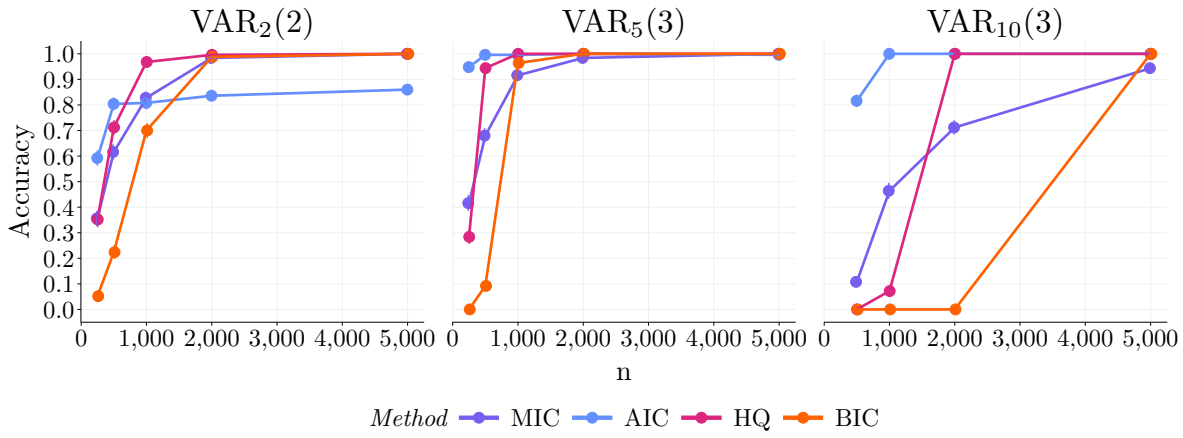


Figure 2.5: **Gaussian mixture errors**. Simulation results for accuracy of specific order selection method and simulation setting with Gaussian mixture errors. Vertical lines indicate standard errors.

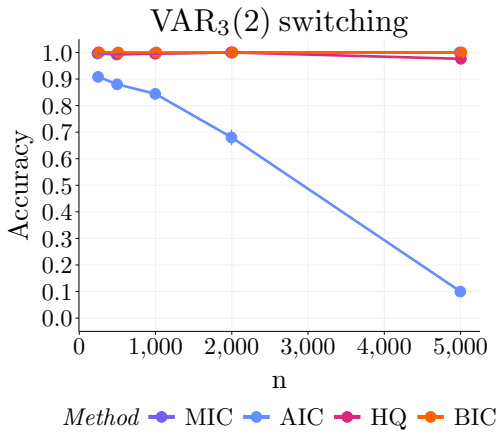


Figure 2.6: **VAR$_3$(2) switching**. Simulation results for accuracy of specific order selection method and simulation setting. Vertical lines indicate standard errors.

While MIC is outperformed by AIC, BIC, and HQ when the true error matrix has off-diagonal elements, it offers better performance when the true error matrix is diagonal and appears to be more robust to misspecification as shown by the performance in the $\mathrm{VAR}_3(2)$ switching setting. Thus MIC offers a viable alternative to AIC/BIC/HQ as in many practical applications errors are unlikely to be Gaussian. In the next section, we investigate the performance of MIC compared to AIC/BIC/HQ by comparing forecast performance on two datasets.

2.5.2 Over and under selection probability

As was previously mentioned, it is known that AIC does not provide a consistent estimate of the VAR order when the dimension is small. To this end, we compare the likelihood of over and under selection of the model order for each of the order selection methods using simulations. Since MIC uses an estimated λ we evaluate its theoretical properties with an oracle λ value, which we denote as MIC-oracle. Specifically we choose $\lambda_{\text{oracle}} = M/2$ where M is defined in Corollary 1. The results for diagonal gaussian errors are shown in Figures A.3 and A.4 while the non-diagonal gaussian errors are shown in Figure A.5. Overall we see that MIC-oracle tends to select an order that is

larger than the true order (over selection) while the alternative methods AIC, BIC, and HQ, tend to select an order that is smaller than the true order (under selection). Figure A.5 shows that, when the dimension of the process is large and errors are non-diagonal Gaussian, MIC-oracle suffers from worse over selection relative to the under selection from AIC, BIC, and HQ.

2.6 Applications

In this section we apply our MIC method to two different data analysis problems and compare to AIC, BIC, and HQ. The first problem is financial forecasting, while the second is forecasting COVID-19 outcomes. In both problems, we follow Nicholson et al. (2020) by comparing the weighted mean squared forecast error (wMSFE). For all applications the first 80% of observations are used to estimate the order for MIC, AIC, BIC, and HQ, while the last 20% are used for forecasting. Formally, if n represents the total number of observations, then the first $T_1 = \lfloor 0.8n \rfloor$ data points are used to estimate the order for each method and the remaining observations are used for testing. We use a rolling window of size T_1 to perform one-step ahead forecasts. That is, if t indexes the observation we are forecasting then we use observations $t - T_1, \ldots, t - 1$ as the rolling window. For each rolling window, we standardize each variable in the series by subtracting the mean and dividing by the standard deviation. The observation we are forecasting is also standardized using the mean/SD from the rolling window. We then fit a VAR model corresponding to the orders chosen by MIC, AIC, BIC, and HQ to this standardized rolling window and predict observation t. The wMSFE for method m is computed over all series and forecast time points as

$$\text{wMSFE}(m) = \frac{1}{k(n - T_1)} \sum_{i=1}^{k} \sum_{t=T_1+1}^{n} \left(\frac{y_{i,t} - \hat{y}_{i,t}^{m}}{\hat{\sigma}_i} \right)^2 ,$$

where $\hat{\sigma}_i$ is the standard deviation of the variable i computed over the forecast observations.

2.6.1 Daily realized stock variances

We compare forecast performance of VAR models with order selected by MIC, AIC, BIC, and HQ using data from the Oxford-Man Institute of Quantitative Finance obtained from an older version

of the `mfGARCH R` package, https://github.com/onnokleen/mfGARCH, from Conrad and Kleen (2020). Specifically, we analyze 5-minute return daily realized variances for up to 17 stocks from January 3, 2000 to June 27, 2018 ($n = 4,847$). We perform two analyses: one using the same $k = 16$ stocks as in Nicholson et al. (2020) as well as $k = 7$ stocks from Son et al. (2023). Many stocks from Nicholson et al. (2020) and Son et al. (2023) overlap and there are only 17 total unique stocks. Due to high levels of missingness (34%) we exclude OMXSPI, an index of the Stockholm Stock Exchange, from our analysis based on the Son et al. (2023) stocks. A full list of the stocks analyzed is given in Appendix A.5. All data are log-transformed to make them stationary. As we are not specifically interested in high-dimensional applications we estimate the order for each order selection method using a $p_{\max} = 10$, equivalent to two trading weeks.

Forecast results for both the $k = 16$ and $k = 7$ analysis are displayed in Figure 2.7. Overall, we see that the methods considered give very similar forecast accuracy despite a large range of orders. For example, in Figure 2.7(a), the order varies from a low of 2 chosen by BIC to a high of 9 chosen by MIC.

2.6.2 COVID-19 in New York City

We next compare performance of order selection methods in forecasting COVID-19 outcomes in New York City. Daily data on deaths, cases, and hospitalizations in New York City due to COVID-19 are available starting Februrary 29, 2020 at City of New York's website. We specifically analyze data from February 29, 2020 to July 8, 2024 ($n = 1,592$). All data are first differenced to make them stationary and we use $p_{\max} = 30$. As a check, we run an Augmented Dickey-Full test (ADF, Said and Dickey, 1984) and a Kwiatkowski-Phillips-Schmidt-Shin (KPSS, Kwiatkowski et al., 1992)) test for each series after differencing. The null hypothesis of the ADF test is that a unit root is present in the time series while the null hypothesis of the KPSS test is that the series is trend-stationary. All series pass the ADF test with $p < 0.01$ and the KPSS test with $p > 0.1$.

Forecast results are displayed in Figure 2.8. We see that AIC, BIC, and HQ all fit models using around a month's worth of prior data points ($p = 30$) to forecast the next day. However, these models are substantially worse than the model fitted using MIC order selection which only uses

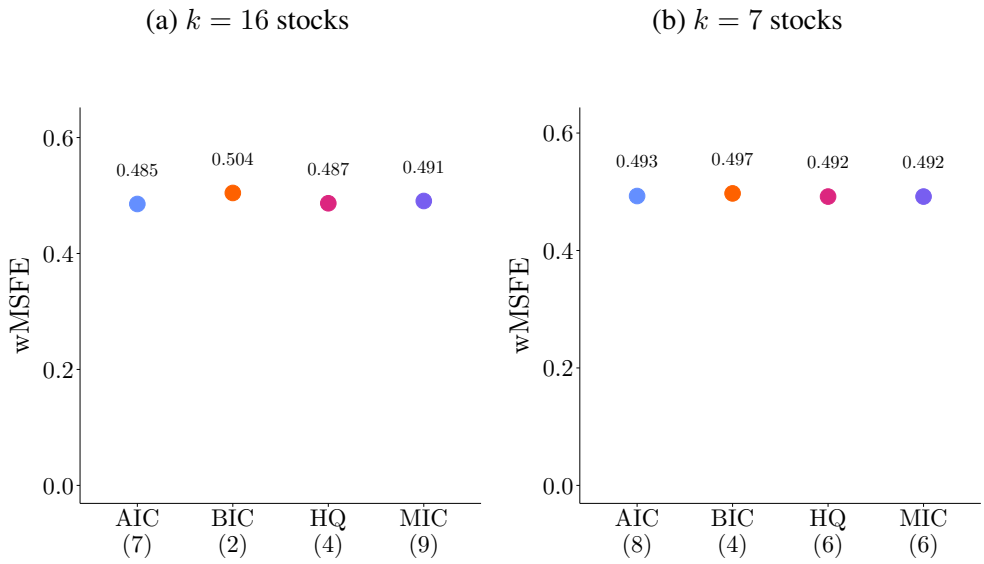


Figure 2.7: Comparison order selection methods based on weighted Mean Squared Forecast Error for daily realized stock variances for (a) $k = 16$ stocks and (b) $k = 7$ stocks. Order selected by each method is displayed in parentheses below method name.

around a week of prior data points ($p = 8$). In fact, the forecast accuracy of the model fitted using MIC is around 30% better than the accruacy of those fit by AIC/BIC/HQ.

2.7 Discussion

In this chapter, we proposed the mean square information criterion (MIC), a new approach for estimating the order of VAR processes. MIC is based on a key new observation: the flatness of the expected squared error loss after the fitted order exceeds the true order. We show, under relatively mild assumptions, that the true order can be estimated consistently by minimizing the MIC. Specifically, consistency of MIC only requires consistent estimates of the autocovariances.

Our proposed method, MIC, was compared to three other order selection criteria: AIC, BIC, and HQ. All criteria were compared based on the proportion of simulations in which the correct order was correctly estimated. Simulation settings ranged from univariate to 10-dimensional VAR models

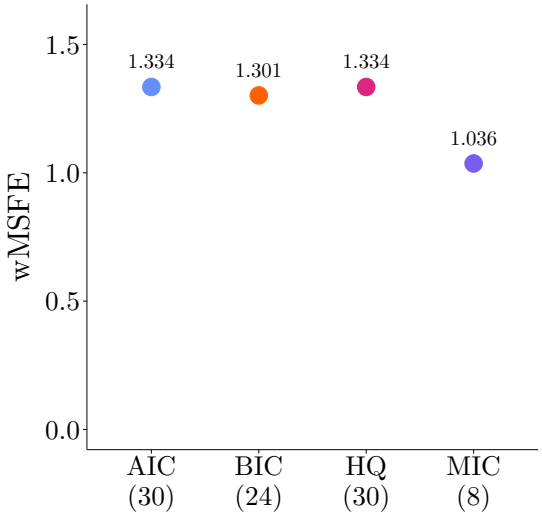


Figure 2.8: Comparison order selection methods based on weighted Mean Squared Forecast Error for COVID-19 outcomes. Order selected by each method is displayed in parentheses below method name.

with between 250 and 5,000 observations. Simulations included both Gaussian and Gaussian mixture error structures, as well as a regime switching $\mathrm{VAR}_3(2)$ model. While outperformed in Gaussian errors and small sample sizes, relative to the other criteria, MIC showed the best performance when the process had regime changes. As errors are unlikely ever Gaussian, these results suggest that MIC can be very useful in practice. This is confirmed in our data applications where order selection via MIC achieved comparable forecast accuracy for daily realized stock variance and substantially better accuracy in forecasting COVID-19 outcomes in NYC when compared to order selected via AIC, BIC, or HQ.

Chapter 3

ESTIMATION AND INFERENCE FOR SPECTRAL DIFFERENTIAL NETWORK ANALYSIS OF HIGH-DIMENSIONAL TIME SERIES

An earlier version of this work on estimation and consistency of high-dimensional spectral differential networks is published in Hellstern et al. (2025).

3.1 Introduction

Spectral network analysis of multivariate time series plays a key role in fields ranging from oceanography and seismology to neuroscience (Laurindo et al., 2019; James et al., 2017; Bloch et al., 2022). Studying individual networks can provide insights into how features interact; however, it is often of interest to study how networks change across conditions or in response to external interventions. In neuroscience, for example, many neurodegenerative disorders are associated with abnormal brain connectivity networks (Bloch et al., 2022). Spectral features are regularly used to interpret many types of neuroscientific data, from electroencephalography to magnetoencephalography data (Gnassounou et al., 2023; Richard et al., 2020; Dupré la Tour et al., 2018).

Coherence, the frequency domain analog to correlation, is a common choice for investigating interactions in multivariate time series analysis. This notion is especially appealing in neuroscience applications, where activities captured at different frequencies better reveal brain oscillations in sensory-cognitive processes. Therefore, despite the availability of nonlinear association measures, such as mutual information (Belghazi et al., 2018) and transfer entropy (Ursino et al., 2020), coherence is commonly used by neuroscientists to define brain functional connectivity networks.

Similarly to correlation, coherence includes the indirect effects of other nodes in the network. Thus, coherence may not be an informative measure of the dependence between nodes. A more direct measure is the inverse spectral density. The inverse spectral density between nodes i and j is

a rescaling of coherence after removing the linear effects of all other nodes $\{k \neq i, j\}$ (Dahlhaus, 2000). Therefore, the inverse spectral density better resembles the effective connectivity between brain regions (Friston, 2011), providing an initial understanding of how two regions may be causally related.

Advances in data collection have enabled the acquisition of datasets whose dimensionality far exceeds the number of observations ($p \gg n$). In the analysis of high-dimensional time series data, regularization techniques, such as the LASSO, are essential for ensuring both computational feasibility and statistical reliability (Banerjee et al., 2008). Similarly, in the frequency domain, regularization enhances numerical stability and improves overall performance (Böhm and von Sachs, 2009).

In this chapter, we propose the Spectral D-trace Difference (SDD), a direct estimator of the difference in inverse spectral densities. To our knowledge, SDD is the first method to target the differential network in spectral domain, with the only other differential network analysis approach for time series available in the time domain (Wang et al., 2021) and without calibrated inference. A key challenge in the theoretical analysis of our SDD estimator is the inherent temporal dependence between observations in time series data. To overcome this challenge, we develop new convergence rates for a general class of spectral density estimators, smoothed periodograms, by generalizing recent results on spectral analysis of time series data (Zhang and Zhang, 2025) to flexibly allow for both varying levels of dependence in the data, as well as different smoothing spans. These results are not only useful in establishing the convergence rates of SDD, but are also essential to our inference procedure, since valid inference requires larger smoothing spans than those covered by existing results. Moreover, we leverage modern proof techniques from Wang et al. (2021) and Negahban et al. (2012) to establish consistency of our SDD estimator by only assuming the difference in inverse spectral densities is sparse. This is in contrast to existing consistency results for direct differential network analysis with i.i.d. data (Yuan et al., 2017), which rely on the stringent irrepresentability assumption (Zhao and Yu, 2006).

Using our rates of convergence for the SDD estimator, we also develop a valid inference procedure for elements of the true difference matrix. To this end, we leverage and extend the

general framework of Neykov et al. (2018) to allow for asymptotic distributions with arbitrary scaling of s_T where $s_T \to \infty$ instead of $\sqrt{T}$. We also address several additional challenges, which include establishing the joint asymptotic distribution of the entire spectral density estimator, proving the required concentration bounds under the appropriate choice of smoothing span, and finding a tractable estimator of the sparse projection direction, which is the inverse of the first derivative of the estimating equation. To address the first challenge, we leverage recent results from Zhang and Zhang (2025) to establish the form of the joint asymptotic distribution of the entire spectral density estimator. This is a crucial aspect of the inference procedure, as the asymptotic distribution of interest is a transformation of this distribution. Next, we use the flexibility of our new convergence rates that account for arbitrary smoothing spans to establish the necessary concentration bounds given the suitable choice of smoothing span for valid inference. Finally, we show that the asymmetric estimating equations for the difference in inverse spectral densities have favorable decompositions. Specifically, we show that the first derivative of these estimating equations decomposes into a Kronecker product of two smaller matrices, which allows us to generate a computationally tractable estimator of the sparse projection direction.

In addition to the theoretical challenges, we also address the computational challenges of developing an inference procedure for differential network analysis in the spectral domain, which necessitates novel computational tools. At a first glance, it may look as if our procedure calls for multiplying matrices of dimension $4p^2 \times 4p^2$—which, for $p = 100$, is $40,000 \times 40,000$, which are too large to even store in memory. To overcome these challenges, we implement sparse vector representations and intricate on-the-fly matrix multiplication in `C++` by carefully accounting for matrix indices.

After developing our SDD estimator and the corresponding inference procedure in Sections 3.2–3.6, we investigate its performance using simulated data in Section 3.7. We also illustrate its utility in a neuroscience application in Sections 3.8.

Notation. For a complex number $a \in \mathbb{C}$ let $\mathrm{Re}(a)$ and $\mathrm{Im}(a)$ be the real and imaginary parts of a. The absolute value of a complex number is defined as $|a| = \sqrt{\mathrm{Re}(a)^2 + \mathrm{Im}(a)^2}$. Similarly

for a complex matrix $A = (a_{ij}) \in \mathbb{C}^{m\times n}$ let $\mathrm{Re}(A)$ and $\mathrm{Im}(A)$ be the $\mathbb{R}^{m\times n}$ matrices of the real and imaginary components of A. The minimum eigenvalue of a matrix A is defined as $\lambda_{\min}(A)$. We define the ℓ_1, Frobenius, and infinity norms of a matrix A as $\|A\|_1 = \sum_{i,j} |a_{ij}|$, $\|A\|_F = \left(\sum_{i,j} |a_{ij}|^2\right)^{1/2}$, $\|A\|_\infty = \max_{i,j} |a_{ij}|$, respectively. We use $\|A\|_0$ to represent the ℓ_0 "norm" of A which counts the number of entries $|a_{ij}|$ that are non-zero. The element-wise absolute value of a matrix A is denoted as $|A|$ while the conjugate transpose of a matrix or vector will be denoted as A^H. We denote the inner product between two matrices A and B as $\langle A, B\rangle = \mathrm{Tr}(AB^T)$ and use $A \otimes B$ to represent their Kronecker product. The minimum and maximum of two real numbers x, y are defined as $x \vee y = \max(x, y)$ and $x \wedge y = \min(x, y)$ respectively. Throughout we will also use the notation $\log^a(x) := (\log(x))^a$. For random vectors and $q \geq 1$ we will denote the L_q norm as $\|X\|_{L_q} = \left(\mathbb{E}\,|X|^q\right)^{1/q}$. We also define $\mathcal{F}_t = (\epsilon_t, \epsilon_{t-1}, \dots)$ where $\epsilon_t, t \in \mathbb{Z}$ are i.i.d. random elements and $\mathrm{vec}\,(X)$ as the operator which stacks the columns of X. We write $x_n \xrightarrow{p} x$ and $x_n \xrightarrow{d} x$ to denote convergence of a random variable x_n to x in probability and distribution respectively. The notation $A_T \asymp B_T$ if A_T and B_T are asymptotically of the same order.

3.2 Direct Estimation of Differential Spectral Networks

Suppose $\{X_{l,t}\}$ is a p-dimensional mean zero stationary time series in condition l with autocovariance matrix $\Gamma_l(h) \in \mathbb{R}^{p\times p}$ defined as $\Gamma_l(h) = \mathbb{E}\left(X_{l,t}X_{l,t+h}^T\right)$ where $h \in \mathbb{Z}$. It is worth mentioning that we only require $\{X_{l,t}\}$ to be stationary in each condition and not across conditions which corresponds to assuming piecewise stationarity. In many applications, in particular in neuroscience, the change points are known and thus an assumption of piecewise stationarity is reasonable. For example in Yazdan-Shahmorad et al. (2016, 2018), the resting state and stimulation state blocks are known from the experimental design.

If $\sum_{h=-\infty}^{\infty} |\Gamma_l(h)| < \infty$, where the inequality is applied element-wise, then the spectral density in condition l at frequency λ exists and is defined as

$$f_l(\lambda) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} e^{-i\lambda h}\Gamma_l(h), \qquad -\pi \leq \lambda \leq \pi.$$

Since $f_l(\lambda)$ is complex-valued we can write $f_l(\lambda) = A_l(\lambda) + iB_l(\lambda)$ for some $A_l(\lambda), B_l(\lambda) \in \mathbb{R}^{p\times p}$.

In practice, many different λ values are of interest. For notational simplicity, however, we assume λ is fixed throughout and suppress the dependence on λ. We express the inverse of the spectral density as $f_l^{-1} = \tilde{A}_l + i\tilde{B}_l$ where $\tilde{A}_l$ and $\tilde{B}_l$ are matrices that represent the real and complex parts of f_l^{-1} respectively. Our goal is to estimate $f_1^{-1} - f_2^{-1}$. An alternative differential network of interest is the difference in partial spectral coherences from Dahlhaus (2000). The (i, j) entry of the partial spectral coherence represents the coherence between nodes i and j after removing the linear effects of all other nodes $\{k \neq i, j\}$ (Dahlhaus, 2000). As noted in Dahlhaus (2000), the inverse spectral density is a rescaling of the partial spectral coherence. The appropriate choice of scale is not straightforward and in this chapter we study the difference in inverse spectral densities.

While the spectral density f_l is complex valued, we can expand it to the real space by writing

$$\Sigma_l = \begin{bmatrix} A_l & -B_l \\ B_l & A_l \end{bmatrix},$$

where $\Sigma_l \in \mathbb{R}^{2p \times 2p}$. By Lemma A.1 of Fiecas et al. (2019) we have that

$$\begin{bmatrix} A_l & -B_l \\ B_l & A_l \end{bmatrix} \begin{bmatrix} \tilde{A}_l & -\tilde{B}_l \\ \tilde{B}_l & \tilde{A}_l \end{bmatrix} = I_{2p},$$

where I_{2p} represents the $2p \times 2p$ identity matrix. Thus, $f_1^{-1} - f_2^{-1}$ can be obtained by taking the (1,1) and (2,1) blocks of

$$\Delta^* = \Sigma_1^{-1} - \Sigma_2^{-1} = \begin{bmatrix} \tilde{A}_1 & -\tilde{B}_1 \\ \tilde{B}_1 & \tilde{A}_1 \end{bmatrix} - \begin{bmatrix} \tilde{A}_2 & -\tilde{B}_2 \\ \tilde{B}_2 & \tilde{A}_2 \end{bmatrix}.$$

Similarly, if we have an estimate $\hat{\Delta}$ of Δ^* available, we can estimate $f_1^{-1} - f_2^{-1}$ using the (1,1) and (2,1) blocks of $\hat{\Delta}$. To estimate Δ^*, we use the D-trace loss function from Yuan et al. (2017). That is, we use the loss

$$L_D(\Delta, \Sigma_2, \Sigma_1) = \frac{1}{4}\left(\langle \Sigma_2 \Delta, \Delta \Sigma_1 \rangle + \langle \Sigma_1 \Delta, \Delta \Sigma_2 \rangle\right) - \langle \Delta, \Sigma_2 - \Sigma_1 \rangle . \tag{3.1}$$

Taking the derivative with respect to Δ we see that Δ^* minimizes the D-trace loss. More specifically, let

$$\frac{\partial L_D}{\partial \Delta} = \frac{1}{2}\left(\Sigma_2 \Delta \Sigma_1 + \Sigma_1 \Delta \Sigma_2\right) - \left(\Sigma_2 - \Sigma_1\right) . \tag{3.2}$$

Evaluating $\frac{\partial L_D}{\partial \Delta}$ at Δ^* yields 0, establishing that Δ^* minimizes L_D. Furthermore, since Σ_1 and Σ_2 are positive definite, the Hessian of L_D with respect to Δ, $\frac{\partial^2 L_D}{\partial \Delta^2} = (\Sigma_1 \otimes \Sigma_2 + \Sigma_2 \otimes \Sigma_1)/2$, is positive; hence, L_D is convex in Δ, which implies that Δ^* is its unique minimizer.

In practice, the population quantites Σ_1 and Σ_2 are not available. Instead, we use the estimates $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$. To estimate the spectral density matrix f_l and in turn Σ_l we can use the smoothed periodogram. The periodogram is formed by taking the Fourier transform of the data. Suppose we have p-variate observations $\{X_{l,t}\}_{t=1,\ldots,T_l}$ for each condition l. For condition l, the periodogram at Fourier frequency $\{\lambda_j = 2\pi j/T_l, -\lfloor (T_l-1)/2 \rfloor \leq j \leq \lfloor T_l/2 \rfloor\}$ is defined as

$$P_l(\lambda_j) = \frac{1}{2\pi T_l} \left(\sum_{t=1}^{T_l} X_{l,t} \exp(-i\lambda_j t) \right) \left(\sum_{t=1}^{T_l} X_{l,t} \exp(-i\lambda_j t) \right)^H .$$

Using the fast Fourier transform algorithm, the periodogram can be computed quickly. While the periodogram is a natural estimate of f_l, it is well known to be inconsistent for the spectral density (Brockwell and Davis, 1991, Theorem 10.3.2). To remedy this, a smoothed version of the periodogram is used where the periodograms of $2B_l$ nearby Fourier frequencies are averaged. B_l is referred to as the bandwidth or smoothing span. That is, the smoothed periodogram at Fourier frequency λ_j is generated as

$$\hat{f}_l(\lambda_j) = \frac{1}{2B_l + 1} \sum_{k=j-B_l}^{j+B_l} P_l(\lambda_k).$$

We can then use $\hat{f}_l = \hat{A}_l + i\hat{B}_l$ to form $\hat{\Sigma}_l$

$$\hat{\Sigma}_l = \begin{bmatrix} \hat{A}_l & -\hat{B}_l \\ \hat{B}_l & \hat{A}_l \end{bmatrix}.$$

It is worth noting that Σ_l and $\hat{\Sigma}_l$ are symmetric. This fact will be used throughout the chapter. We generate our estimator by minimizing the D-trace loss in (3.1) with the estimators $\hat{\Sigma}_1, \hat{\Sigma}_2$. We can additionally incorporate an ℓ_1 penalty to estimate a sparse Δ. Our SDD estimator of Δ^* is then

$$\hat{\Delta} = \arg\min_{\Delta} \ \frac{1}{4} \left(\langle \hat{\Sigma}_2 \Delta, \Delta \hat{\Sigma}_1 \rangle + \langle \hat{\Sigma}_1 \Delta, \Delta \hat{\Sigma}_2 \rangle \right) - \langle \Delta, \hat{\Sigma}_2 - \hat{\Sigma}_1 \rangle + \tau_{T_1,T_2} \|\Delta\|_1 , \qquad (3.3)$$

where τ_{T_1,T_2} is a penalty parameter depending on sample sizes T_1, T_2. Typically τ_{T_1,T_2} is chosen by minimizing a criterion such as BIC or extended BIC (Foygel and Drton, 2010, eBIC). We use the same algorithm as in Yuan et al. (2017) to numerically solve for $\hat{\Delta}$.

3.3 Convergence rates of spectral density estimators

In this section, we develop a new convergence rate for the max norm of the class of smoothed spectral density estimators. That is, we develop a rate for

$$\max_{\lambda\in[0,2\pi]} \left\| \hat{f}_l(\lambda) - f_l(\lambda) \right\|_\infty . \tag{3.4}$$

This convergence rate is an essential ingredient for our theoretical analyses, from proving consistency of SDD in Theorem 4 to verifying concentration bounds in our inference procedure in Appendix B.5. For example, examining the consistency proofs reveals that the convergence rate of SDD depends on the convergence rate of the max norm of the spectral density estimators in each condition.

One approach to derive the convergence rates is to fix the smoothing span at some, potentially optimal, value. For example, Zhang and Zhang (2025) choose B to balance the stochastic deviation and the bias in their Theorem 1 and Proposition 2, which leads to the recommended 'optimal' smoothing span of order $B \asymp (T/(\log^{4\alpha+2}(p \vee T)))^{1/3}$, where α is a measure of dependence in the data. However, as stated in Theorem 3 of Zhang and Zhang (2025) and noted in our theoretical results in Section 3.5, in order for the spectral density estimator to be asymptotically unbiased when scaled by $\sqrt{T/B}$, the smoothing span needs to satisfy $T\Phi_2^4 = o(B^3)$. When the smoothing spans are smaller, as is the case for the optimal rate of $(B \asymp T/(\log^{4\alpha+2}(p \vee T)))^{1/3}$ in Zhang and Zhang (2025), the bias is no longer negligible and the asymptotic distribution of the spectral density estimator is instead centered around its expected value. Therefore, the smoothing span suggested by Zhang and Zhang (2025) does not lead to calibrated inference, as we require an asymptotic distribution centered around the true spectral density. This motivates us to consider convergence rates that allow for flexible smoothing spans.

Although other results, such as Theorem 3.1 in Fiecas et al. (2019), can be used to solve for

an optimal smoothing span B and corresponding convergence rate, we opt to utilize the results in Zhang and Zhang (2025), which are more general and allow for a wider class of dependencies by characterizing the data dependence using the functional dependence framework first conceptualized in Wu (2005). Throughout, we will add a subscript l to make it clear that the dependence can vary between conditions $l \in \{1, 2\}$. We also assume that $X_{l,t}$ is a p-dimensional random variable, i.e., $X_{l,t} = (X_{l,t1}, \ldots, X_{l,tp})$ and that $\max_{1 \le j \le p} \|X_{l,tj}\|_{L_q} < \infty$ for some $q \ge 2$. In the functional dependence framework $X_{l,tj} = g_{l,j}(\epsilon_{l,t}, \epsilon_{l,t-1}, \ldots)$ where $g_{l,j}$ is a measurable function and $\epsilon_{l,t}, t \in \mathbb{Z}$ are i.i.d random variables. For $t \ge 0$ and $1 \le j \le p$ the functional dependence measure is defined as

$$\delta_{l,t,q,j} = \left\| g_{l,j}(\epsilon_{l,t}, \epsilon_{l,t-1}, \ldots) - g_{l,j}(\epsilon_{l,t}, \ldots, \epsilon_{l,1}, \epsilon^*_{l,0}, \epsilon_{l,-1}, \ldots) \right\|_{L_q} ,$$

where $\epsilon^*_{l,0}$ is an i.i.d. copy of $\epsilon_{l,0}$. The process $g_{l,j}(\epsilon_{l,t}, \ldots, \epsilon_{l,1}, \epsilon^*_{l,0}, \epsilon_{l,-1}, \ldots)$ is equivalent to $X_{l,tj}$ except $\epsilon_{l,0}$ is replaced by $\epsilon^*_{l,0}$. In this way, $\delta_{l,t,q,j}$ measures the effect that an innovation t time points in the past has on $X_{l,tj}$. The q-th dependence adjusted moment is defined as follows. Assume there exists $\rho \in (0, 1)$ such that

$$\|X_{l,\cdot j}\|_{L_q} = \sup_{m \ge 0} \rho^{-m} \Delta_{l,m,q,j} < \infty, \text{where } \Delta_{l,m,q,j} = \sum_{t=m}^{\infty} \delta_{l,t,q,j} .$$

As noted in Zhang and Zhang (2025), $\|X_{l,\cdot j}\|_{L_q}$ — which they denote $\|X_{l,\cdot j}\|_q$ — can be interpreted as the q-th moment of the process $X_{l,\cdot j}$ taking dependence into account. By rearranging the inequality to get $\Delta_{l,m,q,j} \le \|X_{l,\cdot j}\|_{L_q} \rho^m$ for all $m \ge 0$, we can see that this definition requires the tail dependence $\Delta_{l,m,q,j}$ to decay geometrically.

We next state our (only) dependence assumption. This assumption is the same as Assumption 1 in Zhang and Zhang (2025).

Assumption 1. *There exists some constant $\alpha_l \ge 0$ such that*

$$\|X_{l,\cdot j}\|_{\psi_{\alpha_l}} := \sup_{q \ge 2} q^{-\alpha_l} \|X_{l,\cdot j}\|_{L_q} < \infty .$$

In Assumption 1, α_l represents the dependence in the process for condition l. As noted in Zhang and Zhang (2025), when the data is i.i.d $\alpha_l = 1/2$ corresponds to the classical sub-Gaussian norm while $\alpha_l = 1$ corresponds to the sub-Exponential norm. The larger α_l is, the

heavier the tails. As stated above, the optimal smoothing span for condition l derived from Zhang and Zhang (2025) is $B_l \asymp \left(T_l/\log^{4\alpha_l+2}(p \vee T_l)\right)^{1/3}$ which is too slow as inference requires $T_l\Phi_2^4 = o(B_l^3)$. Thus, we consider smoothing spans with arbitrary exponents, i.e., B_l has the form $B_l \asymp \left(T_l/\log^{4\alpha_l+2}(p \vee T_l)\right)^{\gamma_l}$ for $\gamma_l > 0$. We will specifically focus on exponents $\gamma_l > 1/3$ as this allows us to use our asymptotic distribution results, but for full generality we present the convergence rates for this general class of smoothing spans, $0 < \gamma_l < 1$, in Theorem 3. To simplify the statement of convergence rates, we denote $\Phi_{l,q} = \max_{1\le j\le p} \left\|X_{l,\cdot j}^2\right\|_{L_q}$ and $\|X_{l,\cdot}\|_{\psi_{\alpha_l}} = \max_{1\le j\le p} \|X_{l,\cdot j}\|_{\psi_{\alpha_l}}$.

Theorem 3. *Suppose Assumption 1 is satisfied for condition l and some $\alpha_l > 0$. Then for $B_l \asymp \left(\frac{T_l}{\log^{4\alpha_l+2}(p\vee T_l)}\right)^{\gamma_l}$,*

$$\max_{\lambda\in[0,2\pi]} \left\|\hat{f}_l(\lambda) - f_l(\lambda)\right\|_\infty = \begin{cases} O_p\left(2^{\alpha_l+1}\|X_{l,\cdot}\|_{\psi_{\alpha_l}}^2 T_l^{\frac{\gamma_l-1}{2}} \log^{(1-\gamma_l)(2\alpha_l+1)}(p\vee T_l)\right) & \text{for } \gamma_l \ge 1/3 \\ O_p\left(\Phi_{l,2}^2 T_l^{-\gamma_l}\log^{\gamma_l(4\alpha_l+2)}(p\vee T_l)\right) & \text{for } \gamma_l < 1/3 \end{cases}$$

The proof of Theorem 3 is given in Appendix B.1. The two rates correspond to whether the stochastic term or the bias is dominant. When $\gamma_l \ge 1/3$ the stochastic term is slower and thus determines the rate. When $\gamma_l < 1/3$, the bias term is slower. With Theorem 3, we now have access to the convergence rate for $\max_{\lambda\in[0,2\pi]} \left\|\hat{f}_l(\lambda) - f_l(\lambda)\right\|_\infty$ when we choose B_l to be large enough to perform inference. It is also worth pointing out that while we focus on the simple averaged periodogram, Theorem 3 also applies to general kernel estimators of the spectral density as in Wu and Zaffaroni (2018). That is, Theorem 3 also applies to kernel estimators of the form

$$\hat{\hat{f}}_l(\lambda_j) = \frac{1}{2\pi T_l}\sum_{k=-B_l}^{B_l} K(k/B_l)\hat{\Gamma}_l(k)e^{-ik\lambda_j},$$

where $K(\cdot)$ is a continuous, symmetric, and bounded kernel on $[-1,1]$ that satisfies $K(0) = 1$ and $\hat{\Gamma}_l(k) = T_l^{-1}\sum_{i=k+1}^{T_l} X_{l,i-k}X_{l,i}^T$ for $k \ge 0$. For $k < 0$ we can use that $\hat{\Gamma}(k) = \hat{\Gamma}(-k)$. Thus, all results presented are also valid for these spectral density estimators.

Next, we provide some examples of convergence rates in the high-dimensional setting $(p > T_l)$ for different choices of γ_l and different assumptions on α_l.

Example 1. We first consider the rate we would obtain if we use the bandwidth suggested by Zhang and Zhang (2025). If we choose $\gamma_l = 1/3$, then the rate in Theorem 3 is $O_p\left(\frac{\log^{(4\alpha_l+2)/3}(p)}{T_l^{1/3}}\right)$.

Example 2 (sub-Gaussian, $\gamma_l = 1/2$). Suppose we choose $\gamma_l = 1/2$ and consider $\alpha_l = 1/2$. The convergence rate is then $O_p\left(\frac{\log(p)}{T_l^{1/4}}\right)$.

Example 3 (sub-Gaussian, $\gamma_l = 3/4$). Consider the same setting as in the previous example, but instead suppose we choose $\gamma_l = 3/4$. The convergence rate becomes $O_p\left(\frac{\sqrt{\log(p)}}{T_l^{1/8}}\right)$.

Example 4 (sub-Exponential). Suppose we choose $\gamma_l = 1/2$ and consider $\alpha_l = 1$. The convergence rate then becomes $O_p\left(\frac{\sqrt{\log^3(p)}}{T_l^{1/4}}\right)$.

As we can see from the examples, larger dependence in the data increases the exponent in $\log(p)$ while larger smoothing spans (i.e., larger γ_l) decreases the exponent a in $\frac{1}{T_l^a}$ and decreases the exponent in $\log(p)$. In other words, larger γ_l makes the convergence rate slower in T_l but also slower in $\log(p)$. With the convergence rates in Theorem 3, we can now establish the consistency of our SDD estimator (3.3) allowing for flexible data dependence and smoothing spans.

3.4 Consistency

In this section, we establish the consistency of $\hat{\Delta}$ to the population quantity Δ^*, allowing for different levels of data dependence and smoothing spans. Although we use an algorithm similar to Yuan et al. (2017) to numerically solve for $\hat{\Delta}$, our theoretical analysis is fundamentally different. Specifically, the theoretical results in Yuan et al. (2017) hinge on an stringent irrepresentability assumption (Zhao and Yu, 2006). This assumption may be especially unrealistic in neuroscience applications, due to the presence of hub nodes, or regions of the brain that are connected to many other brain regions (Buckner et al., 2009; Wang et al., 2021). Moreover, simulations in Wang et al. (2021) using Erdős-Rényi graphs of moderate to large dimensions—e.g., $p > 20$—show that the irrepresentability condition almost never holds, even with very sparse differences and weak data dependence. In contrast, our analysis combines recent theoretical advances in time series data

analysis (Zhang and Zhang, 2025) with more flexible proof techniques (Wang et al., 2021; Negahban et al., 2012) to establish the consistency under mild assumptions. More specificaly, combining these techniques with our new convergence rate in Theorem 3, we establish the convergence rate of SDD only assuming that the true difference in inverse spectral densities is sparse, while flexibly allowing for varying data dependence and smoothing spans of the spectral density estimators. The assumption of sparsity of the difference is more realistic if, for example, the network does not change much between conditions. For instance, despite identifying many nonzero coherence values in resting state and stimulation states, Bloch et al. (2022) found many coherence changes to be near zero. These small differences are likely due to noise in the estimation procedure, corresponding to no underlying change in coherence. An important aspect of our theoretical analysis is that by only assuming the sparsity of the difference, we allow each individual network to be dense.

It is also worth noting that we do not need any parametric assumptions on the data generating processes. Instead, we only require mild assumptions on the data dependence for each condition. The only restriction on the data generating process is that Assumption 1 is satisfied for both conditions.

It is seen in Equation 3.3 that our estimator $\hat{\Delta}$ relies on expanded estimates of the spectral density $\hat{\Sigma}_1, \hat{\Sigma}_2$. Thus, the rate of convergence of $\hat{\Delta}$ to Δ^* naturally relies on the rates of convergence of $\hat{f}_1$ and $\hat{f}_2$ to f_1 and f_2. We will define the number of non-zero entries of Δ^* as $s_{\Delta^*} := \|\Delta^*\|_0$ By Theorem 3, the convergence rates for $\max_{\lambda\in[0,2\pi]} \left\|\hat{f}_l(\lambda) - f_l(\lambda)\right\|_\infty$ for $l \in \{1, 2\}$ follow from Assumption 1, which is all we need to establish the consistency of $\hat{\Delta}$.

Theorem 4. *Suppose there exists constants α_1, α_2 such that Assumption 1 is satisfied for conditions 1 and 2, respectively. Furthermore, suppose the smoothing span for condition l is $B_l \asymp \left(\frac{T_l}{\log^{4\alpha_l+2}(p\vee T_l)}\right)^{\gamma_l}$. Then, for $\tau_{T_1,T_2} \geq C_f O_p\left(R_{T_1,p} + R_{T_2,p}\right)\left(1 + \|\Delta^*\|_1\right)$ and large enough T_1, T_2, with high probability*

$$\|\hat{\Delta} - \Delta^*\|_F = O_p\left(\sqrt{s_{\Delta^*}}\left(R_{T_1,p} + R_{T_2,p}\right) C_f\left(1 + \|\Delta^*\|_1\right)\right),$$
$$\|\hat{\Delta} - \Delta^*\|_1 = O_p\left(s_{\Delta^*}\left(R_{T_1,p} + R_{T_2,p}\right) C_f\left(1 + \|\Delta^*\|_1\right)\right),$$

where $R_{T_1,p}, R_{T_2,p}$ are the convergence rates from applying Theorem 3 to conditions 1 and 2

respectively, $\|\Delta^*\|_1$ *is the* ℓ_1 *norm of the true difference matrix* Δ^**, and* $C_f = \max(\|f_1\|_\infty, \|f_2\|_\infty)$.

The proof of Theorem 4 is given in Appendix B.2. The rates in Theorem 4 show that the convergence depends on the convergence rates of the maximum deviations of the estimated spectral density from the true spectral density and the sparsity of the true difference.

Remark 1. The convergence rate, $O_p\left(R_{T_1,p} + R_{T_2,p}\right)$, can be further simplified to $O_p\left(\max\left(R_{T_1,p}, R_{T_2,p}\right)\right)$ by noting that $R_{T_1,p} + R_{T_2,p} \leq 2\max\left(R_{T_1,p}, R_{T_2,p}\right)$. This makes it more clear that the convergence rate of our SDD estimator is limited by the slower of the two convergence rates in conditions 1 and 2.

Remark 2. Recall that, for notational convenience, we have suppressed the dependence on frequency, λ. Since Theorem 3 applies over all $\lambda \in [0, 2\pi]$, these rates hold for every λ. However, some of the constants will differ based on λ as Δ^* and f_l depend on λ. Specifically, constants such as $\|\Delta^*\|_1$, $\lambda_{\min}(f_l)$, C_f, and s_{Δ^*} can change depending on the λ of interest.

Remark 3. Our SDD method can also be extended to directly estimate a difference in different frequencies, λ_1, λ_2 for the same condition l. In this case, we only require Assumption 1 to hold for the condition of interest l. The same proof techniques can be applied and the resulting rates will be $R_{T_l,p}$ instead of $R_{T_1,p} + R_{T_2,p}$. It is worth stating that for this analysis, Δ^* will be the difference in the expanded spectral densities between frequencies λ_1, λ_2. Thus, this analysis will implicitly assume that the difference between frequencies is sparse.

3.5 Inference

3.5.1 Theory

In this section, we develop theory to generate confidence intervals and perform inference for arbitrary entries of $\mathrm{vec}\left(\Delta^*\right)$ by leveraging the de-biasing framework of Neykov et al. (2018). To adopt this framework to our problem setting, we address several challenges, including the need for a computationally convenient estimating equation, arbitrary scaling of the asymptotic distribution, and

the joint asymptotic distribution of the spectral density estimator. In the remainder of this section, we review this framework and present new results to overcome the above challenges.

Suppose we have an estimating equation $\mathbf{t}\left(Z_T, \beta\right)$ where Z_T is the data and $\beta \in \mathbb{R}^q$ is a parameter. For ease of notation we will denote Z_T as Z. Without loss of generality, we will assume the target of inference is the first entry in β which we will denote as θ and $\beta = (\theta, \gamma)$ where γ is the vector of nuisance parameters. The limiting expected value of the estimating equation is denoted as $E_{\mathbf{t}}(\beta) = \lim_{T\to\infty} \mathbb{E}\mathbf{t}(Z, \beta)$; we assume that the true parameter of interest is the unique solution to $E_{\mathbf{t}}(\beta) = 0$. We will also denote the derivative of the estimating equation as $\mathbf{T}\left(Z, \beta\right) = \frac{\partial}{\partial\beta}\mathbf{t}\left(Z, \beta\right)$ and its limiting expectation as $E_{\mathbf{T}}(\beta) = \lim_{T\to\infty} \mathbb{E}\mathbf{T}(Z, \beta)$. In low-dimensional settings, an estimate of β, $\hat{\beta} = (\hat{\theta}, \hat{\gamma})$ can be obtained by solving $\mathbf{t}(Z, \beta) = 0$. Appealing to classical theory, the following expansion is obtained:

$$\sqrt{T}\left(\hat{\theta} - \theta^*\right) = -\sqrt{T}\left[E_{\mathbf{T}}(\beta^*)\right]^{-1}_{1*}\mathbf{t}(Z, \beta^*) + o_p(1)\,, \tag{3.5}$$

where $[X]_{i*}$ selects row i of X. Typically, it can be shown that the right-hand-side of (3.5) converges to a normal distribution which gives an asymptotic normal distribution for the left-hand-side. In high-dimensions, sparsity assumptions on β and constrained estimation approaches are necessary. Sparsity-inducing regularization leverages this assumption but induces bias in $\hat{\beta}$ and the expansion in (3.5) is no longer valid (Javanmard and Montanari, 2014). The asymptotic distribution of $\hat{\beta}$ must thus be derived using alternative approaches. Neykov et al. (2018) offer a general framework to solve this problem by directly estimating $\mathrm{v}^* = \left[E_{\mathbf{T}}(\beta^*)\right]^{-1}_{1*}$ on the right-hand-side of (3.5) as

$$\hat{\mathrm{v}} = \arg\min \|\mathrm{v}\|_1 \text{ subject to } \left\|\mathrm{v}^T\mathbf{T}(Z, \hat{\beta}) - e_1\right\|_\infty \leq \rho\,, \tag{3.6}$$

where $e_1 = (1, 0, \ldots, 0)$ is a q dimensional row vector. Neykov et al. (2018) then propose to estimate θ by projecting the estimating equation on the sparse direction $\hat{\mathrm{v}}$, plugging in estimates of nuisance parameters γ, and solving the new projected estimating equation. Denoting $\hat{S}(\hat{\beta}_\theta) := \hat{\mathrm{v}}^T\mathbf{t}(Z, \hat{\beta}_\theta)$, the de-biased estimate of θ is defined as

$$\tilde{\theta} := \theta : \hat{S}(\hat{\beta}_\theta) = 0\,,$$

where $\hat{\beta}_\theta = (\theta, \hat{\gamma})$[1]; the framework provides conditions under which $\tilde{\theta}$ is asymptotically normal which allows for valid statistical inference.

The first step in applying this inferential framework is choosing an estimating equation for our parameter of interest $\Delta^* = \Sigma_1^{-1} - \Sigma_2^{-1}$. For simplicity, we assume throughout that $T_1 = T_2 = T$ but note that all results should still hold as long as $T_1 \asymp T_2$. Based on our SDD estimator, a natural choice of estimating equation is the derivative of the D-trace loss (3.2). We define

$$\begin{aligned} \mathbf{t}_{\text{symm}}(Z, \Delta) &:= \frac{1}{2}\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right) \text{vec}\,(\Delta) - \text{vec}\left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right) \\ &= \text{vec}\left(\frac{1}{2}\left(\hat{\Sigma}_2 \Delta \hat{\Sigma}_1 + \hat{\Sigma}_1 \Delta \hat{\Sigma}_2\right) - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right). \end{aligned} \tag{3.7}$$

Recall that $\text{vec}\,(X)$ converts $X \in \mathbb{R}^{m \times n}$ to a vector in $\mathbb{R}^{mn}$ by stacking its columns. The subscript 'symm' indicates that this is the symmetric D-trace loss estimating equation. Assuming data in conditions 1 and 2 are independent, then if the conditions of Theorem 3 are satisfied, $\hat{\Sigma}_l$ are asymptotically unbiased by Corollary 3 and Δ^* is the solution to $E_{\mathbf{t}_{\text{symm}}}(\beta) = 0$ (Zhang and Zhang, 2025).

One problem that arises immediately when using this estimating equation is the need to obtain $\hat{\text{v}}_{\text{symm}}$ in (3.6) which, for this choice of estimating equation, requires forming $\mathbf{T}_{\text{symm}}(Z, \hat{\Delta}) = \frac{1}{2}\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right)$. Since $\hat{\Sigma}_l \in \mathbb{R}^{2p \times 2p}$, then $\mathbf{T}_{\text{symm}}(Z, \hat{\Delta}) \in \mathbb{R}^{4p^2 \times 4p^2}$. In high-dimensions, this matrix will be impossibly large. For example for $p = 100$, this matrix is $40,000 \times 40,000$ which is too large to even store in memory for most personal computers, let alone invert. We seek other estimating equations for Δ^* where it is computationally feasible to estimate $\hat{\text{v}}$.

The two alternative estimating equations we consider are each portion of $\mathbf{t}_{\text{symm}}(Z, \Delta)$:

$$\mathbf{t}_{\text{s1Left}}(Z, \Delta) := \left(\hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right) \text{vec}\,(\Delta) - \text{vec}\left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right) = \text{vec}\left(\left(\hat{\Sigma}_1 \Delta \hat{\Sigma}_2\right) - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right) \tag{3.8}$$

$$\mathbf{t}_{\text{s1Right}}(Z, \Delta) := \left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2\right) \text{vec}\,(\Delta) - \text{vec}\left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right) = \text{vec}\left(\left(\hat{\Sigma}_2 \Delta \hat{\Sigma}_1\right) - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right). \tag{3.9}$$

[1]It is possible for $\tilde{\theta}$ to be set-valued, i.e. $\tilde{\theta} = \left\{\theta : \hat{S}(\hat{\beta}_\theta) = 0\right\}$. However, a key requirement for inference in Theorem 5 is the uniqueness of $\tilde{\theta}$.

The subscripts 's1Left' and 's1Right' indicate whether we use the part of the symmetric estimating equation, $\mathbf{t}_{\text{symm}}(Z, \Delta)$, where s1 or $\hat{\Sigma}_1$ is on the left or the right of Δ. The equations at the end of (3.8, 3.9) make this naming convention more clear. Again, assuming data in conditions 1 and 2 are independent, Δ^* is the solution to $E_{\mathbf{t}_{\text{s1Left}}}(\beta) = 0$ and $E_{\mathbf{t}_{\text{s1Right}}}(\beta) = 0$. While $\mathbf{T}_{\text{s1Left}}(Z, \hat{\Delta}) = \hat{\Sigma}_2 \otimes \hat{\Sigma}_1, \mathbf{T}_{\text{s1Right}}(Z, \hat{\Delta}) = \hat{\Sigma}_1 \otimes \hat{\Sigma}_2 \in \mathbb{R}^{4p^2 \times 4p^2}$, and are similarly too large to store in memory, their forms allow us to leverage the inverse of Kronecker product of matrices, $(X \otimes Y)^{-1} = X^{-1} \otimes Y^{-1}$. Specifically, if we have estimates of Σ_l^{-1}, $\hat{\Sigma}_l^{-1}$, available, then $\hat{\text{v}}_{\text{s1Left}}$ can be generated as $\left[\hat{\Sigma}_2^{-1} \otimes \hat{\Sigma}_1^{-1}\right]_{1*} = \hat{\Sigma}_{2,1*}^{-1} \otimes \hat{\Sigma}_{1,1*}^{-1}$. Thus we only need to form a Kronecker product between two $2p$ vectors, which requires storing a $4p^2$ vector. Since $\Sigma_1, \Sigma_2 \in \mathbb{R}^{2p \times 2p}$, estimating their inverses is computationally feasible even for large p and can be computed efficiently using methods such as the graphical LASSO or CLIME (Friedman et al., 2008; Cai et al., 2011). In fact, with $\hat{\Sigma}_1^{-1}, \hat{\Sigma}_2^{-1}$ we can generate e.g. $\hat{\text{v}}_{\text{s1Left}}$ for any component of Δ^*. For example, suppose we wish to obtain a confidence interval for the j-th entry of $\text{vec}(\Delta^*)$, an estimate of $\left[\Sigma_2^{-1} \otimes \Sigma_1^{-1}\right]_{j*}$ can be formed by $\left[\hat{\Sigma}_2^{-1} \otimes \hat{\Sigma}_1^{-1}\right]_{j*}$ which is formed by the Kronecker product of the relevant rows of $\hat{\Sigma}_2^{-1}$ and $\hat{\Sigma}_1^{-1}$.

Given the above estimating equations for which it is computationally tractable to carry out the inference procedure, the next step is to prove that assumptions in Neykov et al. (2018) hold under our general dependence setting for high-dimensional time series. At the heart of the inference procedure is the asymptotic normality of the de-biased estimating equation (Assumption 3 in Neykov et al. (2018)). Specifically, a key assumption is that e.g. $\text{v}_{\text{s1Left}}^* \mathbf{t}_{\text{s1Left}}(Z, \beta^*)$ is asymptotically normal with a known variance when appropriately scaled. Here we use s1Left as an example estimating equation, but note that this condition must hold when conducting inference for each of s1Left, s1Right, and symm. For all three of our estimating equations, the asymptotic distribution is a linear transformation of the asymptotic distribution of $\left(\text{vec}(\hat{A}), \text{vec}(\hat{B})\right)$. Thus, to show this assumption, we require the asymptotic distribution of $\left(\text{vec}(\hat{A}), \text{vec}(\hat{B})\right)$. Although the asymptotic distribution of any two elements of the spectral density matrix is given in Zhang and Zhang (2025), the form of the joint distribution is not available. We derive this form in Proposition 1, which is proved in Appendix B.3.

Proposition 1. *Denote* $\Phi_4 = \max_{1\leq j\leq p} \|X_{\cdot j}\|_4$. *Suppose Assumption 1 is satisfied,* $B/T \to 0$, $\Phi_4^8 (\log B)/B \to 0$ *and* $T\Phi_2^4 = o(B^3)$. *Denote* $\hat{f}(\lambda) = \hat{A}(\lambda) + i\hat{B}(\lambda)$ *and* $f(\lambda) = A(\lambda) + iB(\lambda)$. *Define* $K = \sum_{i,j=1}^{p} H_{ij} \otimes H_{ij}^T$ *where* H_{ij} *is the* $p \times p$ *matrix with* $h_{ij} = 1$ *and 0 elsewhere.*

(i) If $\lambda \in (0, \pi)$, *then*

$$\sqrt{\frac{T}{B}} \begin{pmatrix} \text{vec}(\hat{A}(\lambda)) - \text{vec}(A(\lambda)) \\ \text{vec}(\hat{B}(\lambda)) - \text{vec}(B(\lambda)) \end{pmatrix} \xrightarrow{d} N(0, V_f),$$

where

$$V_f = \tfrac{1}{2} \begin{bmatrix} (I_{p^2} + K)(A(\lambda) \otimes A(\lambda) + B(\lambda) \otimes B(\lambda)) & (I_{p^2} + K)(B(\lambda) \otimes A(\lambda) - A(\lambda) \otimes B(\lambda)) \\ [(I_{p^2} + K)(B(\lambda) \otimes A(\lambda) - A(\lambda) \otimes B(\lambda))]^T & (I_{p^2} - K)(A(\lambda) \otimes A(\lambda) + B(\lambda) \otimes B(\lambda)) \end{bmatrix}.$$

(ii) If $\lambda = 0$ *or* π *then*

$$\sqrt{\frac{T}{B}} \left(\text{vec}(\hat{A}(\lambda)) - \text{vec}(A(\lambda)) \right) \xrightarrow{d} N(0, V_f),$$

where

$$V_f = (I_{p^2} + K) A(\lambda) \otimes A(\lambda).$$

As discussed in Section 3.3, a requirement of Proposition 1 is that $T\Phi_2^4 = o(B^3)$. To satisfy this requirement for each condition $l \in \{1, 2\}$ we can choose $B_l \asymp \left(T_l / \log^{4\alpha_l+2}(p \vee T_l)\right)^{\gamma_l}$ for $\gamma_l > 1/3$. In practice, the dependence in the data α_l is unknown so we typically choose $B_l \asymp T^{2/3}$ or $B_l \asymp T^{1/2}$. Using the results in Theorem 3, with these choices of smoothing spans we know the convergence rates of $\hat{f}_l$ to f_l; this subsequently give the rates of convergence of SDD by Theorem 4 which are used to prove other technical conditions in establishing the inference procedure. See for example Appendix B.5 and Lemma 10.

Note that due to smoothing of the spectral density estimators, the scaling of the asymptotic distribution of the spectral density estimators as well as the de-biased estimating equations will be $\sqrt{T/B}$. Thus, we cannot proceed with the original framework in Neykov et al. (2018), as it assumes $\sqrt{T}$ scaling. However, we can extend this framework to allow for asymptotic distributions with general scaling s_T where it is assumed $s_T \overset{T\to\infty}{\to} \infty$. Although most of the results generally follow

directly from Neykov et al. (2018), for completeness the full set of assumptions and derivations are available in Appendix B.4. The major changes include an arbitrary scaling of s_T in the asymptotic distribution of Assumption 7 and a change from $\sqrt{T}$ to s_T scaling in Assumption 9. Assumption 9 is used to ensure that the second order terms are negligible when scaling by s_T. With these modifications we have

Theorem 5. *Let the map $\theta \mapsto \hat{S}\left(\hat{\beta}_\theta\right)$ be continuous with a single root $\tilde{\theta}$ or non-decreasing. Further, suppose that for any $\epsilon > 0$*

$$\mathrm{v}^{*T}\left[E_{\mathbf{t}}\left(\beta^*_{\theta^*-\epsilon}\right)\right]\mathrm{v}^{*T}\left[E_{\mathbf{t}}\left(\beta^*_{\theta^*+\epsilon}\right)\right] < 0.$$

If $\hat{\sigma}^2$ is a consistent estimator of $\sigma^2 := \mathrm{v}^{*T} G \mathrm{v}^*$ and $G := \lim_{T\to\infty} s_T^2 \operatorname{Cov}\left(\mathbf{t}(Z, \beta^*)\right)$ and Assumptions 5, 6, 7, 8, 9 hold, then for any $x \in \mathbb{R}$, we have:

$$\lim_{T\to\infty}\left|\mathbb{P}(\hat{U}_T \le x) - \mathbf{\Phi}(x)\right| = 0, \qquad \textit{where } \hat{U}_T = \frac{s_T}{\hat{\sigma}}(\tilde{\theta} - \theta^*)$$

and $\mathbf{\Phi}$ is the CDF of a standard normal random variable.

Theorem 5 generalizes the result in Neykov et al. (2018) and states that the appropriately scaled quantity $\frac{s_T}{\hat{\sigma}}(\tilde{\theta} - \theta^*)$ converges to a standard normal distribution. The proof of Theorem 5 is available in Appendix B.4. Now that we have addressed all the challenges with inference we can formally discuss the procedure.

3.5.2 Procedure

We next establish the inference procedure for each of the three estimating equation types: symm, s1Left, s1Right. Specifically, we first present details of the inference procedures and then prove their theoretical validity in Theorem 6. The de-biased $\tilde{\theta}$ for each estimating equation is given as

$$\begin{aligned}
\tilde{\theta}_{\text{symm}} &:= \theta : \hat{\mathrm{v}}^T_{\text{symm}} \mathbf{t}_{\text{symm}}(Z, \hat{\beta}_\theta) = 0 \\
\tilde{\theta}_{\text{s1Left}} &:= \theta : \hat{\mathrm{v}}^T_{\text{s1Left}} \mathbf{t}_{\text{s1Left}}(Z, \hat{\beta}_\theta) = 0 \\
\tilde{\theta}_{\text{s1Right}} &:= \theta : \hat{\mathrm{v}}^T_{\text{s1Right}} \mathbf{t}_{\text{s1Right}}(Z, \hat{\beta}_\theta) = 0\,,
\end{aligned}$$

where, as before, $\hat{\beta}_\theta = (\theta, \hat{\gamma})$ and $\hat{\gamma}$ are estimated nuisance parameters and can be obtained from the relevant entries of our SDD estimator $\hat{\Delta}$. Confidence intervals can be constructed using

$$\begin{aligned}
\hat{U}_{\text{symm}} &= \frac{1}{\hat{\sigma}_{\text{symm}}} \sqrt{\frac{T}{B}} \left(\tilde{\theta}_{\text{symm}} - \theta^* \right) \\
\hat{U}_{\text{s1Left}} &= \frac{1}{\hat{\sigma}_{\text{s1Left}}} \sqrt{\frac{T}{B}} \left(\tilde{\theta}_{\text{s1Left}} - \theta^* \right) \\
\hat{U}_{\text{s1Right}} &= \frac{1}{\hat{\sigma}_{\text{s1Right}}} \sqrt{\frac{T}{B}} \left(\tilde{\theta}_{\text{s1Right}} - \theta^* \right) ,
\end{aligned}$$

The forms of $\hat{\sigma}_{\text{symm}}, \hat{\sigma}_{\text{s1Left}}, \hat{\sigma}_{\text{s1Right}}$ involve a lengthy linear transformation and thus are stated in Lemma 10 which proves the consistency of each. The form of $\hat{\sigma}_{\text{s1Left}}$ can also be found in Section 3.6 where we study a computational example. Theorem 6 establishes the asymptotic normality of $\hat{U}_{\text{symm}}, \hat{U}_{\text{s1Left}}, \hat{U}_{\text{s1Right}}$, which suggests that each of these statistics can be used to generate asymptotic confidence intervals for a parameter of interest θ^*.

Theorem 6. *Suppose the conditions from Theorem 4 and Proposition 1 hold and*

$$\begin{aligned}
\rho_{\text{symm}} &= \|\text{v}_{\text{symm}}\|_1 \, C_f O_p(R_{T_1,p} + R_{T_2,p}) \\
\rho_{\text{s1Left}} &= \|\text{v}_{\text{s1Left}}\|_1 \, C_f O_p(R_{T_1,p} + R_{T_2,p}) \\
\rho_{\text{s1Right}} &= \|\text{v}_{\text{s1Right}}\|_1 \, C_f O_p(R_{T_1,p} + R_{T_2,p}) \, .
\end{aligned}$$

Then, we have

$$\begin{aligned}
\hat{U}_{\text{symm}} &\xrightarrow[T_1,T_2\to\infty]{d} N(0,1) \\
\hat{U}_{\text{s1Left}} &\xrightarrow[T_1,T_2\to\infty]{d} N(0,1) \\
\hat{U}_{\text{s1Right}} &\xrightarrow[T_1,T_2\to\infty]{d} N(0,1) \, .
\end{aligned}$$

The ρ_{type} in Theorem 6, the forms of which are derived in Appendix B.5, are the penalty parameters used in estimating the sparse projection direction as in (3.6) for the respective estimating equations. For instance, ρ_{symm} is the penalty parameters for estimating $\hat{\text{v}}_{\text{symm}}$ from the relevant formulation of (3.6). Recall also that τ_{T_1,T_2} is the penalty parameter for our SDD estimator in (3.3).

Given these quantities, Theorem 6 provides valid inference only using the assumptions of Theorem 4 and Proposition 1 which are very mild. Specifically, recall that Theorem 4 only requires a mild data dependence assumption for each condition and sparsity of the true difference. Meanwhile, for Proposition 1 we only need a finite moment condition as well as conditions on the rate of smoothing spans B_l relative to the sample sizes T_l. Importantly, by Theorem 6 we can perform inference without any parametric assumptions on the data generating process.

3.6 Computational challenges

In this section, we address various computational challenges that arise when implementing our methods. Specifically, a major computational hurdle is computing the asymptotic variances $\hat{\sigma}^2_{\text{symm}}, \hat{\sigma}^2_{\text{s1Left}}, \hat{\sigma}^2_{\text{s1Right}}$. Although we have discussed methods to generate $\hat{\text{v}}_{\text{s1Left}}, \hat{\text{v}}_{\text{s1Right}}$ in a computationally efficient manner, the other components of the variance computation present additional challenges. We will consider s1Left as an example. From Lemma 10,

$$\hat{\sigma}^2_{\text{s1Left}} = \hat{\text{v}}^T_{\text{s1Left}} \left(\hat{M}_{1,\text{s1Left}} \hat{V}_1 \hat{M}^T_{1,\text{s1Left}} + \hat{M}_{2,\text{s1Left}} \hat{V}_2 \hat{M}^T_{2,\text{s1Left}} \right) \hat{\text{v}}_{\text{s1Left}} \,,$$

where

$$\hat{M}_{1,\text{s1Left}} = \hat{\Sigma}_2^T \hat{\Delta}^T \otimes I_{2p} + I_{4p^2}$$

$$\hat{M}_{2,\text{s1Left}} = I_{2p} \otimes \hat{\Sigma}_1 \hat{\Delta} - I_{4p^2}$$

$$\hat{V}_l = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \frac{1}{2} \begin{bmatrix} (I_{p^2} + K)\left(\hat{A}_l \otimes \hat{A}_l + \hat{B}_l \otimes \hat{B}_l\right) & (I_{p^2} + K)\left(\hat{B}_l \otimes \hat{A}_l - \hat{A}_l \otimes \hat{B}_l\right) \\ [(I_{p^2} + K)\left(\hat{B}_l \otimes \hat{A}_l - \hat{A}_l \otimes \hat{B}_l\right)]^T & (I_{p^2} - K)\left(\hat{A}_l \otimes \hat{A}_l + \hat{B}_l \otimes \hat{B}_l\right) \end{bmatrix} \begin{bmatrix} P_1^T & P_2^T \end{bmatrix},$$

and P_1, P_2 are matrices of 0's, 1's, and -1's such that

$$\text{vec}(\hat{\Sigma}_l) = \begin{bmatrix} \text{vec}\left(\begin{bmatrix} \hat{A}_l \\ \hat{B}_l \end{bmatrix} \right) \\ \text{vec}\left(\begin{bmatrix} -\hat{B}_l \\ \hat{A}_l \end{bmatrix} \right) \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \begin{bmatrix} \text{vec}(\hat{A}_l) \\ \text{vec}(\hat{B}_l) \end{bmatrix}.$$

In this way, P_1, P_2 allow us to compute the asymptotic variance of $\text{vec}\left(\hat{\Sigma}_l\right)$ from the asymptotic variance of $\begin{bmatrix} \text{vec}(\hat{A}_l)^T & \text{vec}(\hat{B}_l)^T \end{bmatrix}^T$ solved for in Proposition 1. The exact forms of P_1, P_2 are given

in the Lemma 8. Note that, for ease of presentation, we have used the form of $\hat{V}_l$ that we would get for $\lambda \in (0, \pi)$. If $\lambda = 0$ or $\lambda = \pi$ the spectral density is real valued and the expansion to the real space is not necessary. In this case, Σ_l is the spectral density in condition l and is estimated by the smoothed periodogram. Furthermore, $\hat{V}_l = (I_{p^2} + K)\, \hat{A}_l \otimes \hat{A}_l$.

The difficulty with estimating $\hat{\sigma}^2_{\text{s1Left}}$ lies in the fact that $\hat{M}_{l,\text{s1Left}}, \hat{V}_l \in \mathbb{R}^{4p^2 \times 4p^2}$ which are too large to form except when p is small. However, since $\hat{\text{v}}_{\text{s1Left}}$ will be sparse in general, we will only need to compute specific entries of $\hat{D} := \hat{M}_{1,\text{s1Left}} \hat{V}_1 \hat{M}^T_{1,\text{s1Left}} + \hat{M}_{2,\text{s1Left}} \hat{V}_2 \hat{M}^T_{2,\text{s1Left}}$. Examining each component matrix in $\hat{M}_{1,\text{s1Left}} \hat{V}_1 \hat{M}^T_{1,\text{s1Left}} + \hat{M}_{2,\text{s1Left}} \hat{V}_2 \hat{M}^T_{2,\text{s1Left}}$ shows that they are entirely determined by $\hat{\Sigma}_l, \hat{\Delta}, \hat{A}_l$, and $\hat{B}_l$ which are at most of dimension $2p \times 2p$. Thus, an entry $\hat{D}_{ij}$ can be computed using only these matrices as inputs. As an example, consider computation of $\hat{\sigma}^2_{\text{s1Left}}$, and suppose $\hat{\text{v}}^T_{\text{s1Left}} = \begin{bmatrix} c_1 & 0 & c_3 & 0 & \dots \end{bmatrix}$. That is, $\hat{\text{v}}_{\text{s1Left}}$ only has two non-zero entries in the first and third positions. Then estimating $\hat{\sigma}^2_{\text{s1Left}}$ requires only four computations:

$$\hat{\sigma}^2_{\text{s1Left}} = \begin{bmatrix} c_1 & 0 & c_3 & 0 & \dots \end{bmatrix} \hat{D} \begin{bmatrix} c_1 \\ 0 \\ c_3 \\ 0 \\ \vdots \end{bmatrix} = c_1^2 \hat{D}_{11} + c_1 c_3 \hat{D}_{31} + c_1 c_3 \hat{D}_{13} + c_3^2 \hat{D}_{33}\,.$$

In general, if $\hat{\text{v}}_{\text{s1Left}}$ has $s_{\hat{\text{v}}_{\text{s1Left}}}$ non-zero entries, then $s^2_{\hat{\text{v}}_{\text{s1Left}}}$ computations are needed. However, computing the entries of $\hat{D}_{ij}$ is also non-trivial. Specifically, $\hat{D}_{ij}$ can be computed as

$$\hat{D}_{ij} = \left(\hat{M}_{1,\text{s1Left}}\right)_{i*} \hat{V}_1 \left(\hat{M}^T_{1,\text{s1Left}}\right)_{*j} + \left(\hat{M}_{2,\text{s1Left}}\right)_{i*} \hat{V}_2 \left(\hat{M}^T_{2,\text{s1Left}}\right)_{*j}\,,$$

where $(X)_{i*}$ and $(X)_{*j}$ select the i^{th} row and j^{th} column of X, respectively. From the form of $\hat{M}_{l,\text{s1Left}}$, we see that $\hat{M}_{l,\text{s1Left}}$ will be block diagonal with $2p$ blocks of dense $2p \times 2p$ matrices. Thus, forming one half of $\hat{D}_{ij}$ requires $4p^2$ computations. Combining these observations, we see that, in general, $O\left(s^2_{\hat{\text{v}}_{\text{s1Left}}} 8p^2\right)$ computations are needed to compute $\hat{\sigma}^2_{\text{s1Left}}$. Consider the case with $p = 100$ and $s_{\hat{\text{v}}_{\text{s1Left}}} = 100$ (since $\hat{\text{v}}_{\text{s1Left}} \in \mathbb{R}^{4p^2}$ this is a very sparse $\hat{\text{v}}_{\text{s1Left}}$). Then, computing $\hat{\sigma}^2_{\text{s1Left}}$ would require $8,000,000$ computations. This is only to compute the variance of one de-biased parameter.

If confidence intervals on many parameters are of interest, we will need to repeat this process for each parameter.

In practice, we use these concepts to estimate the asymptotic variances using only the sparse projection estimators, $\hat{\mathrm{v}}_{\mathrm{symm}}, \hat{\mathrm{v}}_{\mathrm{s1Left}}, \hat{\mathrm{v}}_{\mathrm{s1Right}}$ and the matrices $\hat{\Sigma}_l, \hat{\Delta}, \hat{A}_l$, and $\hat{B}_l$ as inputs. Computation of the asymptotic variances in this manner requires on-the-fly matrix multiplication and careful accounting of matrix indices. We implement these operations in `C++` by leveraging the `Rcpp` package (Eddelbuettel and François, 2011) in `R` (R Core Team, 2021). To maximize speed, the Dantzig selectors as well as rows and columns of $\hat{M}_l$ matrices are stored as sparse vectors using `RcppEigen` (Bates and Eddelbuettel, 2013).

3.7 Simulation studies

Next we study the performance of the SDD estimator and inference procedure in simulations. In Section 3.7.1 we compare how well SDD estimates the true difference in inverse spectral densities using various metrics. We further compare SDD to other competing methods. In Section 3.7.2 we study the performance of our inference procedure for all three estimating equations: symm, s1Left, s1Right.

3.7.1 Consistency

All simulations in this section use VAR(1) processes as this allows us to compute the true spectral density using results from Sun et al. (2018). Specifically we generate data in condition l as

$$X_{l,t} = A_l X_{l,t-1} + \boldsymbol{\epsilon}_{l,t}$$

where $\boldsymbol{\epsilon}_{l,t} \sim N_p(\mathbf{0}, I_p)$. From Sun et al. (2018), the spectral density at frequency λ is known to be

$$f_l(\lambda) = \frac{1}{2\pi} \left(\mathcal{A}_l(e^{-i\lambda})\right)^{-1} I_p \left(\left(\mathcal{A}_l(e^{-i\lambda})\right)^{-1}\right)^{\dagger},$$

where $\mathcal{A}_l(z) = I_p - A_l z$. When the transition matrix A_l is block diagonal, the spectral density $f_l(\lambda)$ is also block diagonal. Since the inverse of a block diagonal matrix can be computed block by block we can easily generate a sparse difference matrix by enforcing the transition matrix in conditions 1

and 2 to be the same except for in a small block. For example, with $p = 54$ if we generate A_1 and A_2 as block diagonal with the same 51×51 block and only differing in the final 3×3 block, then their spectral densities will only differ in this last 3×3 block. When converting to the real space, the expanded $\Sigma_i \in (2p \times 2p)$ and thus the true difference $\Delta = \Sigma_1 - \Sigma_2$ will differ in four 3×3 blocks.

We consider three different simulation settings all with dimension $p = 54$. In all simulation settings, the transition matrix in condition 1 is the same as that in condition 2 except the last 3×3 block is multiplied by -1. The first setting uses similar coefficients as in Sun et al. (2018), which we will refer to as Sim-Sun. In the second and third simulation settings, the transition matrix consists of one 51×51 block and one 3×3 block. The larger block was generated with 60% and 95% sparsity in the second and third settings which are referred to as Sim-Dense and Sim-Sparse respectively. All simulations were performed using $T = 100, 200, 500, 1000, 2000$ observations for both conditions. Smoothed periodograms were computed from data for each condition using a smoothing window of $B_l = \lceil T^{2/3} \rceil$ and then converted to the real space to generate $\hat{\Sigma}_l$. For all settings 100 evenly spaced Fourier frequencies from 0 to $\pi - n^{-1}$ were used. These were used as the spectral density is conjugate symmetric around 0. In the case of $n = 100$ only 50 Fourier frequencies were used as there are only 50 Fourier frequencies from 0 to $\pi - n^{-1}$. Given the availability of results from multiple frequencies, each simulation was run 50 times and the results are averaged across all frequencies and all runs. The standard error of each metric was also computed across frequencies and are reported in parentheses. More information on the simulation setup, optimization techniques, and tuning parameter selection can be found in Appendix B.6.

To solve the SDD estimation problem (3.3), the alternating direction method of multipliers (ADMM) algorithm from Yuan et al. (2017) was used. Specifically, we used the `L1_dts` function in the `Difdtl` package available on GitHub at SusanYuan/Difdtl (GPL ($\geq$ 2)). To generate the sequence of penalties, $\{\tau_{n_1,n_2}\}$, we used 20 values on a log-linear scale from $0.001 * \tau_{n_1,n_2,\max}$ to $\tau_{n_1,n_2,\max}$ where $\tau_{n_1,n_2,\max}$ represents the minimum value where all entries of $\hat{\Delta}$ are 0. In this case, $\tau_{n_1,n_2,\max} = 2\max(|\hat{\Sigma}_1 - \hat{\Sigma}_2|)$. The penalty $\tau^*_{n_1,n_2}$ was chosen as the τ_{n_1,n_2} that minimizes the eBIC (Foygel and Drton, 2010) which is computed as

$$\text{eBIC}(\hat{\Delta})_\gamma = \min(n_1, n_2) \left\| \frac{1}{2} \left(\hat{\Sigma}_1 \hat{\Delta} \hat{\Sigma}_2 + \hat{\Sigma}_2 \hat{\Delta} \hat{\Sigma}_1 - \hat{\Sigma}_2 + \hat{\Sigma}_1 \right) \right\|_\infty + \log(\min(n_1, n_2)) |E| + 4\gamma |E| \log(p) \,, \tag{3.10}$$

where $|E|$ is the number of unique edges in $\hat{\Delta}$. Since $\hat{\Delta}$ represents the difference in expanded spectral densities $|E|$ is the number of non-zero entries in the upper triangular portions, including diagonals, of the submatrices $\hat{\Delta}_{1:p,1:p}$ and $\hat{\Delta}_{1:p,(p+1):2p}$. For all applications we use $\gamma = 0.5$.

As previously noted, our method is the first to directly estimate the difference in inverse spectral densities. We now discuss competing methods which we compare SDD to in simulations. While not focused on directly estimating the difference, a related method is the joint graphical lasso with a fusion penalty (FGL, Danaher et al., 2014). This method aims to simultaneously estimate inverses in each condition that are believed to share structural similarities. It accomplishes this by using an ℓ_1 penalty on the difference of inverse matrices to encourage sparsity. Unlike SDD where we directly estimate the difference without estimating either inverse matrix, FGL estimates the inverse matrices in each condition. FGL has two penalty parameters, λ_1, λ_2 which control the sparsity of the invidual inverses and the sparsity of their difference respectively. As sparse differences are of more interest in our application, we tuned FGL for a wider range of λ_2 values than λ_1. Specifically, we tuned FGL using AIC for two λ_1 values and 10 λ_2 values for a total of 20 combinations, the same number of penalty values considered for other methods. The values for λ_1 were $(0.01\lambda_{1,\max}, 0.1\lambda_{1,\max})$ where $\lambda_{1,\max} = \left\| \hat{\Sigma}_{1,O} + \hat{\Sigma}_{2,O} \right\|_\infty / 2$ and $\hat{\Sigma}_{i,O}$ is the off-diagonals of $\hat{\Sigma}_i$. The λ_2 sequence was generated using 10 values on a log-linear scale from $0.0001 * \lambda_{2,\max}$ to $\lambda_{2,\max}$ where $\lambda_{2,\max} = \max\left(\left\| \hat{\Sigma}_{1,O} \right\|_\infty - 0.01\lambda_{1,\max}, \left\| \hat{\Sigma}_{2,O} \right\|_\infty - 0.01\lambda_{1,\max} \right)$. For additional comparisons, we conceptualized two potential alternatives to our SDD method. The first alternative method is the naïve method, which estimates Δ by taking the difference after estimating individual inverse spectral densities separately, $\hat{\Delta}_{\text{N}} = \hat{\Sigma}_1^{-1} - \hat{\Sigma}_2^{-1}$. For the naïve method, the graphical LASSO (GLASSO, Friedman et al., 2008) was used to estimate a sparse inverse spectral density in each condition. Specifically we used the fast implementation available in the `glassoFast` package in `R` (GPL ($\geq$ 3.0)) from Sustik and Calderhead (2012). To induce sparsity in individual estimates

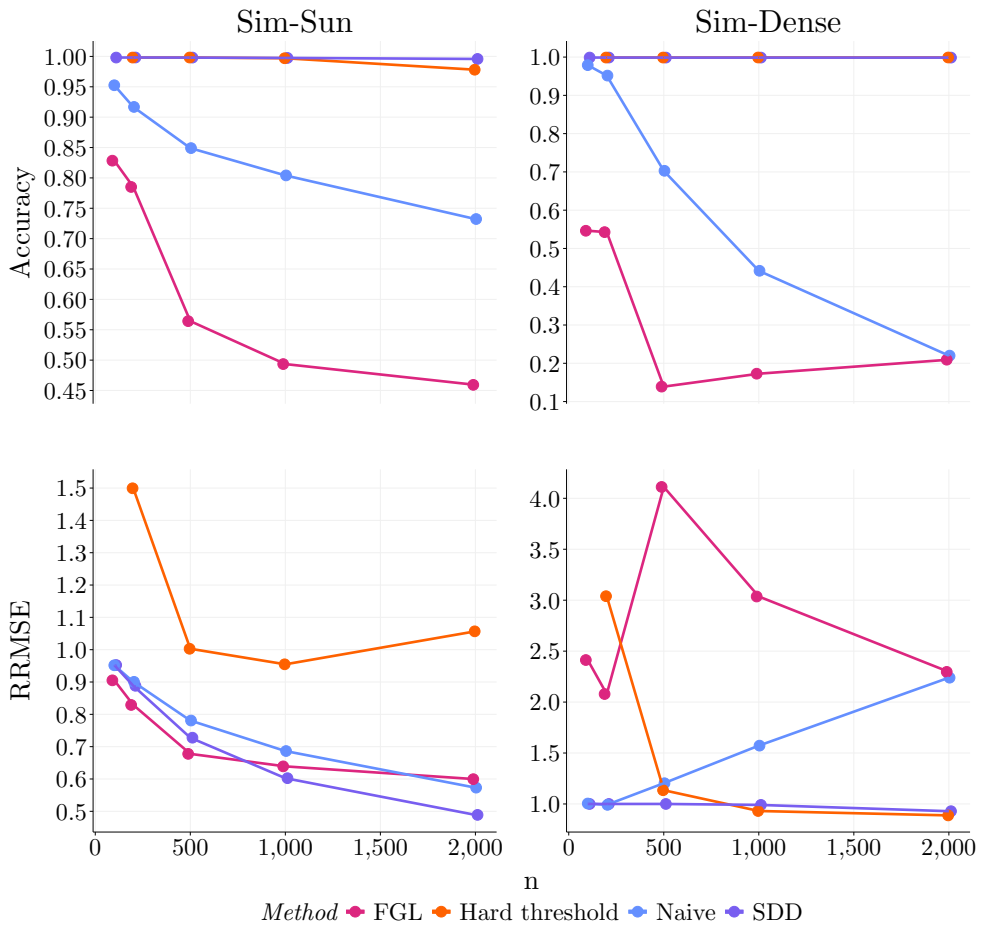


Figure 3.1: Sim-Sun (left) and Sim-Dense (right). Results are reported as mean (dots) and SE (vertical lines) where the mean and SE are taken across all frequencies and iterations for a given sample size T. Note that SE may appear as 0 due to the large number of frequencies and iterations.

and (potentially) their difference, GLASSO was tuned for each condition separately using eBIC with $\gamma = 0.5$ (Foygel and Drton, 2010). To select the tuning parameters for the Naïve method, we first computed $\lambda_{\max,i} = \left\|\hat{\Sigma}_i\right\|_\infty$ in each condition and then computed the tuning parameters as 20 evenly-spaced values on a log-linear scale from $0.0001\lambda_{\max,i}$ to $\lambda_{\max,i}$. The next method involves directly thresholding the difference to induce sparse estimates. Specifically, we estimate the inverse spectral density in each condition and take the difference and then apply Hard thresholding. To tune the threshold we used 20 evenly-spaced values on a log-linear scale from $\min(|\hat{\Delta}_H|)$ to $\max(|\hat{\Delta}_H|)$ where $\hat{\Delta}_H$ is generated by inverting $\hat{\Sigma}_i$ and forming the difference. Note that for $n = 100$, it is not possible to invert $\hat{\Sigma}_i$. The thresholds were tuned via eBIC. Similar to Deb et al. (2024), all methods are compared based on the following metrics: Accuracy, Precision, Recall, and Relative Root Mean Square Error (RRMSE). The full definitions can be found in Appendix B.6.

Simulation results for Accuracy and RRMSE in Sim-Sun and Sim-Dense are reported in Figure 3.1. Results for Precision and Recall for Sim-Sun and Sim-Dense can be found in Appendix B.6 Figure B.1. Results from Sim-Sparse are in Appendix B.6, Figures B.2. The full set of results for each of the simulations are in Appendix B.6 Tables B.1, B.2, B.3. In general, SDD has better accuracy than both the naïve method and FGL with the difference in accuracy increasing as the sample size increases. This is likely due to the fact that, in general, SDD estimates much fewer edges than the naïve or FGL methods. This also results in fewer false positive edges but more false negatives, resulting in better precision for SDD at the expense of reduced recall; see Figure B.1 in Appendix B.6. Across all settings we also see that SDD generally has lower RRMSE compared to the naïve method especially for large sample sizes. Overall, compared to the naïve method, SDD identifies a similar number of true edges using a much smaller edge set. Compared Hard thresholding, SDD has much better RRMSE in Sim-Sun and better recall across all settings and sample sizes. To further support our analysis, we have also summarized the difference between SDD and each of the competing methods for each metric in Appendix B.6, Tables B.4, B.5, B.6, B.7, B.8, B.9, B.10, B.11, B.12.

3.7.2 Inference

We study the performance of our inference procedure for all estimating equations for different combinations of p and T using simulations. Specifically, we consider $p = 15, 50, 100$ and $T = 100, 500, 1500, 2500$. For each combination, we perform 200 simulations. Similar to the consistency simulations in Section 3.7.1, we simulate data in conditions 1 and 2 as a VAR(1) process where the transition matrix is block diagonal with one large $p-3 \times p-3$ block and one small 3×3 block. All non-zero entries were randomly generated from either a $\mathrm{Uniform}(-0.8, -0.4)$ or a $\mathrm{Uniform}(0.4, 0.8)$ with equal probability. The transition matrix in condition 1 is generated with 50% sparsity in the larger block and 40% sparsity in the smaller block. As in Section 3.7.1, the transition matrix in condition 2 is the same as condition 1 except the smaller block is multiplied by -1. Using this setup, the true difference in inverse spectral densities is sparse and only differs in the smaller block.

For the symmetric estimating equation, it is not computationally feasible to perform inference for $p > 15$ so we only report results for s1Left and s1Right in these cases. The SDD estimate of the difference was generated using a smoothing span of $B = \lceil T^{2/3} \rceil$ and was tuned using BIC and 20 tuning parameter values as in Section 3.7.1. In the case of the symmetric estimating equation, the sparse projection estimates, $\hat{\mathrm{v}}_{\mathrm{symm}}$, were estimated using GLASSO and were tuned using eBIC with $\gamma = 0.5$. For s1Left and s1Right, $\hat{\Sigma}_l^{-1}$ was estimated using GLASSO for 20 tuning parameter values in each condition. Thus for each of the two conditions we have 20 GLASSO estimates of Σ_l^{-1} corresponding to each tuning parameter value. For each of the 400 combinations of the GLASSO estimates in each condition we choose the one that minimizes eBIC with $\gamma = 0.5$.

We perform our inference procedure for both the real and imaginary components of the 6 off-diagonal positions in the 3×3 block for a total of 12 entries. These are entries where the difference is potentially non-zero. We further perform inference on 12 randomly selected entries from the larger block. Since the larger blocks are the same between condition, these are entries where the true difference is 0. Lastly, we perform our inference procedure for 9 frequency values $\left(\lceil 0.1\frac{n}{2} \rceil \quad \lceil 0.2\frac{n}{2} \rceil \quad \ldots \quad \lceil 0.9\frac{n}{2} \rceil\right)$. We use 95% confidence intervals and study the Type I error,

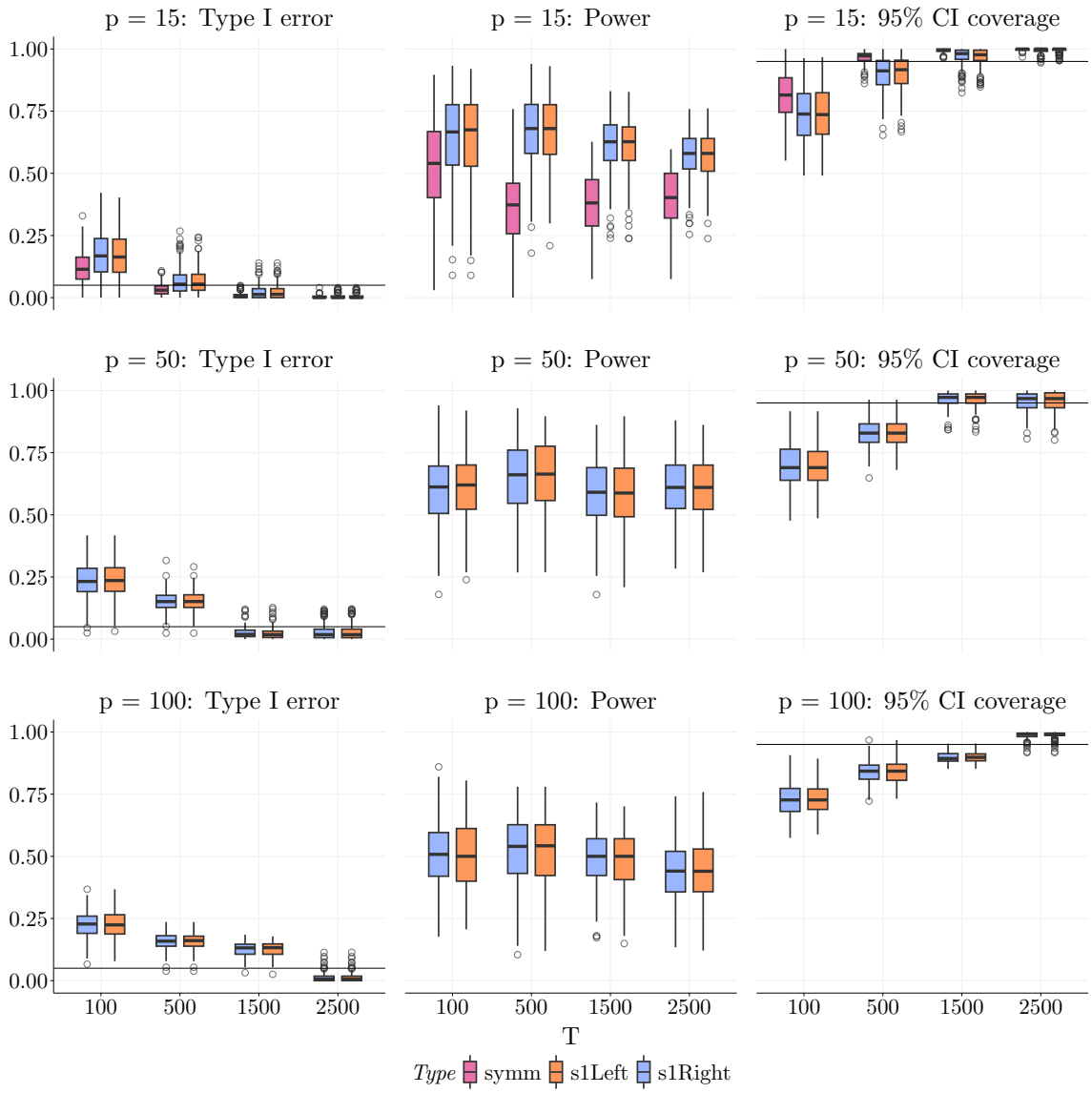


Figure 3.2: Type I error, power, and coverage for different combinations of p and T.

coverage, and power for each of the combinations of p and T and each estimating equation type. Each metric is averaged for each simulation over all frequencies and boxplots of the results for all 200 simulations are shown in Figure 3.2.

The results in Figure 3.2 show that for $p = 15$, compared to the symmetric estimating equation,

s1Left and s1Right have slightly worse type I error and coverage for small sample sizes but all methods achieve nominal type I error control and coverage when sample sizes are large. This is also true for $p = 50, 100$ although type I error and coverage are worse for smaller sample sizes compared to $p = 15$. Interestingly, s1Left and s1Right achieve better power than symm when type I error is controlled. We also see that the power does not change much with the increasing sample size and the effect of sample size is primarily observed in improved control of type I error. This is likely due to the fact that we are averaging over frequencies and it takes very large sample sizes to increase power given the scaling $T/B = T^{1/3}$.

3.8 Application to EEG Data

Next, we apply our direct difference estimator (SSD) and competing estimators to electroencephalograms (EEG) data from Hatlestad-Hall et al. (2022). Data were recorded with a 64 channel EEG array for 111 healthy subjects at a sampling frequency of 1024 Hz. Four minutes of brain activity was recorded while subjects were resting with their eyes closed. For 42 subjects, a second session was recorded 2–3 months after the initial session. We refer to these 42 subjects as those "with follow-up" and the remaining 69 as those "without follow-up." For our analysis, we used the pre-cleaned data provided in OpenNeuro Dataset ds003775 (Markiewicz et al., 2021). Specific cleaning steps can be found in Hatlestad-Hall et al. (2022). We also downsampled the data to 512 Hz.

To validate our method, we analyzed the subjects with follow-up and without follow-up separately. For those with follow-up, we estimated the difference in networks from the first to the second session (*across session analysis*). For those without, we estimated the difference in networks from 0-60s to 120-180s (*within session analysis*). The 60-120s block was used as a rest. Brain networks have been shown to be temporally dynamic (Zalesky et al., 2014; Nobukawa et al., 2019). Therefore, *a priori*, we expect the differential networks to be sparser in the within session analysis compared to the across session analysis and finding such results would provide validation of our method.

We compare the sparsity of the estimated differential network for SDD, the Naïve, Hard threshold, and FGL methods across the commonly-used Theta, Beta, Gamma and High-Gamma

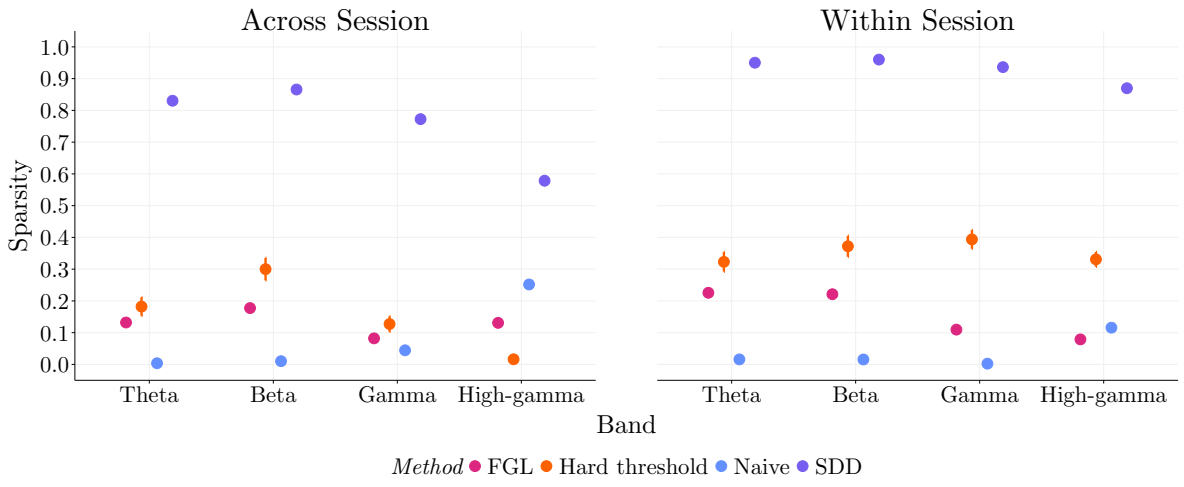


Figure 3.3: EEG Across (left) and Within (right) session analyses. Vertical lines indicate SE.

bands. Sparsity was defined as the proportion of entries in the estimated difference that were non-zero. Five or six evenly spaced frequencies were considered for each band. They were (in Hz): Theta: (4,5,6,7,8), Beta: (12, 16, 20, 24, 28), Gamma: (30, 40, 50, 60, 70), High-gamma: (80, 95, 110, 125, 140, 150). Results averaged over subjects and frequencies within each band are presented in Figure 3.3.

Figure 3.3 shows that SDD is the only method that estimates a meaningfully sparser difference in the within session analysis compared to the across session analysis across all frequency bands. While the Hard threshold method comes close, the estimated sparsity in the Beta band is similar between the across and within session analyses. All other competing methods have similar or less sparsity in the within session analysis compared to the across-session analysis. These results indicate that SDD estimates are more consistent with the experimental setting in this application.

3.9 Discussion

We proposed a direct estimate of the differences in inverse spectral densities between two conditions, termed the Spectral D-trace Difference (SDD) estimator. To our knowledge this is the first method of its kind in the spectral domain and only the second for time series data. We leveraged recent advances in the analysis of time series data to develop new convergence rates of the spectral density estimators. Using these rates we established convergence rates of SDD that can flexibly handle varying data dependence and smoothing spans and only require assuming sparsity in the difference of inverse spectral densities. Compared to the usual assumptions of sparsity of each inverse spectral density, the sparsity of the difference is more realistic if, for example, the inverse spectral densities do not change much between conditions. This is indeed what we expect in many biological settings, especially neurodegenerative disorders that are associated with changes in brain connectivity.

We additionally developed new inference procedures, only assuming mild dependence conditions, by considering estimating equations with computationally tractable decompositions and expanding existing results to handle asymptotic distributions with arbitrary scaling. A key ingredient to our inference procedure is the joint asymptotic distribution of the entire spectral density estimator which is of independent interest. We make our inference procedure computationally efficient by using sparse representations and intricate on-the-fly matrix multiplication in `C++`.

In simulations, the SDD estimator was compared to the fused graphical lasso (FGL), as well as a naïve GLASSO difference and hard thresholding estimator. The four methods were compared across three different simulation settings, where the true difference in inverse spectral densities was sparse. Comparison metrics included precision, recall, accuracy, and relative root mean square error (RRMSE). When compared to the other methods, SDD performed better in accurately identifying edges and non-edges and the SDD estimates had a lower relative root mean square error. The inference procedure was also studied using simulations which show small sample bias that becomes negligible as sample size increases to moderate to large values. This is true even for settings where p is large ($p > 50$). Lastly, the SDD estimator was applied to an EEG study where we further validated SDD by showing it estimated sparser differences between networks that were closer in

time, in line with expected behavior.

Chapter 4

DYNAMIC DEEP LEARNING FOR CHANGE-POINT DETECTION

4.1 Introduction

The change-point detection problem aims to identify changes in an underlying process. Some of the earliest works in this area used a cumulative sum score to detect changes in the fraction of defective products in a manufacturing plant (Page, 1954, 1955). Since then, new change-point detection methods have been developed and applied across a variety of fields, from climatology (Ducré-Robitaille et al., 2003) to finance (Kim et al., 2005) and neuroscience (Koepcke et al., 2016). Change-point detection methods can generally be categorized into offline or online methods. Compared to the online setting, where data is continuously collected and the goal is to identify changes as quickly as possible, in the offline setting data is retrospectively analyzed for changes. In this chapter, we focus on the offline setting.

There is a vast literature of non-deep-learning-based change-point detection methods, including parametric and nonparametric approaches. Frick et al. (2014) develop a procedure for the change in the parameter of a one-dimensional exponential family while changes in the mean of a signal with non-constant variance are considered in Arlot and Celisse (2011). Wang and Samworth (2018) consider a high-dimensional framework with a sparse change in the mean of a mean plus noise data model. Nonparametric methods to detect any change in the underlying probability distribution have been proposed in Matteson and James (2014), Arlot et al. (2019), and Padilla et al. (2021). Chakraborty and Zhang (2021) further extend nonparametric methods to the high-dimensional case. While flexible, these nonparametric methods assume the underlying data is independent and identically distributed (i.i.d) within each segment. However, many change-point detection applications involve time series data which has an inherent dependence between observations. As a result, methods based on the piecewise i.i.d assumption may be invalid. To address this, several

methods account for temporal dependence under the assumption that the data is generated as a vector autoregressive model (Safikhani and Shojaie, 2022; Safikhani et al., 2022; Bai et al., 2023). These methods are extended to include general high-dimensional linear models in Bai and Safikhani (2023). A more comprehensive review of existing offline change-point detection methods can be found in Truong et al. (2020).

Deep learning methods have the ability to learn complex data representations and have achieved success across a wide variety of fields, from speech and image recognition to natural language understanding (LeCun et al., 2015; Hinton et al., 2012; Krizhevsky et al., 2017; Vaswani et al., 2017; Radford et al., 2018). A natural question is whether machine learning or deep learning-based methods can be developed for change-point detection. Many recent papers have approached this question from both a supervised and unsupervised perspective.

For unsupervised methods, Londschien et al. (2023) use random forests as classifiers in combination with a new classifier log-likelihood ratio to detect change-points. While the method is flexible, it assumes i.i.d. data within segments. Several unsupervised deep learning methods for depedendent have been developed. For instance, De Ryck et al. (2021), uses autoencoders on both time-domain and frequency domain features, while Chang et al. (2019) approach the problem from a two-sample testing perspective and use synthetic data generating models with RNNs to detect change-points. Ryzhikov et al. (2023) combine model-based and deep learning approaches using latent stochastic differential equations. Huang et al. (2023) use a hybrid supervised and unsupervised approach by generating synthetic signals. Lastly, Jones and Harchaoui (2020) provide an end-to-end deep learning method that jointly learns feature representations and change-points. Their method, XSCPE, detects changes in the mean feature representation and allows for any number of known change-point locations. However, a major drawback is that XSCPE requires training, validation, and test sequences.

Contrary to unsupervised methods, supervised deep learning approaches require labeled training data. Li et al. (2024) require training data with known change-points in order to learn neural network-based change-point classifiers. Similarly, Ebrahimzadeh et al. (2019) require a set of training data to learn model parameters in their proposed pyramid recurrent neural network. A

semi-supervised approach is taken in Khan et al. (2019), where a prespecified number of samples at the start of the data are assumed to be from the reference distribution. Kanrar et al. (2024) use a similar sample splitting approach but make the additional assumption that samples at the end of the series belong to a separate regime. Hushchyn and Ustyuzhanin (2021) directly estimate density ratios with deep learning methods, but again rely on reference and test set splitting which implicitly assumes there is no change-point within the reference or test set. Kloska et al. (2023) makes use of expert knowledge through a human-in-the-loop framework. In practice, labeled change-point data may not be available or it may be difficult to the pick the number of samples at the beginning or end of the data that are known to be apart of the same regime.

We present a fully unsupervised deep learning framework for change-point detection based on the novel concept of Dynamic Deep Learning (DDL). Key to this new framework is the representation of change-points as differentiable parameters, which allows for joint optimization of predictive models and change-points. This allows DDL to estimate both the location of unknown changes in the prediction function as well as the prediction functions within each regime. Our method is flexible and can be tailored to a variety of architectures and loss functions. While our primary focus in this chapter is on time series data, DDL is applicable to both dependent and independent data. Moreover, DDL integrates easily into existing deep learning libraries such as `PyTorch`, allowing computationally efficient training even for very large models and datasets.

The rest of the chapter is organized as follows. In Section 4.2, we formulate the change-point detection problem and introduce our new dynamic deep learning framework. Simulations comparing DDL change-point detection to existing deep learning and benchmark methods are carried out in Section 4.3. DDL is applied to COVID-19 outcomes in New York City in Section 4.4. Finally, extensions of DDL and concluding remarks are discussed in Sections 4.5 and 5.1.2, respectively.

4.2 Change-point Detection

Suppose we observe time-series data $(y_t, x_t)_{t=1/T}^{1}$ and our goal is to predict y_t based on covariates $x_t \in \mathbb{R}^p$. Note that we use the time indexing $\{1/T, \ldots, 1\}$ instead of $\{1, \ldots, T\}$. In the absence of change-points, we would make predictions $\hat{y}_t = f_{\hat{\theta}}(x_t)$ where $f_{\hat{\theta}}(z)$ could be a simple linear

predictor or a feed-forward neural network. That is, we would use the same prediction function $f_{\hat{\theta}}(x_t)$ for all t where $\hat{\theta}$ is estimated as

$$\hat{\theta} = \arg\min_{\theta} \sum_{t=1/T}^{1} L\left(y_t, f_\theta(x_t)\right) ,$$

for some loss function L. In practice, when $f_\theta(z)$ is, for example, a feed-forward neural network, this minimization problem is non-convex and $\hat{\theta}$ is learned by stochastic gradient methods. When analyzing time series data, this regression function may change over time due to changes in, e.g., fiscal policy or system behavior. In this case, we wish to allow the regression function to change. It is worth noting that compared to nonparametric methods that aim to detect changes in the entire underlying probability distribution of the data, we focus on detecting changes in the regression function which is a determined by the joint probability distribution (Arlot et al., 2019; Matteson and James, 2014). Consider first the case where there is a single change-point that is known to be located at $t = \tau_1$, then we could fit

$$\hat{\theta}_0 = \arg\min_{\theta_0} \sum_{t=1/T}^{\tau_1} L(y_t, f_{\theta_0}(x_t))$$
$$\hat{\theta}_1 = \arg\min_{\theta_1} \sum_{t=\tau_1+1/T}^{1} L(y_t, f_{\theta_1}(x_t)) ,$$

and our predictions for y_t would be

$$\hat{y}_t = f_{\hat{\theta}_0}(x_t) I\left(t \leq \tau_1\right) + f_{\hat{\theta}_1}(x_t) I\left(t > \tau_1\right) ,$$

where $I\left(A\right)$ is the indicator function that takes a value of 1 if event A occurs and 0 otherwise. In practice, the change-point locations are unknown and must be estimated. A brute-force approach would be to estimate $\hat{\theta}_0, \hat{\theta}_1$ for every possible $\tau_1 = 1/T, \ldots, 1$ and choose the one that minimizes an error or loss. However, when $f_\theta(z)$ is a feed-forward neural network, this requires re-training two networks for each candidate change-point (i.e., before and after the candidate change-point) and is computationally prohibitive. Instead, we will formulate the problem so that the change-points are parameters of the model and can be directly optimized through stochastic gradient methods.

Consider a fixed number of change-points D as $\{\tau_j\}_{j=1}^{D}$. We further define $\tau_0 = -\infty$ and $\tau_{D+1} = \infty$. The choice of $\pm\infty$ instead of $0, 1$ will become clear later. For D change-points, there are $D + 1$ regimes, from $(\tau_0, \tau_1], \ldots (\tau_D, \tau_{D+1}]$. For now, we assume that the number of change-points D is known. Extensions to the unknown case are discussed in Section 5.1.2. We then wish to solve

$$\hat{\theta}_0, \ldots, \hat{\theta}_D, \hat{\tau}_1, \ldots, \hat{\tau}_D = \underset{\theta_0,\ldots,\theta_D,\tau_1,\ldots\tau_D}{\arg\min} \sum_{t=1/T}^{1} L\left(y_t, \sum_{j=0}^{D} f_{\theta_j}(x_t) I\left(\tau_j < t \leq \tau_{j+1}\right)\right) . \tag{4.1}$$

4.2.1 Dynamic Deep Learning

We cannot use stochastic optimization techniques to approximately solve Eq. (4.1) as the indicator functions are not differentiable with respect to τ_j. Instead, we replace the indicator functions with a differentiable approximation. This is the key idea behind DDL. By introducing a differentiable approximation to the indicator function, we can directly optimize both the change-point parameters and the model parameters. To this end, we approximate the indicator functions by a difference in sigmoids:

$$w_t(\tau_j, \tau_{j+1}) := \frac{1}{1 + \exp(-k(t - \tau_j))} - \frac{1}{1 + \exp(-k(t - \tau_{j+1}))} \approx I\left(\tau_j < t \leq \tau_{j+1}\right) ,$$

where k is a scaling parameter that controls how quickly the sigmoid approximation rises from 0 to 1. Larger values of k result in sharper approximations, though very large values of k can result in exploding gradients. While this can be addressed with gradient clipping, we find that, in practice, $k = 50$ offers good performance and does not require clipping. An example of this approximation is shown in Figure 4.1.

With this differentiable approximation to the indicator function, our optimization problem is now

$$\hat{\theta}_0, \ldots, \hat{\theta}_D, \hat{\tau}_1, \ldots, \hat{\tau}_D = \underset{\theta_0,\ldots,\theta_D,\tau_1,\ldots\tau_D}{\arg\min} \sum_{t=1/T}^{1} L\left(y_t, \sum_{j=0}^{D} f_{\theta_j}(x_t) w_t(\tau_j, \tau_{j+1})\right) , \tag{4.2}$$

which can be approximately solved using existing stochastic gradient methods, in e.g. `PyTorch`. Moreover, given the formulation of the indicator approximation, we use $\tau_0 = -\infty, \tau_{D+1} = \infty$ so

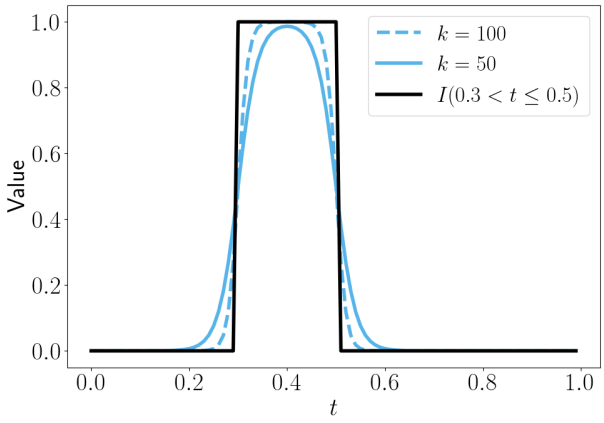


Figure 4.1: Example of indicator function and it's corresponding sigmoid approximation, $\frac{1}{1+\exp(-k(t-0.3))} - \frac{1}{1+\exp(-k(t-0.5))}$ for $k = 100$ and $k = 50$.

that the value of $w_0(\tau_0, \tau_1) = 1$ and $w_1(\tau_D, \tau_{D+1}) = 1$. An example of how DDL uses weights and predictions is given in Figure 4.2.

So far we have considered the analysis of time series data where model parameters are allowed to change over time. However, the proposed DDL framework is general and can be applied to allow model parameters to change over any variable. As a motivating example, consider the case where we are trying to predict disease risk and it is known that disease risk differs between children and adults, but the exact age at which risk changes is unknown. In this example we assume observations do not have time ordering and thus will denote our data as $(y_i, x_i)_{i=1}^n$. For simplicity, we assume we consider detecting change-points in the first covariate, $x_{i,1}$. We also assume, for convenience, that $x_{i,1} \in (0, 1]$ which can be achieved by scaling, $(x_{i,1} - \min_i(x_{i,1}))/(\max_i(x_{i,1}) - \min({}_i x_{i,1}))$. To distinguish from the time series case, we denote the change-points as $\{\delta_i\}_{i=1}^D$ and again assume that $\delta_0 = -\infty, \delta_1 = \infty$. The DDL optimization problem in this case becomes

$$\hat{\theta}_0, \ldots, \hat{\theta}_D, \hat{\delta}_1, \ldots, \hat{\delta}_D = \underset{\theta_0,\ldots,\theta_D,\delta_1,\ldots\delta_D}{\arg\min} \sum_{i=1}^{n} L\left(y_i, \sum_{j=0}^{D} f_{\theta_j}(x_i) w_{x_{i,1}}(\delta_j, \delta_{j+1})\right),$$

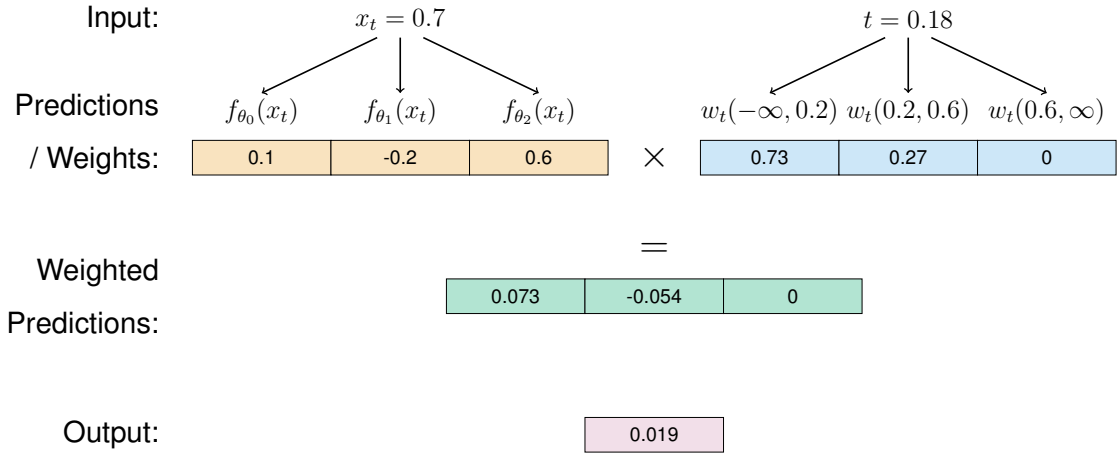


Figure 4.2: DDL workflow. **Inputs**. The inputs to DDL change-point detection are a covariate vector x_t and its corresponding time t. **Predictions/Weights**. Predictions from each regime, f_{θ_j}, are generated based on the covariates x_t. Simultaneously, weights for each regime are computed based on t and the current change-point locations. In this case there are two change-points: $\tau_1 = 0.2$, $\tau_2 = 0.6$. **Weighted Predictions**. Predictions for each regime and their corresponding weights are multiplied element-wise. **Output**. All weighted predictions are summed to compute the output.

where

$$w_{x_{i,1}}(\delta_j, \delta_{j+1}) := \frac{1}{1 + \exp(-k(x_{i,1} - \delta_j))} - \frac{1}{1 + \exp(-k(x_{i,1} - \delta_{j+1}))} .$$

This is essentially the same as the time series case except the change-points now partition $x_{i,1}$ instead of time t. Again, this optimization problem can be easily approximated by stochastic gradient methods.

In the next section, we dive into further detail on the mechanism for which change-points and model parameters are learned.

4.2.2 How Change-points and Parameters are Learned

Suppose we are in the time series regression setting, where y_t is a univariate real output and we wish to train our model using squared error loss. Then, the loss over the entire dataset can

be written as $L(y, \hat{y}) = \sum_{t=1/T}^{1}(y_t - \hat{y}_t)^2$, where $\hat{y} = \{\hat{y}_t\}_{t=1/T}^{1}$, $y = \{y_t\}_{t=1/T}^{1}$, and $\hat{y}_t = \sum_{j=0}^{D} f_{\hat{\theta}_j}(x_t) w_t(\tau_j, \tau_{j+1})$. Consider the case with one change-point and thus two regimes. The gradients with respect to $\tau_1, \theta_0, \theta_1$ are given by

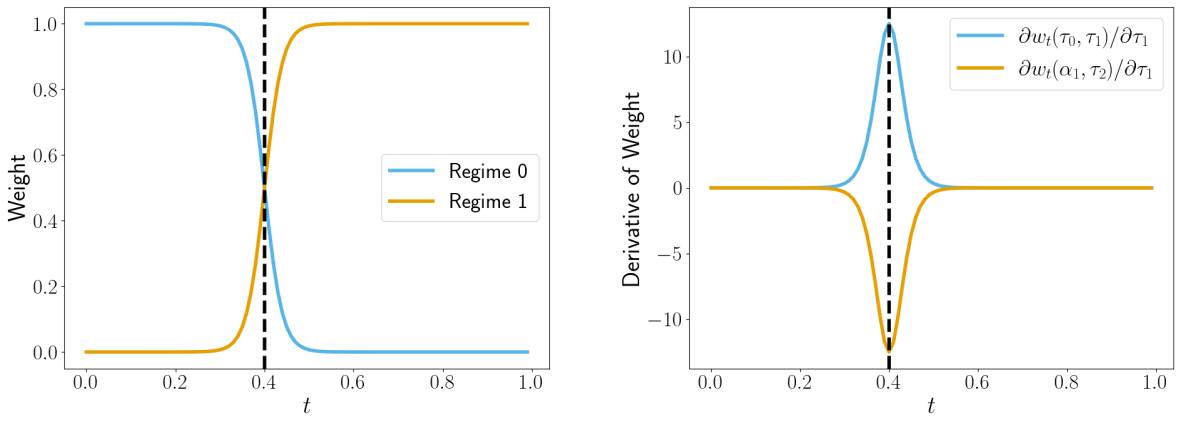
$$\frac{\partial L(y, \hat{y})}{\partial \tau_1} = -\sum_{t=1/T}^{1} 2(y_t - \hat{y}_t)\frac{\partial \hat{y}_t}{\partial \tau_1} = -\sum_{t=1/T}^{1} 2(y_t - \hat{y}_t)\left(f_{\theta_0}(x_t)\frac{\partial w_t(\tau_0, \tau_1)}{\partial \tau_1} + f_{\theta_1}(x_t)\frac{\partial w_t(\tau_1, \tau_2)}{\partial \tau_1}\right), \tag{4.3}$$

$$\frac{\partial L(y, \hat{y})}{\partial \theta_0} = -\sum_{t=1/T}^{1} 2(y_t - \hat{y}_t)\frac{\partial \hat{y}_t}{\partial \theta_0} = -\sum_{t=1/T}^{1} 2(y_t - \hat{y}_t)\left(\frac{\partial f_{\theta_0}(x_t)}{\partial \theta_0} w_t(\tau_0, \tau_1)\right), \tag{4.4}$$

$$\frac{\partial L(y, \hat{y})}{\partial \theta_1} = -\sum_{t=1/T}^{1} 2(y_t - \hat{y}_t)\frac{\partial \hat{y}_t}{\partial \theta_1} = -\sum_{t=1/T}^{1} 2(y_t - \hat{y}_t)\left(\frac{\partial f_1(x_t)}{\partial \theta_1} w_t(\tau_1, \tau_2)\right). \tag{4.5}$$

While these calculations are standard due to the very general form of $\hat{y}_t$, they provide useful insight into which observations impact the gradient of which parameters. Specifically, from Eq. (4.3), we see that any observations where $\partial w_t(\tau_0, \tau_1)/\partial \tau_1 = 0$ or $\partial w_t(\tau_1, \tau_2)/\partial \tau_1 = 0$ would not contribute to the gradient of τ_1 and thus would not determine how the change-point is updated. Only observations where $\partial w_t(\tau_0, \tau_1)/\partial \tau_1$ or $\partial w_t(\tau_1, \tau_2)/\partial \tau_1 \neq 0$ move the change-point around. As we can see from Figure 4.3b, these observations are in a small window around the change-point, approximately observations where $t \in (0.3, 0.5)$ when the change-point is 0.4. This window gets wider with smaller k.

To see which observations impact the gradients of θ_0, θ_1, we can study Eqs. (4.4) and (4.5). These equations show that only the observations where $w_t(\tau_0, \tau_1) \neq 0$ or $w_t(\tau_1, \tau_2) \neq 0$ contribute to the gradients of θ_0, θ_1, respectively. From Figure 4.3, we see that these correspond to the observations in *Regime 0* and *Regime 1*, respectively. Put succinctly, only the observations in *Regime 0* are used to update f_{θ_0}, while only the observations in Regime 1 are used to update f_{θ_1}. However, in settings where $f_{\theta_0}, f_{\theta_1}$ share parameters, these decompositions won't be as clean. See additional discussions in Section 5.1.2.



(a) Example weights.

(b) Derivative of weights w.r.t τ_1.

Figure 4.3: (a) Example of weights for each regime. Change-points are indicated by dashed lines. The weights for each regime decay rapidly outside regime boundary. For example, the boundary of *Regime 0* is $t = (0, 0.4]$. When $t = 0.6$ the weight for *Regime 0* is effectively 0. (b) Derivative of weight function in (a) w.r.t. τ_1.

4.3 Simulations

In this section, we study the performance of our dynamic deep learning change-point detection method in simulations and compare its performance to existing deep learning and baseline methods. It is worth noting that when detecting change-points with each method we will assume the true number of change-points is known when the method allows.

4.3.1 Simulation settings

We consider three simulation settings with a combination of linear and non-linear vector autoregressive processes. More information on these processes and visualizations for one realization of the data can be found in Appendix C.1. The simulation settings considered correspond to challenging settings where change-point locations may be difficult to visually identify.

We use $\mathrm{VAR}_p(l)$ to denote a linear VAR process with p variables and order l, i.e. the VAR model depends on l prior lags. In the first setting, we simulate data from a $\mathrm{VAR}_3(2)$ process with 3 change-points. The entries from the transition matrices are generated from a $\mathrm{Unif}(-2, 2)$ distribution. The matrices are then rescaled so that the maximum modulus of the eigenvalues of the $\mathrm{VAR}(1)$ representation is less than 0.7, which ensures stability of the process. For more information on stability of VAR processes and how any VAR process with l lags can be written as a VAR process of lag 1, see Lütkepohl (2005, pp.15-16 and Eq. (2.1.8)). In the second and third settings, similar to the NLAR1 and NLAR1U2 processes from Huang and Yang (2004), data is simulated as a non-linear combination of prior lags. The maximum lag used in each setting is 4 while the dimension of the processes are 3 and 2, respectively. Using the same notation as for VAR models, we will use the shorthands $\mathrm{NLAR}_3(4)$ and $\mathrm{NLARU}_2(4)$ to denote these processes and their exact mathematical formulations can be found in Appendices C.1.1 and C.1.3, respectively. There are 2 change-points in the $\mathrm{NLAR}_3(4)$ and 1 change-point in the $\mathrm{NLARU}_2(4)$ setting. For all settings, we simulate errors from a mean zero Gaussian distribution with identity covariance. Data in each regime is centered and scaled separately which ensures that methods cannot detect change-points as a result of mean or covariance shifts. We perform 100 simulations with $T = 2000$ and a burn-in

of $B = 2000$. Thus a total of 4000 samples are generated, but the first 2000 are discarded. The locations of the change-points as well as the transition matrices and parameters of the non-linear models are generated once and used across all repetitions.

4.3.2 Competing methods

The first method considered is KL-CPD of Chang et al. (2019). This method compares samples before and after t to detect whether or not time point t is a change-point. Specifically, it combines synthetic data generation with RNNs to learn kernels that maximize the power of a maximum mean discrepancy test. The output of KL-CPD is a vector of the length of the time series, where each entry represents the score of time point t as a change-point, with higher scores indicating a higher likelihood of being a change-point. Since an official package is not available, we use the implementation from https://github.com/HolyBayes/klcpd.

We next consider the TIRE method from De Ryck et al. (2021), which estimates and detects changes in time-invariant features. Intuitively, a change-point occurs when such time-invariant features change. Autoencoders and time-invariant features are separately learned for both the time and frequency domains. To encourage sparse changes in time-invariant features, the authors penalize the difference across many time-points. Finally, the time-invariant features from the time and frequency domains are combined to detect the change-points. Similar to KL-CPD, the output of TIRE is a change-point score vector that is the length of the time series and similarly we use the implementation from https://github.com/HolyBayes/TIRE_pytorch.

The last machine learning method we consider is `changeforest` from Londschien et al. (2023). The authors use a classifier log-likelihood ratio to detect change-points. They use random forests as their classifiers and combine them with a two-step search procedure and binary segmentation to maximize approximate gain in the classifier log-likelihood ratio (Breiman, 2001; Vostrikova, 1981). The method is implemented in the Python package `changeforest`. While this method was designed for i.i.d. data, we adapt it to the time series case by concatenating data from l prior lags for each observation as mentioned in Section 5 of Londschien et al. (2023).

In addition to the machine learning methods mentioned, we also include KCP and ECP which

are two well-known non-parameteric change-point detection methods (Arlot et al., 2019; Matteson and James, 2014). KCP is a kernel change-point detection method that is implemented in Python in the `ruptures` package (Truong et al., 2020). We use the Gaussian kernel with default parameters. ECP uses an empirical divergence measure between two distributions. An implementation of ECP is available in the R package `ecp`, which we use in Python through the `rpy2` package (James and Matteson, 2015).

We also considered the latent SDE method of Ryzhikov et al. (2023). However, since the paper computes evaluation metrics after each epoch and saves the best results, this method was not comparable in our settings; To adapt the method to our setting, we computed the change-point scores after 100 epochs, but we found that the change-point scores were always zero. As such, latent SDE was excluded from the analysis.

Since `changeforest`, KCP, and ECP were developed for i.i.d. data, we briefly discuss how to adapt these methods to the time series setting. To do so, we follow the suggestion in Section 5 of Londschien et al. (2023) and augment the data at time t with l prior lags. That is, to detect change-points at time t, we use the observation $\begin{bmatrix} x_t & x_{t-1/T} & \dots & x_{t-l/T} \end{bmatrix} \in \mathbb{R}^{(p+1)l}$. In this way, we transform the $\mathbb{R}^{T \times p}$ data matrix into a $\mathbb{R}^{T-l \times (p+1)l}$ matrix, which is then used by `changeforest`, KCP, and ECP to detect change-points. In practice, the number of prior lags l is unknown and needs to be selected. One method is to select a l that is hypothesized to be larger than the true lag order to capture all the relevant historical data. With this in mind, we choose l to be greater than or equal to the true number of lags. For the $\mathrm{VAR}_3(2)$ and $\mathrm{NLAR}_3(4)$ settings, we use the true lag order of the processes $l = 2$ and $l = 4$, respectively. For $\mathrm{NLARU}_2(4)$, we assume the true lags are unknown and set the number of prior lags to be $l = 5$, larger than the true lag order 4. Experiments with misspecified lags l that are smaller than the true number of lags are available in Appendix C.1.4. Most methods, including DDL, perform worse under misspecification but are not materially worse. While KL-CPD and TIRE do not explicitly require concatenation of prior lags, an equivalent parameter for KL-CPD and TIRE are the window sizes, both of which are larger than the true lag sizes.

Whenever available, we use the default package parameters. We train TIRE and KL-CPD

using 200 and 100 epochs, respectively. Fewer epochs are used for KL-CPD as it is much more computationally intensive than TIRE. For `changeforest` we use random forests as our classifier and the binary segmentation search procedure. We use KCP with a Gaussian kernel and a minimum segment length of 10 observation while all default parameters are used for ECP.

Next, we discuss how change-points are selected for each method. When possible, we assume the true number of change-points, D^*, is known. For example, in the $\mathrm{VAR}_3(2)$ setting, we assume we know there are 3 change-points so $D^* = 3$. Recall that TIRE and KL-CPD are score-based methods, which return a change-point score for each data-point. For both of these methods, we select the D^* change-points with the highest score while ensuring that the minimum distance between any two change-points is 0.05 (for change-points in the $(0, 1]$ interval). The `ruptures` implementation of KCP allows to fit a known number of change-points so we use D^* for this parameter. Both `changeforest` and ECP assume the number of change-points is unknown and neither have easy methods to fix the number of known change-points or select those with the highest score. As a result, we simply use the change-points returned from each method. KL-CPD and TIRE are both trained on NVIDIA L40S GPU with 48 GB. Other methods cannot leverage GPU acceleration and are trained using 64 GB of RAM of the same NVIDIA machine.

4.3.3 DDL setup

So far, we have presented a general formulation of the DDL change-point detection problem to highlight its versatility. In this section, we discuss our implementation for the simulation settings in Section 4.3.1. We use the same set-up for each setting. In each regime we model f_{θ_j} as a feed-forward neural network. For a time t, the goal is to predict x_t based on prior lags. Thus, the output at time t is $y_t := x_t$ while the inputs to the feed-forward neural network are $z_t := \begin{bmatrix} x_{t-1/T} & \dots & x_{t-l/T} \end{bmatrix} \in \mathbb{R}^{(p+1)l}$. The input to the weighting function is the corresponding observation time, $t \in (0, 1]$. The observation time is only used to compute the weights for each regime as in Figure 4.2. It is not used by the neural networks. Since our output is $y_t \in \mathbb{R}^p$, we configure the network to output a $\mathbb{R}^p$ vector and use MSE as our loss.

The network in each regime is configured using two hidden layers each with 8 nodes, ReLU

activation, and a dropout rate of 0.1. All methods are trained for 200 epochs with a batch size of 32. We use the Adam optimizer with a learning rate of 0.0003 for the change-points and 0.001 for the parameters of the neural network. We find that using a learning rate for the change-points that is roughly three to ten times smaller than the learning rate for the network parameters gives good results. Due to the relatively small number of parameters, we train all DDL models on a CPU with 64GB of RAM. For these small models and dataset sizes, training one epoch of the DDL model takes roughly a second on a personal MacBook pro with a 2.3 GHz Quad-Core Intel Core i7 processor and 16 GB of RAM.

4.3.4 Metrics

We use three metrics to evaluate performance of each method. The first is the adjusted Rand index (ARI) (Hubert and Arabie, 1985). The Rand index is computed as the number of agreements between two clusterings divided by the total number of possible pairs of observations, which is $\binom{T}{2}$. The number of agreements is computed as the number of pairs of observations that are assigned to the same or different clusters for each clustering. The adjusted Rand index centers the Rand index by its expected value and then rescales by its maximum. The maximum value of ARI is 1 and indicates perfect clustering agreements.

The second metric we study is the F_1 score, which is the harmonic mean of precision, P, and recall, R. We first define precision and recall. Denote the true change-point locations as D^* and the estimated change-point locations as $\hat{D}$. We follow Van den Burg and Williams (2020) and define the true positives TP of $\hat{D}$ to be those $\tau^* \in D^*$ for which $\exists\, \hat{\tau} \in \hat{D}$ such that $|\tau^* - \hat{\tau}| < M$. We also ensure that only one $\hat{\tau}$ can be used for each τ^*. This avoids a setting where $\hat{\tau}_1 = 0.1, \hat{\tau}_2 = 0.12$ are both true positives for $\tau^* = 0.11$. We detect change-points in the $(0, 1]$ representation of time so we set $M = 0.05$. With this, precision and recall are given by

$$\mathrm{P} = \frac{\mathrm{TP}}{|\hat{D}|},$$
$$\mathrm{R} = \frac{\mathrm{TP}}{|D^*|},$$

where $|D^*|$ represents the cardinality of D^*. F_1 score is then defined as

$$F_1 = 2\frac{\mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}}\,. \tag{4.6}$$

The F_1 score takes values between 0 and 1 with 1 indicating perfect precision and recall.

The last metric we consider is the symmetric Hausdorff distance which is computed as the maximum of the directed Hausdorff distances

$$d_H(\hat{D}, D^*) = \max\left(\max_{\hat{\tau}\in\hat{D}} \min_{\tau^*\in D^*} |\hat{\tau} - \tau^*|, \max_{\tau^*\in D^*} \min_{\hat{\tau}\in\hat{D}} |\tau^* - \hat{\tau}|\right)\,. \tag{4.7}$$

Since we map the time to the $(0, 1]$ interval, the maximum Hausdorff distance is 1 while the minimum is 0.

4.3.5 Results

The simulation results for the three settings and all metrics are presented in Figures 4.4 to 4.6. For each setting, 100 replicates were performed and the mean and standard error were computed for each metric. Despite being visually easier to detect change-points (see Figure C.1), the $\mathrm{NLAR}_3(4)$ is the hardest setting with all methods performing worse relative to $\mathrm{VAR}_3(2)$ and $\mathrm{NLARU}_2(4)$. Overall, we see that DDL is the best performing method across all metrics and simulation settings. In both the $\mathrm{VAR}_3(2)$ and $\mathrm{NLARU}_2(4)$ settings, DDL achieves nearly perfect ARI and F_1 score, while DDL achieves ARI and $F_1 \gtrapprox 0.8$ in $\mathrm{NLAR}_3(4)$. The mean Hausdorff distance for change-points estimated by DDL are ≈ 0.08 in the $\mathrm{NLAR}_3(4)$ setting, indicating reasonable change-point recovery and are nearly 0 in the other settings indicating near perfect change-point recovery. The performance gap between DDL and other methods is most notable in the $\mathrm{NLAR}_3(4)$, the hardest setting for all methods.

Despite being deep learning-based, KL-CPD and TIRE do not perform well across any metrics or any settings. Similarly, ECP appears to perform poorly in all settings. While performing nearly perfectly in the $\mathrm{NLARU}_2(4)$ setting, KCP performs comparably or slightly better than existing deep learning methods in the other settings with an ARI < 0.6 and the F_1 score < 0.4 for both $\mathrm{VAR}_3(2)$ and $\mathrm{NLAR}_3(4)$. Although `changeforest` performs poorly in the $\mathrm{NLAR}_3(4)$ setting, it achieves

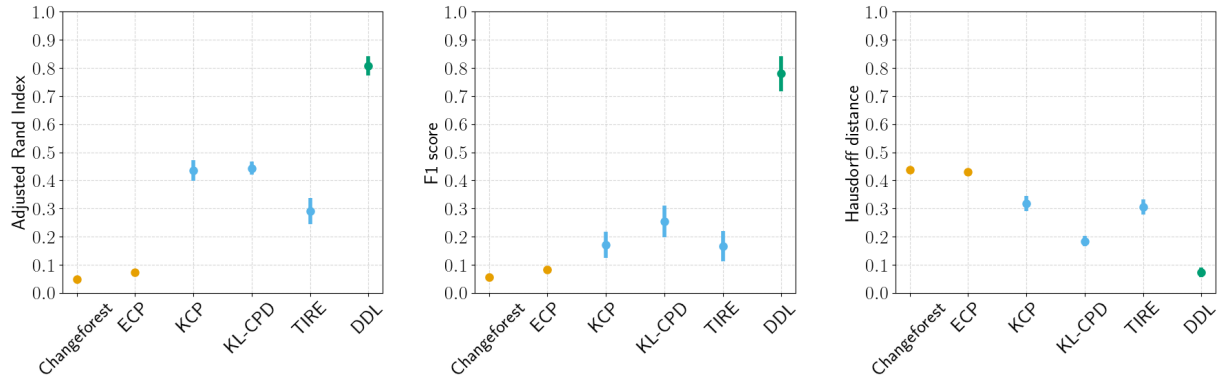


Figure 4.4: Simulation results for $\mathrm{NLAR}_3(4)$ true number of lags, $l = 4$. Vertical lines indicate 95% confidence intervals. Methods for which it is not possible to specify the number of change-points are colored in orange while methods where the number of change-points can be specified are colored in blue. DDL is represented by green dots.

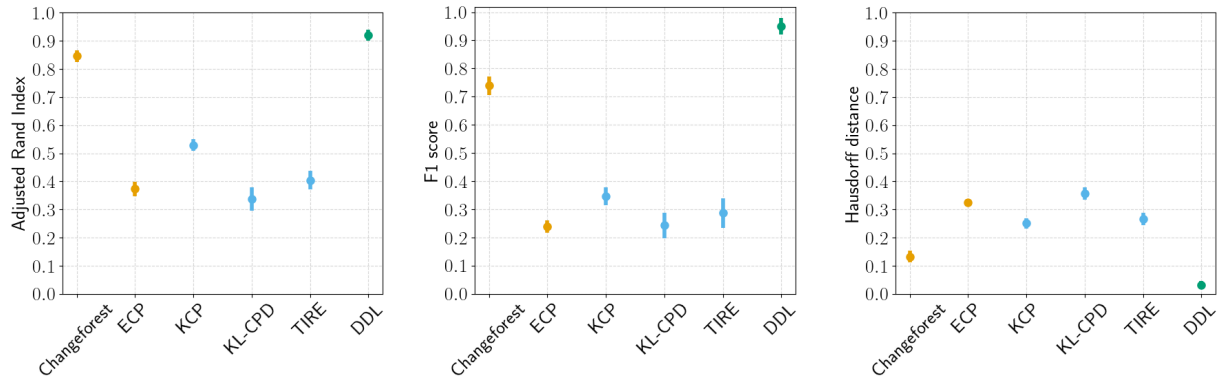


Figure 4.5: Simulation results for $\mathrm{VAR}_3(2)$ true number of lags, $l = 2$. Vertical lines indicate 95% confidence intervals. Methods for which it is not possible to specify the number of change-points are colored in orange while methods where the number of change-points can be specified are colored in blue. DDL is represented by green dots.

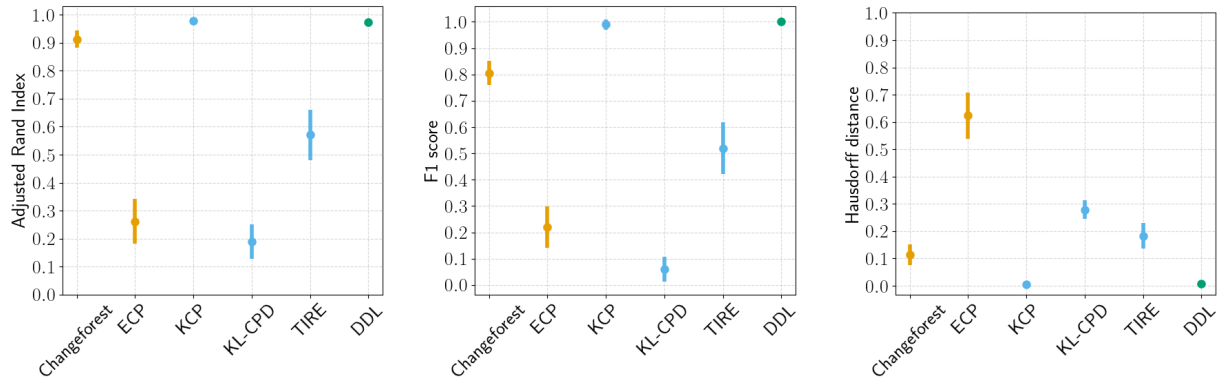


Figure 4.6: Simulation results for $\mathrm{NLARU}_2(4)$ with $l = 5$ lags. True number of lags are $l = 4$. Vertical lines indicate 95% confidence intervals. Methods for which it is not possible to specify the number of change-points are colored in orange while methods where the number of change-points can be specified are colored in blue. DDL is represented by green dots.

the second best ARI, F_1 scores, and Hausdorff distance in the $\mathrm{VAR}_3(2)$ setting and a close third behind KCP for the $\mathrm{NLARU}_2(4)$ model.

4.4 COVID-19 in New York City

We apply DDL, KCP, and `changeforest` to detect changes in COVID-19 outcomes in New York City. Daily data on deaths, cases, and hospitalizations in New York City due to COVID-19 are available starting Februrary 29, 2020 at City of New York's website. We specifically analyze data from February 29, 2020 to July 8, 2024 ($T = 1,592$). All data are first differenced to make them stationary and each variable is centered and scaled. This time period encompasses several major events such as the introduction of COVID vaccines on December 14, 2020, as well as two major variants: Delta and Omicron. Delta was recorded as the dominant COVID strain on July 23, 2021, while the Omicron variant become a major threat around December 19, 2021 (Millman, 2021; Garcia, 2021). Due to the stochastic nature of DDL we run it for 100 random seeds and plot the frequency of detected change-points in Figure 4.7. Note that the frequency is computed for each

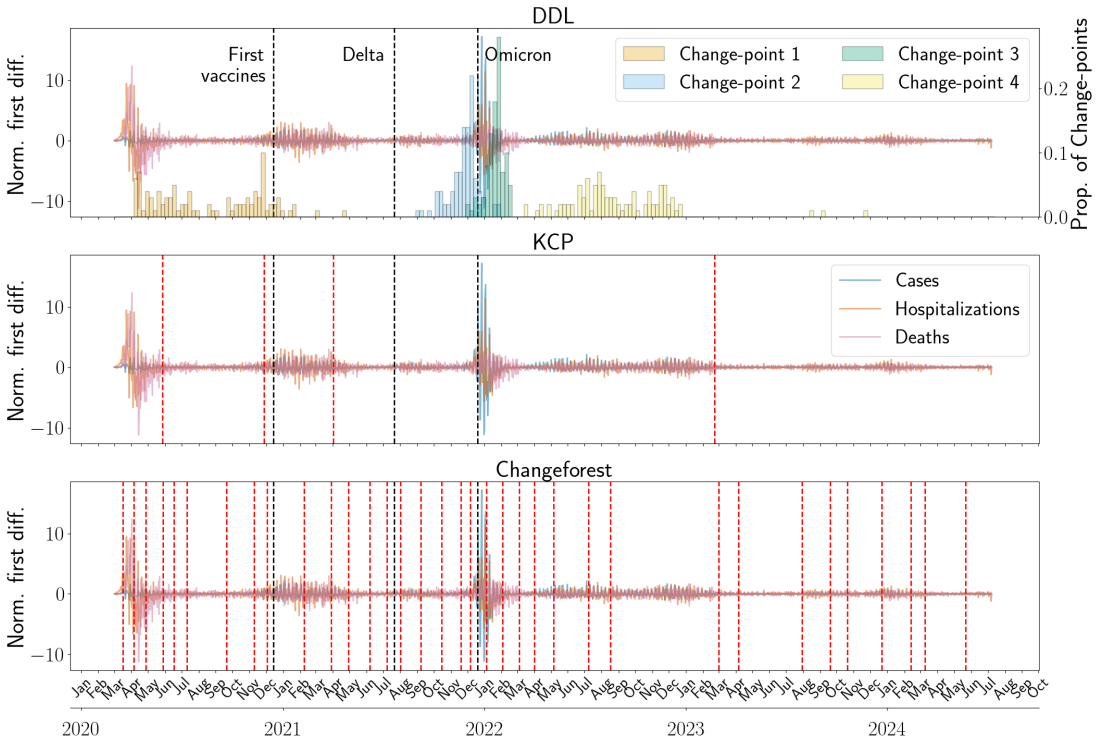


Figure 4.7: Comparison of change-point detection methods on COVID-19 outcomes in NYC. Normalized first differences of cases, hospitalizations, and deaths are plotted for all methods. Black dashed lines indicate notable events. Red dashed lines indicate estimated change-points for KCP or `changeforest`. For the DDL plot, histograms of estimated change-point locations are shown for each of the 4 change-points.

change-point separately. We use the same network setup as in Section 4.3.3 with 4 change-points and 300 epochs. For each day, we use data from the prior 7 days to predict outcomes which translates to $l = 7$ lags. KCP and `changeforest` are not stochastic methods so are only run once. We use the same setups for both methods as discussed in Section 4.3.2 and, similar to DDL, we specify a known number of 4 change-points for KCP.

While `changeforest` recovers the notable events, it also detects many other change-points.

Of course, since `changeforest` does not allow a fixed number of change-points, this could be a potential reason for the poor performance. KCP appears to recover the availability of vaccines well, but does not detect change-points around the Delta or Omicron variants. DDL appears to recover the start and end of the Omicron wave as shown by the histograms for change-points 2 and 3. However, similar to KCP, it does not detect any change-points near the Delta variant. One possible explanation is that the vaccines were effective for the Delta variant, but not Omicron. Indeed, Braeye et al. (2023) found that the BNT162b2 vaccine, known as the Pfizer-BioNTech vaccine, had a 87% effectiveness against transmisson for Delta, but only 31% for Omicron. The location of the first change-point estimated by DDL appears to be bimodal with a peak near the start of the COVID-19 pandemic and another peak near the first vaccinations. The final change-point appears roughly uniformly distributed between June 2022 and Feb 2023 likely signaling a steady-state for outcomes. Overall, when compared to competing methods, it appears that DDL recovers notable events in the COVID-19 pandemic reasonably well.

4.5 Discussion

This chapter introduces DDL, a novel and flexible deep learning framework for change-point detection. We introduce the core concepts behind DDL and evaluate its performance compared to existing competing deep learning methods, KL-CPD, `changeforest`, and TIRE, and other nonparametric benchmarks, KCP and ECP, across a variety of simulation settings. DDL outperforms all competing methods across these simulations with larger performance gaps in the most challenging settings. DDL was also compared to KCP and `changeforest`, the two top competing methods, in detection of change-points in COVID-19 outcomes in New York City. Overall, DDL recovered the locations of notable events, such as vaccine introduction and Delta and Omicron variants, better than competing methods.

While DDL allows for flexible deep learning architectures, its main limitation is the assumption of a fixed and known number of change-points. Extending this framework to allow for an unknown number of change-points is of primary interest in future work. In addition, the development of statistical theory for the DDL framework and extensions to the online setting are important avenues

for future work.

Chapter 5

DISCUSSION

In this thesis, we proposed novel methods for time series network analysis covering an array of topics. In Chapter 2, we introduced the flat loss phenomenon in VAR models which states that the population squared error loss is flat once the fitted order reaches or exceeds the true order. By replacing population estimates with sample estimates and adding a penalty on the fitted order, we developed a new criteria to select the order in VAR models, MIC. We showed under mild assumptions that MIC recovers the true VAR model order. In Chapter 3, we moved into frequency-domain network analyses and devised SDD, a procedure to directly estimate and generate confidence intervals for the difference in two inverse spectral densities. SDD is the first method of its kind in the frequency domain and is valid under varying degrees of data dependence. As a result, we expect SDD to be suitable across a wide range of data applications. Lastly, we approach time series network analysis from a deep learning perspective and propose dynamic deep learning (DDL) in Chapter 4. DDL is a novel deep learning framework that allows for changing predictive models. By casting the problem as a change-point detection problem and formulating change-points as learnable parameters of a deep learning model, DDL allows for the joint optimization of predictive models and change-points. While DDL can be used to study dynamic time series networks, its applicability extends to other forms of dependent data as well as independent data.

5.1 Directions for future work

5.1.1 Future directions for MIC

An interesting direction for future work is to extend the proposed method to high-dimensional settings. In high dimensions, we can substitute the least squares estimate by e.g. ridge or LASSO estimators. Alternatively, estimates of the autocovariances in high dimensions may be used and

plugged directly into the loss. It would then be interesting to compare the resulting estimator to recently proposed regularization-based approaches (Shojaie and Michailidis, 2010; Shojaie et al., 2012; Nicholson et al., 2017).

5.1.2 Future directions for DDL

While the DDL framework is quite general with regards to architecture, there are some important areas for future work. The primary direction of future work is to address the assumption of a fixed number of change-points. In practice, the number of change-points is unknown and needs to be estimated. Consider the case where the prediction function in regime j is parameterized by θ_j. For example, θ_j may be the weights and biases of a feed-forward neural network. One way to extend the DDL framework to allow for an unknown number of change-points would be to parameterize the prediction function in each regime as the prediction function in the first regime plus a set of differences, Δ_j. That is we would define $\theta_j = \theta_0 + \Delta_j$. Then, we can jointly learn $\theta_0, \Delta_1, \ldots, \Delta_{D-1}$ with a fused lasso penalty to encourage sparsity in Δ_j as in Safikhani and Shojaie (2022); Harchaoui and Lévy-Leduc (2010).

Another interesting avenue of future direction would be to extend DDL to the online setting. One idea would be to first learn prediction functions and change-point locations with a fixed dataset. Then, when it is ready to deploy in an online setting, the model parameters are frozen and another change-point is added at the end of the dataset. This change-point and the most recent prediction function would then be continuously trained and updated. If there were no change-point, one would expect the final change-point to stay near the end of the dataset. However, if a change occurs, we would expect the change-point to gradually shift away from the end of the dataset. A fusion-type penalty here would be important so that the most recent prediction function does not overfit the streaming data.

The last direction for future research is to develop statistical theory for this framework. Recently, Schmidt-Hieber (2020) showed that deep feed-forward ReLU networks excel at approximating regression functions that are compositions of smooth functions. Unfortunately, since the indicator function is not smooth, this theory cannot be directly applied. Even though we use a smooth

approximation to this function with the difference in sigmoid functions, this may not solve the problem entirely. Instead, by directly encoding this approximation of a non-smooth function into our model, we may be aiding the training of the neural network so that it only needs to learn the change-points and not the entire function. Using this idea, we believe the framework of Schmidt-Hieber (2020) could be extended to show that DDL achieves good approximations when the regression function is piecewise smooth.

In Section 4.2.2, we mentioned how it is possible for prediction functions to share parameters. We briefly discuss this further. One use case for this parameterization is when change-points occur in how an embedding space is used to predict the output. In this case, it is possible to use the same encoder, and thus the same learned embedding space, for each regime while allowing for separate decoders. A similar idea is used in Jones and Harchaoui (2020) except the authors detect changes in the embedding space itself rather than how that space is used by a decoder. Of course, this would result in gradients that are different from those is Section 4.2.2. Another interesting extension would be to use DDL in combination with variational autoencoders (VAEs) to learn any changes in the underlying probability distributions, similar to non-parametric methods like KCP (Arlot et al., 2019) and ECP (Matteson and James, 2014).

BIBLIOGRAPHY

Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory*, pages 267–81, 1973.

Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

Sylvain Arlot and Alain Celisse. Segmentation of the mean of heteroscedastic data via cross-validation. *Statistics and Computing*, 21(4):613–632, 2011.

Sylvain Arlot, Alain Celisse, and Zaid Harchaoui. A kernel multiple change-point algorithm via model selection. *Journal of machine learning research*, 20(162):1–56, 2019.

Peiliang Bai, Abolfazl Safikhani, and George Michailidis. Multiple change point detection in reduced rank high dimensional vector autoregressive models. *Journal of the American Statistical Association*, 118(544):2776–2792, 2023.

Yue Bai and Abolfazl Safikhani. A unified framework for change point detection in high-dimensional linear models. *Statistica Sinica*, 33:1721–1748, 2023.

Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

Douglas Bates and Dirk Eddelbuettel. Fast and elegant numerical linear algebra using the RcppEigen package. *Journal of Statistical Software*, 52(5):1–24, 2013. doi: 10.18637/jss.v052.i05.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron

Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR, 2018.

Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience*, 18 (24):10464–10472, 1998.

Peter J. Bickel, Ya'acov Ritov, and Alexandre B. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, 37(4):1705 – 1732, 2009. doi: 10.1214/08-AOS620. URL https://doi.org/10.1214/08-AOS620.

Julien Bloch, Alexander Greaves-Tunnell, Eric Shea-Brown, Zaid Harchaoui, Ali Shojaie, and Azadeh Yazdan-Shahmorad. Network structure mediates functional reorganization induced by optogenetic stimulation of non-human primate sensorimotor cortex. *Iscience*, 25(5), 2022.

Hilmar Böhm and Rainer von Sachs. Shrinkage estimation in the frequency domain of multivariate time series. *Journal of Multivariate Analysis*, 100(5):913–935, 2009.

Toon Braeye, Lucy Catteau, Ruben Brondeel, Joris AF van Loenhout, Kristiaan Proesmans, Laura Cornelissen, Herman Van Oyen, Veerle Stouten, Pierre Hubin, Matthieu Billuart, et al. Vaccine effectiveness against transmission of alpha, delta and omicron sars-cov-2-infection, belgian contact tracing, 2021–2022. *Vaccine*, 41(20):3292–3300, 2023.

Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

Peter J Brockwell and Richard A Davis. *Time series: theory and methods*. Springer science & business media, 1991.

Randy L Buckner, Jorge Sepulcre, Tanveer Talukdar, Fenna M Krienen, Hesheng Liu, Trey Hedden, Jessica R Andrews-Hanna, Reisa A Sperling, and Keith A Johnson. Cortical hubs revealed by intrinsic functional connectivity: mapping, assessment of stability, and relation to alzheimer's disease. *Journal of neuroscience*, 29(6):1860–1873, 2009.

Tony Cai, Weidong Liu, and Xi Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

Emmanuel Candes and Terence Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313 – 2351, 2007. doi: 10.1214/009053606000001523. URL https://doi.org/10.1214/009053606000001523.

Shubhadeep Chakraborty and Xianyang Zhang. High-dimensional change-point detection using generalized homogeneity metrics. *arXiv preprint arXiv:2105.08976*, 2021.

Wei-Cheng Chang, Chun-Liang Li, Yiming Yang, and Barnabás Póczos. Kernel change-point detection with auxiliary deep generative models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=r1GbfhRqF7.

Song Zan Chiou-Wei, Ching-Fu Chen, and Zhen Zhu. Economic growth and energy consumption revisited—evidence from linear and nonlinear granger causality. *Energy economics*, 30(6): 3063–3076, 2008.

Christian Conrad and Onno Kleen. Two are better than one: volatility forecasting using multiplicative component garch-midas models. *Journal of Applied Econometrics*, 35(1):19–45, 2020.

Rainer Dahlhaus. Graphical interaction models for multivariate time series. *Metrika*, 51:157–172, 2000.

Patrick Danaher, Pei Wang, and Daniela M Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(2):373–397, 2014.

Tim De Ryck, Maarten De Vos, and Alexander Bertrand. Change point detection in time series data using autoencoders with a time-invariant representation. *IEEE Transactions on Signal Processing*, 69:3513–3524, 2021.

Navonil Deb, Amy Kuceyeski, and Sumanta Basu. Regularized estimation of sparse spectral precision matrices. *arXiv preprint arXiv:2401.11128*, 2024.

Jean-François Ducré-Robitaille, Lucie A Vincent, and Gilles Boulet. Comparison of techniques for detection of discontinuities in temperature series. *International Journal of Climatology: A Journal of the Royal Meteorological Society*, 23(9):1087–1101, 2003.

Tom Dupré la Tour, Thomas Moreau, Mainak Jas, and Alexandre Gramfort. Multivariate convolutional sparse coding for electromagnetic brain signals. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Zahra Ebrahimzadeh, Min Zheng, Selcuk Karakas, and Samantha Kleinberg. Deep learning for multi-scale changepoint detection in multivariate time series. *arXiv preprint arXiv:1905.06913*, 2019.

Dirk Eddelbuettel and Romain François. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011. doi: 10.18637/jss.v040.i08.

Mark Fiecas, Chenlei Leng, Weidong Liu, and Yi Yu. Spectral analysis of high-dimensional time series. *Electronic Journal of Statistics*, 13(2):4079 – 4101, 2019. doi: 10.1214/19-EJS1621. URL https://doi.org/10.1214/19-EJS1621.

Rina Foygel and Mathias Drton. Extended bayesian information criteria for gaussian graphical models. *Advances in neural information processing systems*, 23, 2010.

Klaus Frick, Axel Munk, and Hannes Sieling. Multiscale change point inference. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(3):495–580, 2014.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Karl J Friston. Functional and effective connectivity: a review. *Brain connectivity*, 1(1):13–36, 2011.

Jean Gallier et al. The schur complement and symmetric positive semidefinite (and definite) matrices (2019). *URL https://www. cis. upenn. edu/jean/schur-comp. pdf*, 2020.

Deanna Garcia. De blasio tells biden: New york needs help now. *Politico*, 2021. URL https://www.politico.com/states/new-york/albany/story/2021/12/19/de-blasio-tells-biden-new-york-needs-help-now-1401826.

Abigail G Garrity, Godfrey D Pearlson, Kristen McKiernan, Dan Lloyd, Kent A Kiehl, and Vince D Calhoun. Aberrant "default mode" functional connectivity in schizophrenia. *American journal of psychiatry*, 164(3):450–457, 2007.

Théo Gnassounou, Rémi Flamary, and Alexandre Gramfort. Convolution monge mapping normalization for learning on sleep data. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 10457–10476. Curran Associates, Inc., 2023.

Edward J Hannan and Barry G Quinn. The determination of the order of an autoregression. *Journal of the Royal Statistical Society: Series B*, 41(2):190–195, 1979.

Zaıd Harchaoui and Céline Lévy-Leduc. Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association*, 105(492):1480–1493, 2010.

Christoffer Hatlestad-Hall, Trine Waage Rygvold, and Stein Andersson. Bids-structured resting-state electroencephalography (eeg) data extracted from an experimental paradigm. *Data in Brief*, 45:108647, 2022.

Michael Hellstern, Byol Kim, Zaid Harchaoui, and Ali Shojaie. Spectral differential network analysis for high-dimensional time series. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025. URL https://openreview.net/forum?id=tuHVCBN5fw.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

Jianhua Z Huang and Lijian Yang. Identification of non-linear additive autoregressive models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 66(2):463–477, 2004.

Ting-Ji Huang, Qi-Le Zhou, Han-Jia Ye, and De-Chuan Zhan. Change point detection via synthetic signals. In *International Workshop on Advanced Analytics and Learning on Temporal Data*, pages 25–35. Springer, 2023.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

Mikhail Hushchyn and Andrey Ustyuzhanin. Generalization of change-point detection in time series data based on direct density ratio estimation. *Journal of Computational Science*, 53:101385, 2021.

Nicholas A James and David S Matteson. ecp: An r package for nonparametric multiple change point analysis of multivariate data. *Journal of Statistical Software*, 62:1–25, 2015.

SR James, HA Knox, RE Abbott, and EJ Screaton. Improved moving window cross-spectral analysis for resolving large temporal seismic velocity changes in permafrost. *Geophysical Research Letters*, 44(9):4018–4026, 2017.

Adel Javanmard and Andrea Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909, 2014.

Iain M Johnstone and D Michael Titterington. Statistical challenges of high-dimensional data, 2009.

Corinne Jones and Zaïd Harchaoui. End-to-end learning for retrospective change-point estimation.

2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP), pages 1–6, 2020.

Maria Kalli and Jim E Griffin. Bayesian nonparametric vector autoregressive models. *Journal of econometrics*, 203(2):267–282, 2018.

Rohit Kanrar, Feiyu Jiang, and Zhanrui Cai. Model-free change-point detection using modern classifiers. *arXiv preprint arXiv:2404.06995*, 2024.

Haidar Khan, Lara Marcuse, and Bülent Yener. Deep density ratio estimation for change point detection. *arXiv preprint arXiv:1905.09876*, 2019.

Chang-Jin Kim, James C Morley, and Charles R Nelson. The structural break in the equity premium. *Journal of Business & Economic Statistics*, 23(2):181–191, 2005.

Takayoshi Kitaoka and Harutaka Takahashi. Improved prediction of new covid-19 cases using a simple vector autoregressive model: evidence from seven new york state counties. *Biology Methods and Protocols*, 8(1):bpac035, 2023.

Matej Kloska, Gabriela Grmanova, and Viera Rozinajova. Expert enhanced dynamic time warping based anomaly detection. *Expert Systems with Applications*, 225:120030, 2023.

Lena Koepcke, Go Ashida, and Jutta Kretzberg. Single and multiple change point detection in spike trains: Comparison of different cusum methods. *Frontiers in systems neuroscience*, 10:51, 2016.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Denis Kwiatkowski, Peter CB Phillips, Peter Schmidt, and Yongcheol Shin. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics*, 54(1-3):159–178, 1992.

Lucas C Laurindo, Leo Siqueira, Arthur J Mariano, and Ben P Kirtman. Cross-spectral analysis of the sst/10-m wind speed coupling resolved by satellite products and climate model simulations. *Climate dynamics*, 52(9):5071–5098, 2019.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Jie Li, Paul Fearnhead, Piotr Fryzlewicz, and Tengyao Wang. Automatic change-point detection in time series via deep learning. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(2):273–285, 2024.

Malte Londschien, Peter Bühlmann, and Solt Kovács. Random forests for change point detection. *Journal of Machine Learning Research*, 24(216):1–45, 2023.

Tzon-Tzer Lu and Sheng-Hua Shiou. Inverses of 2$\times$ 2 block matrices. *Computers & Mathematics with Applications*, 43(1-2):119–129, 2002.

Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

Christopher J Markiewicz, Krzysztof J Gorgolewski, Franklin Feingold, Ross Blair, Yaroslav O Halchenko, Eric Miller, Nell Hardcastle, Joe Wexler, Oscar Esteban, Mathias Goncavles, et al. The openneuro resource for sharing of neuroscience data. *Elife*, 10:e71774, 2021.

David S Matteson and Nicholas A James. A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109(505):334–345, 2014.

Jennifer Millman. Delta replacing all other nyc covid strains as new case average soars 32%. *NBC New York*, 2021. URL https://www.nbcnewyork.com/news/coronavirus/delta-replacing-all-other-nyc-covid-strains-as-new-case-average-soars-32/3170311/.

Robb J Muirhead. *Aspects of multivariate statistical theory*. John Wiley & Sons, 2009.

Sahand N. Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A Unified Framework for High-Dimensional Analysis of M-Estimators with Decomposable Regularizers. *Statistical Science*, 27(4):538 – 557, 2012. doi: 10.1214/12-STS400. URL https://doi.org/10.1214/12-STS400.

Matey Neykov, Yang Ning, Jun S. Liu, and Han Liu. A Unified Theory of Confidence Regions and Testing for High-Dimensional Estimating Equations. *Statistical Science*, 33(3):427 – 443, 2018. doi: 10.1214/18-STS661. URL https://doi.org/10.1214/18-STS661.

William B Nicholson, David S Matteson, and Jacob Bien. VARX-L: Structured regularization for large vector autoregressions with exogenous variables. *International Journal of Forecasting*, 33 (3):627–651, 2017.

William B Nicholson, Ines Wilms, Jacob Bien, and David S Matteson. High dimensional forecasting via interpretable vector autoregression. *Journal of Machine Learning Research*, 21(166):1–52, 2020.

Sou Nobukawa, Mitsuru Kikuchi, and Tetsuya Takahashi. Changes in functional connectivity dynamics with aging: a dynamical phase synchronization approach. *Neuroimage*, 188:357–368, 2019.

Oscar Hernan Madrid Padilla, Yi Yu, Daren Wang, and Alessandro Rinaldo. Optimal nonparametric multivariate change point detection and localization. *IEEE Transactions on Information Theory*, 68(3):1922–1944, 2021.

Ewan S Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

Ewan Stafford Page. A test for a change in a parameter occurring at an unknown point. *Biometrika*, 42(3/4):523–527, 1955.

Jostein Paulsen and Dag Tjøstheim. On the estimation of residual variance and order in autoregressive time series. *Journal of the Royal Statistical Society: Series B*, 47(2):216–228, 1985.

Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

Barry G Quinn. Order determination for a multivariate autoregression. *Journal of the Royal Statistical Society: Series B*, 42(2):182–185, 1980.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021. URL https://www.R-project.org/.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Hugo Richard, Luigi Gresele, Aapo Hyvarinen, Bertrand Thirion, Alexandre Gramfort, and Pierre Ablin. Modeling shared responses in neuroimaging studies through multiview ica. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19149–19162. Curran Associates, Inc., 2020.

Artem Ryzhikov, Mikhail Hushchyn, and Denis Derkach. Latent stochastic differential equations for change point detection. *IEEE Access*, 11:104700–104711, 2023.

Abolfazl Safikhani and Ali Shojaie. Joint structural break detection and parameter estimation in high-dimensional nonstationary var models. *Journal of the American Statistical Association*, 117 (537):251–264, 2022.

Abolfazl Safikhani, Yue Bai, and George Michailidis. Fast and scalable algorithm for detection of structural breaks in big var models. *Journal of Computational and Graphical Statistics*, 31(1): 176–189, 2022.

Said E Said and David A Dickey. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71(3):599–607, 1984.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4):1875 – 1897, 2020. doi: 10.1214/19-AOS1875. URL https://doi.org/10.1214/19-AOS1875.

Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, pages 461–464, 1978.

Anil K Seth, Adam B Barrett, and Lionel Barnett. Granger causality analysis in neuroscience and neuroimaging. *Journal of Neuroscience*, 35(8):3293–3297, 2015.

Ali Shojaie and Emily B Fox. Granger causality: A review and recent advances. *Annual Review of Statistics and Its Application*, 9(1):289–319, 2022.

Ali Shojaie and George Michailidis. Analysis of gene sets based on the underlying regulatory network. *Journal of Computational Biology*, 16(3):407–426, 2009.

Ali Shojaie and George Michailidis. Discovering graphical granger causality using the truncating lasso penalty. *Bioinformatics*, 26(18):i517–i523, 2010.

Ali Shojaie, Sumanta Basu, and George Michailidis. Adaptive thresholding for reconstructing regulatory networks from time-course gene expression data. *Statistics in Biosciences*, 4:66–83, 2012.

Christopher A Sims. Macroeconomics and reality. *Econometrica*, pages 1–48, 1980.

Bumho Son, Yunyoung Lee, Seongwan Park, and Jaewook Lee. Forecasting global stock market volatility: The impact of volatility spillover index in spatial-temporal graph-based model. *Journal of Forecasting*, 42(7):1539–1559, 2023.

Cornelis J Stam, BF Jones, Guido Nolte, Michael Breakspear, and Ph Scheltens. Small-world networks and functional connectivity in alzheimer's disease. *Cerebral cortex*, 17(1):92–99, 2007.

Yiming Sun, Yige Li, Amy Kuceyeski, and Sumanta Basu. Large spectral density matrix estimation by thresholding. *arXiv preprint arXiv:1812.00532*, 2018.

Mátyás A Sustik and Ben Calderhead. Glassofast: an efficient glasso implementation. *UTCS Technical Report TR-12-29 2012*, 2012.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.

Mauro Ursino, Giulia Ricci, and Elisa Magosso. Transfer entropy as a measure of brain connectivity: A critical analysis with the help of neural mass models. *Frontiers in computational neuroscience*, 14:45, 2020.

Gerrit JJ Van den Burg and Christopher KI Williams. An evaluation of change point detection algorithms. *arXiv preprint arXiv:2003.06222*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

L. Yu. Vostrikova. Detecting "disorder" in multidimensional random processes. *Doklady Akademii Nauk SSSR*, 259(2):270–274, 1981. http://mi.mathnet.ru/dan44582.

Tengyao Wang and Richard J Samworth. High dimensional change point estimation via sparse projection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80(1): 57–83, 2018.

Yue Wang, Jing Ma, and Ali Shojaie. Direct estimation of differential granger causality between two high-dimensional time series. *arXiv preprint arXiv:2109.07609*, 2021.

Wei Biao Wu. Nonlinear system theory: Another look at dependence. *Proceedings of the National Academy of Sciences*, 102(40):14150–14154, 2005.

Wei Biao Wu and Paolo Zaffaroni. Asymptotic theory for spectral density estimates of general multivariate time series. *Econometric Theory*, 34(1):1–22, 2018.

Azadeh Yazdan-Shahmorad, Camilo Diaz-Botia, Timothy L Hanson, Viktor Kharazia, Peter Ledochowitsch, Michel M Maharbiz, and Philip N Sabes. A large-scale interface for optogenetic stimulation and recording in nonhuman primates. *Neuron*, 89(5):927–939, 2016.

Azadeh Yazdan-Shahmorad, Daniel B Silversmith, Viktor Kharazia, and Philip N Sabes. Targeted cortical reorganization using optogenetics in non-human primates. *Elife*, 7:e31034, 2018.

Huili Yuan, Ruibin Xi, Chong Chen, and Minghua Deng. Differential network analysis via lasso penalized d-trace loss. *Biometrika*, 104(4):755–770, 2017.

Andrew Zalesky, Alex Fornito, Luca Cocchi, Leonardo L Gollo, and Michael Breakspear. Time-resolved resting-state brain networks. *Proceedings of the National Academy of Sciences*, 111(28): 10341–10346, 2014.

Chi Zhang and Danna Zhang. Spectral inference for high dimensional time series. *IEEE Transactions on Information Theory*, 2025.

Peng Zhao and Bin Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.

Appendix A

APPENDICES FOR CHAPTER 2

A.1 Assumptions

In this section we list all assumptions used in our analyses and provide a brief discussion of each. We view all these assumptions as relatively mild.

Assumption 2. *$Z_t \in \mathbb{R}^{k\times 1}$ is a stable mean zero process. That is* $\det(I_k - A_1 z - \cdots - A_{p_0} z^{p_0}) \neq 0$ *for* $|z| \leq 1$.

This stability condition is standard and is identical to that used in Lütkepohl (2005, Eq. (2.1.12)). Note that stability implies stationarity (Lütkepohl, 2005, Proposition 2.1) and this assumption is required to replace second order expectations with autocovariances. That is, this is required to have $\mathbb{E}\left(Z_t Z_{t-h}^T\right) = \Gamma_h$. This is also required to use the Yule-Walker equations.

Assumption 3.

$$\mathbb{E}\left(X_{p_{\max}} X_{p_{\max}}^T\right) = \begin{bmatrix} \Gamma_0 & \Gamma_1 & \dots & \Gamma_{p_{\max}-1} \\ \Gamma_1^T & \Gamma_0 & \dots & \Gamma_{p_{\max}-2} \\ \vdots & & & \vdots \\ \Gamma_{p_{\max}-1}^T & \Gamma_{p_{\max}-2}^T & \dots & \Gamma_0 \end{bmatrix},$$

is invertible.

Note that due to the quadratic form of $\mathbb{E}\left(X_{p_{\max}} X_{p_{\max}}^T\right)$, this matrix is symmetric and positive semidefinite. By assuming invertibility, we ensure this matrix is also positive definite. We also only need to make this assumption for $p_{\max}$ and we will have positive definiteness for all $\mathbb{E}\left(X_{p_{\max}} X_{p_{\max}}^T\right)$ for $i = 1, \dots p_{\max}$ since a matrix is positive definite if and only if all its principle minors are positive (see Lütkepohl (2005) Appendix A.8.3). The ki^{th} principle minor of $\mathbb{E}\left(X_{p_{\max}} X_{p_{\max}}^T\right)$ is $\det\left(\mathbb{E}\left(X_i X_i^T\right)\right)$. Again, $\mathbb{E}\left(X_i X_i^T\right)$ is symmetric and positive semi-definite so ensuring a positive

determinant ensures strictly positive eigenvalues and thus positive definiteness. In all simulation settings this assumption has been met.

Assumption 4. *We assume that for a VAR(p_0) process when $p < p_0$,*

$$\begin{bmatrix} \Gamma_1 & \dots & \Gamma_{p-1} \end{bmatrix} \begin{bmatrix} \Gamma_0 & \dots & \Gamma_{p-1} \\ \vdots & & \vdots \\ \Gamma_{p-1}^T & \dots & \Gamma_0 \end{bmatrix}^{-1} \begin{bmatrix} \Gamma_1^T \\ \vdots \\ \Gamma_p^T \end{bmatrix} - \Gamma_p \neq 0.$$

This assumption essentially states that we need at least p_0 lags to generate the $p\text{th}$ autocovariance Γ_p. We view this assumption as very mild. For example it is implicitly made in Lütkepohl (2005, Eq. (2.1.37)) where for a VAR(p_0) process, the autocovariance matrix is determined by the prior p_0 lags.

A.2 Simplifying multivariate loss

In this section we show that the profiled loss can be written as

$$\mathbb{E}\left[\left(Y_t - A_p^* X_p\right)^T \left(Y_t - A_p^* X_p\right)\right] = \mathrm{Tr}\left(\Gamma_0\right) - \mathrm{Tr}\left(\begin{bmatrix} \Gamma_1 & \dots & \Gamma_p \end{bmatrix} \begin{bmatrix} \Gamma_0 & \dots & \Gamma_{p-1} \\ \vdots & & \vdots \\ \Gamma_{p-1}^T & \dots & \Gamma_0 \end{bmatrix}^{-1} \begin{bmatrix} \Gamma_1^T \\ \vdots \\ \Gamma_p^T \end{bmatrix} \right).$$

To write the profiled loss in this way we will need to replace expectations with atuocovariances as in $\mathbb{E}\left(Z_t Z_{t-h}^T\right) = \Gamma_h$. Thus Assumption 2 (stability) is required.

A.2.1 Simplifying equalities

To begin we first compute some expectations that will be needed to simplify the loss. We make note of several identities we will use. First, $\mathrm{vec}(A)^T \mathrm{vec}(B) = \mathrm{Tr}(A^T B)$ and $\mathrm{vec}(ABC) = (C^T \otimes A)\mathrm{vec}(B)$. The j^{th} row and column of Γ_i are denoted as $\Gamma_{i,j\cdot}$ and $\Gamma_{i,\cdot j}$ respectively.

We also note that $\mathbb{E}\left(X_p^T \otimes X_p^T\right) \in \mathbb{R}^{1\times k^2p^2}$ and can be written as

$$\mathbb{E}\left(X_p^T \otimes X_p^T\right) = \mathbb{E}\left(\begin{bmatrix} Z_{t-1,1} & \dots & Z_{t-1,k} & Z_{t-2,1} & \dots & Z_{t-p,k}\end{bmatrix} \otimes \begin{bmatrix} Z_{t-1,1} & \dots & Z_{t-1,k} & Z_{t-2,1} & \dots & Z_{t-p,k}\end{bmatrix}\right)$$

$$= \mathbb{E}\begin{pmatrix} [\, Z_{t-1,1}^2 & \dots & Z_{t-1,1}Z_{t-1,k} & Z_{t-1,1}Z_{t-2,1} & \dots & Z_{t-1,1}Z_{t-2,k} & \dots & Z_{t-1,1}Z_{t-p,k} \\ Z_{t-1,2}Z_{t-1,1} & \dots & Z_{t-1,2}Z_{t-1,k} & Z_{t-1,2}Z_{t-2,1} & \dots & Z_{t-1,2}Z_{t-2,k} & \dots & Z_{t-1,2}Z_{t-p,k} \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ Z_{t-1,k}Z_{t-1,1} & \dots & Z_{t-1,k}Z_{t-1,k} & Z_{t-1,k}Z_{t-2,1} & \dots & Z_{t-1,k}Z_{t-2,k} & \dots & Z_{t-1,k}Z_{t-p,k} \\ Z_{t-2,1}Z_{t-1,1} & \dots & Z_{t-2,1}Z_{t-1,k} & Z_{t-2,1}Z_{t-2,1} & \dots & Z_{t-2,1}Z_{t-2,k} & \dots & Z_{t-2,1}Z_{t-p,k} \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots\,] \end{pmatrix}.$$

We have

$$\mathbb{E}\left(\begin{bmatrix} Z_{t-1,1}^2 & \dots & Z_{t-1,1}Z_{t-1,k}\end{bmatrix}^T\right) = \Gamma_{0,\cdot 1}.$$

Similarly,

$$\mathbb{E}\left(\begin{bmatrix} Z_{t-1,1}Z_{t-2,1} & \dots & Z_{t-1,1}Z_{t-2,k}\end{bmatrix}^T\right) = \Gamma_{-1,\cdot 1} = (\Gamma_1^T)_{\cdot 1}\,,$$

$$\mathbb{E}\left(\begin{bmatrix} Z_{t-1,1}Z_{t-p,1} & \dots & Z_{t-1,1}Z_{t-p,k}\end{bmatrix}^T\right) = \left(\Gamma_{p-1}^T\right)_{\cdot 1}\,.$$

We define

$$C := \begin{matrix} [\Gamma_{0,\cdot 1} & (\Gamma_1^T)_{\cdot 1} & (\Gamma_2^T)_{\cdot 1} & \dots & (\Gamma_{p-1}^T)_{\cdot 1} \\ \Gamma_{0,\cdot 2} & (\Gamma_1^T)_{\cdot 2} & (\Gamma_2^T)_{\cdot 2} & \dots & (\Gamma_{p-1}^T)_{\cdot 2} \\ \vdots & & & & \vdots \\ \Gamma_{0,\cdot k} & (\Gamma_1^T)_{\cdot k} & (\Gamma_2^T)_{\cdot k} & \dots & (\Gamma_{p-1}^T)_{\cdot k} \\ \Gamma_{1,\cdot 1} & \Gamma_{0,\cdot 1} & (\Gamma_1^T)_{\cdot 1} & \dots & (\Gamma_{p-1}^T)_{\cdot k} \\ \vdots & & & & \vdots \\ \Gamma_{p-1,\cdot k} & \Gamma_{p-2,\cdot k} & \Gamma_{p-3,\cdot k} & \dots & \Gamma_{0,\cdot k}]\,. \end{matrix}$$

Note that each element in C is $\in \mathbb{R}^{k\times 1}$. For example, both $\Gamma_{0,\cdot 1}$ and $(\Gamma_1)_{\cdot 1}^T$ are $\in \mathbb{R}^{k\times 1}$. This gives us that $C \in \mathbb{R}^{k\times kp^2}$. Therefore we can write

$$\mathbb{E}\left(X_p^T \otimes X_p^T\right) = \text{vec}(C)^T$$
$$= \text{vec}\left(\begin{bmatrix} \Gamma_0 & \Gamma_1 & \dots & \Gamma_{p-1} \\ \Gamma_1^T & \Gamma_0 & \dots & \Gamma_{p-2} \\ \vdots & & & \vdots \\ \Gamma_{p-1}^T & \Gamma_{p-2}^T & \dots & \Gamma_0 \end{bmatrix}\right)^T.$$

Using a similar idea we have that

$$\mathbb{E}\left(X_p^T \otimes Y_t^T\right) = \mathbb{E}\left(\begin{bmatrix} Z_{t-1,1} & \dots & Z_{t-1,k} & Z_{t-2,1} & \dots & Z_{t-p,k} \end{bmatrix} \otimes \begin{bmatrix} Z_{t,1} & \dots & Z_{t,k} \end{bmatrix}\right)$$
$$= \text{vec}\left(\begin{bmatrix} \Gamma_1 & \dots & \Gamma_p \end{bmatrix}\right)^T.$$

for space considerations, we do not write out the steps in detail. The method proceeds similarly to the above. Next,

$$\mathbb{E}\left(X_p X_p^T\right) = \mathbb{E}\left(\begin{bmatrix} Z_{t-1} \\ Z_{t-2} \\ \vdots \\ Z_{t-p} \end{bmatrix} \begin{bmatrix} Z_{t-1}^T & Z_{t-2}^T & \dots & Z_{t-p}^T \end{bmatrix}\right) = \mathbb{E}\left(\begin{bmatrix} Z_{t-1}Z_{t-1}^T & Z_{t-1}Z_{t-2}^T & \dots & Z_{t-1}Z_{t-p}^T \\ Z_{t-2}Z_{t-1}^T & Z_{t-2}Z_{t-2}^T & \dots & Z_{t-2}Z_{t-2}^T \\ \vdots & & & \vdots \\ Z_{t-p}Z_{t-1}^T & Z_{t-p}Z_{t-2}^T & \dots & Z_{t-p}Z_{t-p}^T \end{bmatrix}\right)$$
$$= \begin{bmatrix} \Gamma_0 & \Gamma_1 & \dots & \Gamma_{p-1} \\ \Gamma_1^T & \Gamma_0 & \dots & \Gamma_{p-2} \\ \vdots & & & \vdots \\ \Gamma_{p-1}^T & \Gamma_{p-2}^T & \dots & \Gamma_0 \end{bmatrix}.$$

Lastly,

$$\mathbb{E}\left(Y_t X_p^T\right) = \begin{bmatrix} \Gamma_1 & \dots & \Gamma_p \end{bmatrix}.$$

A.2.2 Simplifying the loss

We can now proceed in simplifying the population loss. First, note that

$$\begin{aligned}\mathbb{E}\left[\left(Y_t - A_p^* X_p\right)^T \left(Y_t - A_p^* X_p\right)\right] &= \mathbb{E}\left(Y_t^T Y_t\right) - 2\mathbb{E}\left(Y_t^T A_p^* X_p\right) + \mathbb{E}\left(X_p^T A_p^{*T} A_p^* X_p\right) \\ &= \operatorname{Tr}\left(\Gamma_0\right) - 2\mathbb{E}\left(X_p^T \otimes Y_t^T\right) \operatorname{vec}\left(A_p^*\right) + \mathbb{E}\left(X_p^T \otimes X_p^T\right) \operatorname{vec}\left(A_p^{*T} A_p^*\right) \\ &= \operatorname{Tr}\left(\Gamma_0\right) - 2\operatorname{Tr}\left(\begin{bmatrix}\Gamma_1^T \\ \vdots \\ \Gamma_p^T\end{bmatrix} A_p^*\right) + \operatorname{Tr}\left(\begin{bmatrix}\Gamma_0 & \dots & \Gamma_{p-1} \\ \vdots & & \vdots \\ \Gamma_{p-1}^T & \dots & \Gamma_0\end{bmatrix} A_p^{*T} A_p^*\right).\end{aligned}$$

Since $A_p^{*T} = \mathbb{E}\left(X_p X_p^T\right)^{-1,T} E\left(Y_t X_p^T\right)^T = \begin{bmatrix}\Gamma_0 & \dots & \Gamma_{p-1} \\ \vdots & & \vdots \\ \Gamma_{p-1}^T & \dots & \Gamma_0\end{bmatrix}^{-1,T} \begin{bmatrix}\Gamma_1^T \\ \vdots \\ \Gamma_p^T\end{bmatrix}$ and $\begin{bmatrix}\Gamma_0 & \dots & \Gamma_{p-1} \\ \vdots & & \vdots \\ \Gamma_{p-1}^T & \dots & \Gamma_0\end{bmatrix}$ is symmetric and so is its inverse and using $\operatorname{Tr}(A^T B) = \operatorname{Tr}(AB^T)$ we get that

$$\begin{aligned}\mathbb{E}\left[\left(Y_t - A_p^* X_p\right)^T \left(Y_t - A_p^* X_p\right)\right] &= \operatorname{Tr}\left(\Gamma_0\right) - \operatorname{Tr}\left(\begin{bmatrix}\Gamma_1^T \\ \vdots \\ \Gamma_p^T\end{bmatrix} A_p^*\right) \\ &= \operatorname{Tr}\left(\Gamma_0\right) - \operatorname{Tr}\left(\begin{bmatrix}\Gamma_1 & \dots & \Gamma_p\end{bmatrix} \begin{bmatrix}\Gamma_0 & \dots & \Gamma_{p-1} \\ \vdots & & \vdots \\ \Gamma_{p-1}^T & \dots & \Gamma_0\end{bmatrix}^{-1} \begin{bmatrix}\Gamma_1^T \\ \vdots \\ \Gamma_p^T\end{bmatrix}\right).\end{aligned}$$

A.3 Proofs

In this section we prove Theorems 1 and 2 and Corollary 1. Specifically, Appendix A.3.1 focuses on Theorem 1, Appendix A.3.2 proves Corollary 1, and Appendix A.3.3 proves Theorem 2.

A.3.1 Flat loss

In this section we prove Theorem 1 which establishes that the loss decreases until the true order p_0 at which point it remains constant at $\mathrm{Tr}(\Sigma_\epsilon)$. The proof proceeds in several steps. We first relate the loss at fitted order p to the loss at fitted order $p-1$. In the second step we consider the cases when $p > p_0$ and $p \leq p_0$ separately. Finally we show that the loss at p_0 is equal to $\mathrm{Tr}(\Sigma_\epsilon)$.

Proof of Theorem 1. Recall,

$$L_{\mathrm{VAR}}(p) := \mathrm{Tr}\left(\Gamma_0\right) - \mathrm{Tr}\left(\begin{bmatrix}\Gamma_1 & \dots & \Gamma_p\end{bmatrix}\begin{bmatrix}\Gamma_0 & \dots & \Gamma_{p-1}\\ \vdots & & \vdots \\ \Gamma_{p-1}^T & \dots & \Gamma_0\end{bmatrix}^{-1}\begin{bmatrix}\Gamma_1^T \\ \vdots \\ \Gamma_p^T\end{bmatrix}\right).$$

To express $L_{\mathrm{VAR}}(p)$ as a function of $L_{\mathrm{VAR}}(p-1)$ we partition the matrices as follows

$$\left[\begin{array}{ccc|c}\Gamma_1 & \dots & \Gamma_{p-1} & \Gamma_p\end{array}\right] := \begin{bmatrix} g & \Gamma_p\end{bmatrix}.$$

$$\left[\begin{array}{ccc|c}\Gamma_0 & \dots & \Gamma_{p-2} & \Gamma_{p-1}\\ \Gamma_1^T & \dots & \Gamma_{p-3} & \Gamma_{p-2}\\ \vdots & & \vdots & \vdots \\ \Gamma_{p-2}^T & \dots & \Gamma_0 & \vdots \\ \hline \Gamma_{p-1}^T & \dots & \Gamma_1^T & \Gamma_0\end{array}\right] := \begin{bmatrix} B & C^T \\ C & D\end{bmatrix}.$$

Now note from the 2 x 2 block matrix inversion formula from Lu and Shiou (2002),

$$\begin{bmatrix}\Gamma_0 & \dots & \Gamma_{p-1}\\ \vdots & & \vdots \\ \Gamma_{p-1}^T & \dots & \Gamma_0\end{bmatrix}^{-1} = \begin{bmatrix} B & C^T \\ C & D\end{bmatrix}^{-1} = \begin{bmatrix} B^{-1} + B^{-1}C^THCB^{-1} & -B^{-1}C^TH \\ -HCB^{-1} & H\end{bmatrix},$$

where $H = (D - CB^{-1}C^T)^{-1}$ is the inverse of the Schur-complement. We carry out the multiplication

$$\begin{bmatrix}\Gamma_1 & \dots & \Gamma_p\end{bmatrix}\begin{bmatrix}\Gamma_0 & \dots & \Gamma_{p-1}\\ \vdots & & \vdots \\ \Gamma_{p-1}^T & \dots & \Gamma_0\end{bmatrix}^{-1}\begin{bmatrix}\Gamma_1^T \\ \vdots \\ \Gamma_p^T\end{bmatrix} = \begin{bmatrix} g & \Gamma_p\end{bmatrix}\begin{bmatrix} B^{-1} + B^{-1}C^THCB^{-1} & -B^{-1}C^TH \\ -HCB^{-1} & H\end{bmatrix}\begin{bmatrix} g^T \\ \Gamma_p^T\end{bmatrix}$$

$$= gB^{-1}g^T + gB^{-1}C^THCB^{-1}g^T - \Gamma_pHCB^{-1}g^T - gB^{-1}C^TH\Gamma_p^T + \Gamma_pH\Gamma_p^T\,.$$

Thus we get

$$\begin{aligned}
L_{\mathrm{VAR}}(p) &= \mathrm{Tr}\left(\Gamma_0\right) - \mathrm{Tr}\left(gB^{-1}g^T + gB^{-1}C^THCB^{-1}g^T - \Gamma_pHCB^{-1}g^T - gB^{-1}C^TH\Gamma_p^T + \Gamma_pH\Gamma_p^T\right)\\
&= L_{\mathrm{VAR}}(p-1) - \mathrm{Tr}\left(gB^{-1}C^THCB^{-1}g^T - \Gamma_pHCB^{-1}g^T - gB^{-1}C^TH\Gamma_p^T + \Gamma_pH\Gamma_p^T\right)\\
&= L_{\mathrm{VAR}}(p-1) - \mathrm{Tr}\left(gB^{-1}C^THCB^{-1}g^T\right) + \mathrm{Tr}\left(\Gamma_pHCB^{-1}g^T\right) + \mathrm{Tr}\left(gB^{-1}C^TH\Gamma_p^T\right) - \mathrm{Tr}\left(\Gamma_pH\Gamma_p^T\right)\\
&= L_{\mathrm{VAR}}(p-1) - \mathrm{Tr}\left(\left(gB^{-1}C^T - \Gamma_p\right)H\left(gB^{-1}C^T - \Gamma_p\right)^T\right)\,.
\end{aligned}$$

Note that this relationship holds in general. To get this result we only rely on Assumption 2 (stability) and Assumption 3 (invertibility).

Now we use that the true process is VAR(p_0) to study the loss when $p > p_0$. If $p > p_0$, from the Yule-Walker equations,

$$\begin{bmatrix}\Gamma_1 & \dots & \Gamma_{p-1}\end{bmatrix} = \begin{bmatrix}A_1 & \dots & A_{p_0}\end{bmatrix}\begin{bmatrix}\Gamma_0 & \Gamma_1 & \dots & \Gamma_{p-2}\\ \Gamma_{-1} & \Gamma_0 & \dots & \Gamma_{p-3}\\ \vdots & & & \vdots\\ \Gamma_{-(p_0-1)} & \Gamma_{-(p_0-2)} & \dots & \Gamma_{-(p_0-(p-1))}\end{bmatrix}\,.$$

We can extend this to

$$\begin{bmatrix}\Gamma_1 & \dots & \Gamma_{p-1}\end{bmatrix} = \begin{bmatrix}A_1 & \dots & A_{p_0} & 0 & \dots & 0\end{bmatrix}\begin{bmatrix}\Gamma_0 & \Gamma_1 & \dots & \Gamma_{p-2}\\ \Gamma_{-1} & \Gamma_0 & \dots & \Gamma_{p-3}\\ \vdots & & & \vdots\\ \Gamma_{2-p} & \Gamma_{3-p} & \dots & \Gamma_0\end{bmatrix}\,.$$

Using that $\Gamma_{-i} = \Gamma_i^T$ and substituting in the definitions of B and g and under Assumption 3 (invertibility),

$$gB^{-1}C^T = \begin{bmatrix}A_1 & \dots & A_{p_0} & 0 & \dots & 0\end{bmatrix}\begin{bmatrix}\Gamma_{p-1}\\ \vdots\\ \Gamma_1\end{bmatrix} = \Gamma_p\,.$$

Thus we get for any $p > p_0$

$$L_{\mathrm{VAR}}(p) = L_{\mathrm{VAR}}(p-1)\,.$$

Next we study the loss when $p \leq p_0$. From Assumption 3 (invertibility)

$$\begin{bmatrix} \Gamma_0 & \dots & \Gamma_{p-1} \\ \vdots & & \vdots \\ \Gamma_{p-1}^T & \dots & \Gamma_0 \end{bmatrix},$$

is positive definite for all $p = 1, \dots, p_{\max}$. From Proposition 2.2 of Gallier et al. (2020), the Schur-Complement is also positive definite and hence so is the inverse, H. From Assumption 4 (irreducibility) when $p \leq p_0$, in general,

$$gB^{-1}C^T - \Gamma_p := \Delta \neq 0\,.$$

Recall our equation relating $L_{\text{VAR}}(p)$ to $L_{\text{VAR}}(p-1)$:

$$L_{\text{VAR}}(p) = L_{\text{VAR}}(p-1) - \text{Tr}\left(\left(gB^{-1}C^T - \Gamma_p\right) H \left(gB^{-1}C^T - \Gamma_p\right)^T\right).$$

Using δ_i^T to denote the i^{th} row of Δ,

$$\text{Tr}\left(\Delta H \Delta^T\right) = \sum_{i=1}^{k} \delta_i^T H \delta_i,$$

since H is positive definite and $\delta_i \neq 0$ for at least one $i = 1, \dots, k$, $\text{Tr}\left(\Delta H \Delta^T\right) > 0$ and thus

$$L_{\text{VAR}}(p) < L_{\text{VAR}}(p-1)\,.$$

Lastly we show that the loss flattens out at $\text{Tr}(\Sigma_\epsilon)$. Since $L_{\text{VAR}}(p) = L_{\text{VAR}}(p-1)$ when $p > p_0$ it suffices to show that $L_{\text{VAR}}(p_0) = \text{Tr}(\Sigma_\epsilon)$.

$$\begin{aligned} L_{\text{VAR}}(p_0) &= \text{Tr}\left(\Gamma_0\right) - \text{Tr}\left(\begin{bmatrix} \Gamma_1 & \dots & \Gamma_{p_0} \end{bmatrix} \begin{bmatrix} \Gamma_0 & \dots & \Gamma_{p_0-1} \\ \vdots & & \vdots \\ \Gamma_{p_0-1}^T & \dots & \Gamma_0 \end{bmatrix}^{-1} \begin{bmatrix} \Gamma_1^T \\ \vdots \\ \Gamma_{p_0}^T \end{bmatrix}\right) \\ &= \text{Tr}(\Gamma_0) - \text{Tr}\left(\sum_{i=1}^{p_0} A_i \Gamma_{-i}\right) \\ &= \text{Tr}\left(\Gamma_0 - (\Gamma_0 - \Sigma_\epsilon)\right) \\ &= \text{Tr}(\Sigma_\epsilon)\,, \end{aligned}$$

where in the second line we use the Yule-Walker equations to get the A_i's and we use $\Gamma_i^T = \Gamma_{-i}$ and in the third line we use the form of Γ_0 from Lütkepohl (2005, Eq. (2.1.36)).

□

A.3.2 Penalized loss recovers true order

In this section we prove Corollary 1 which states that, for the correct choice of penalty, the true VAR order can be recovered by penalizing the population loss.

Proof of Corollary 1. We first show that for $p > p_0$, $L_{\text{VAR}}(p) + \lambda p > L_{\text{VAR}}(p_0) + \lambda p_0$. This follows immediately by noting that $L_{\text{VAR}}(p) - L_{\text{VAR}}(p_0) = 0$ from Theorem 1 and $\lambda p > \lambda p_0$ for $p > p_0$ as $\lambda > 0$. Next we show that for $p < p_0$, $L_{\text{VAR}}(p) + \lambda p > L_{\text{VAR}}(p_0) + \lambda p_0$. Since $p = p_0 - i$ for some i it suffices to show that $\lambda < \left(L_{\text{VAR}}(p_0 - i) - L_{\text{VAR}}(p_0)\right)/i$ for every $i = 1, \ldots, p_0$. By definition of λ in Corollary 1 this holds. □

A.3.3 Consistency of sample loss

Next, we prove Theorem 2 which establishes that $\hat{L}_{\text{VAR}}(p)$ converges to the population loss $L_{\text{VAR}}(p)$. This is proved using Lemma 1 and the convergence of $\hat{\Gamma}_i$ to Γ_i.

Lemma 1. *Let $\Gamma_{0,L}$ represent the lower triangular portion of Γ_0 including the diagonals. Γ_0 is symmetric so only the lower triangular portion is needed. Letting $C = \text{vec}(\begin{bmatrix}\Gamma_{0,L} & \Gamma_1 & \ldots & \Gamma_p\end{bmatrix}) \in \mathbb{R}^{(k(k+1)/2+k^2p)\times 1}$, we have under Assumption 3 the population squared error loss*

$$f(C) = \text{Tr}\left(\Gamma_0 - \begin{bmatrix}\Gamma_1 & \ldots & \Gamma_p\end{bmatrix}\begin{bmatrix}\Gamma_0 & \ldots & \Gamma_{p-1} \\ \vdots & & \vdots \\ \Gamma_{p-1}^T & \ldots & \Gamma_0\end{bmatrix}^{-1}\begin{bmatrix}\Gamma_1^T \\ \vdots \\ \Gamma_p^T\end{bmatrix}\right),$$

is Lipschitz continuous with respect to C. That is

$$\left|f(\hat{C}) - f(C)\right| \leq L\left\|\hat{C} - C\right\|_1,$$

for some $L < \infty$.

Proof. For ease of notation we define

$$G = \begin{bmatrix} \Gamma_1 & \dots & \Gamma_p \end{bmatrix},$$
$$T = \begin{bmatrix} \Gamma_0 & \dots & \Gamma_{p-1} \\ \vdots & & \vdots \\ \Gamma_{p-1}^T & \dots & \Gamma_0 \end{bmatrix}.$$

From (36) of Petersen et al. (2008) note that $\partial(\mathrm{Tr}(\mathbf{X})) = \mathrm{Tr}(\partial\mathbf{X})$ so that

$$\frac{\partial f(C)}{\partial C_i} = \mathrm{Tr}\left(\frac{\partial \Gamma_0}{\partial C_i} - \frac{\partial G}{\partial C_i}T^{-1}G^T - G\frac{\partial T^{-1}}{\partial C_i}G^T - GT^{-1}\frac{\partial G^T}{\partial C_i}\right).$$

From (59) of Petersen et al. (2008), $\frac{\partial T^{-1}}{\partial C_i} = -T^{-1}\frac{\partial T}{\partial C_i}T^{-1}$. Thus,

$$\frac{\partial f(C)}{\partial C_i} = \mathrm{Tr}\left(\frac{\partial \Gamma_0}{\partial C_i} - \frac{\partial G}{\partial C_i}T^{-1}G^T + GT^{-1}\frac{\partial T}{\partial C_i}T^{-1}G^T - GT^{-1}\frac{\partial G^T}{\partial C_i}\right).$$

Note that each of $\frac{\partial \Gamma_0}{\partial C_i}, \frac{\partial G}{\partial C_i}, \frac{\partial T}{\partial C_i}$ are simply matrices containing 1 in the entries of Γ_0, G, T where C_i is present and 0 otherwise. For example, for $i = 1$,

$$\frac{\partial \Gamma_0}{\partial C_1} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix},$$
$$\frac{\partial G}{\partial C_1} = 0,$$
$$\frac{\partial T}{\partial C_1} = \begin{bmatrix} \frac{\partial \Gamma_0}{\partial C_1} & 0_{k\times k} & \dots & 0_{k\times k} \\ 0_{k\times k} & \frac{\partial \Gamma_0}{\partial C_1} & \dots & 0_{k\times k} \\ \vdots & & \ddots & \vdots \\ 0_{k\times k} & 0_{k\times k} & \dots & \frac{\partial \Gamma_0}{\partial C_1} \end{bmatrix}.$$

Since matrix multiplication, inversion, and the trace are all continuous functions, $\frac{\partial f(C)}{\partial C_i}$ is continuously differentiable for all i. By the mean value theorem and Hölder's inequality

$$\left|f(\hat{C}) - f(C)\right| \leq \left\|\nabla f((1-\nu)C + \nu\hat{C})\right\|_\infty \left\|\hat{C} - C\right\|_1,$$

for some $\nu \in (0,1)$. Since $\frac{\partial f(C)}{\partial C_i}$ is continuous for each C_i, it is bounded on any closed interval including between C and $\hat{C}$. Thus $\left\|\nabla f((1-\nu)C + \nu\hat{C})\right\|_\infty$ is bounded and

$$\left|f(\hat{C}) - f(C)\right| \leq L\left\|\hat{C} - C\right\|_1 ,$$

for some $L < \infty$. □

We are now ready to prove Theorem 2. From Quinn (1980), it is known that for stationary VAR processes, $n^{1/2-\delta}\left(\hat{\Gamma}_{h,ij} - \Gamma_{h,ij}\right)$ converges almost surely to $0\ \forall\delta > 0$ where in our notation with $p_{\max}$ presample values and n sample values, $\hat{\Gamma}_h = \frac{1}{n+p_{\max}}\sum_{t=1}^{n+p_{\max}-h}(\mathbf{z}_t - \overline{\mathbf{z}})(\mathbf{z}_{t+h} - \overline{\mathbf{z}})^T$. However, we do not use $\hat{\Gamma}_h$ to estimate Γ_h. Instead, we use the least-squares estimate, denoted as $\hat{\Gamma}_h^{\text{LS}}$. The difference between the two estimators is that $\hat{\Gamma}_h$ uses all possible observations for every h, which is $n + p_{\max} - h$ observations, while $\hat{\Gamma}_h^{\text{LS}}$ only uses n observations for every h. The exact relationship between the two is defined in Eq. (A.1). Thus, we cannot immediately use the convergence rates from Quinn (1980). As a result, the proof of Theorem 2 first shows that the difference between our least-squares estimates, $\hat{\Gamma}_h^{\text{LS}}$, and $\hat{\Gamma}_h$ as defined in Quinn (1980) is asymptotically negligible. With this, the rate of convergence of $\hat{\Gamma}_h^{\text{LS}}$ to Γ_h is the same as the rate of convergence of $\hat{\Gamma}_h$ to Γ_h. Once this is established, Theorem 2 is proved by applying Lemma 1.

Proof of Theorem 2. For our least squares estimator of the loss,

$$\begin{aligned}\hat{L}_{\text{VAR}}(p) &= \text{Tr}\left(\left(\mathbf{Y} - \hat{A}_p\mathbf{X}_p\right)\left(\mathbf{Y} - \hat{A}_p\mathbf{X}_p\right)^T\right) \\ &= \text{Tr}\left(\mathbf{Y}\,\mathbf{Y}^T - \mathbf{Y}\,\mathbf{X}_p^T\left(\mathbf{X}_p\mathbf{X}_p^T\right)^{-1}\mathbf{X}_p\,\mathbf{Y}^T\right)\end{aligned} .$$

where for de-meaned data,

$$\mathbf{Y}\,\mathbf{Y}^T = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{z}_i-\overline{\mathbf{z}})(\mathbf{z}_i-\overline{\mathbf{z}})^T$$

$$\mathbf{Y}\,\mathbf{X}_p^T = \sum_{i=1}^{n}(\mathbf{z}_i-\overline{\mathbf{z}})\left(\mathbf{x}_{i,p}-\overline{\mathbf{x}}_{i,p}\right)^T = \frac{1}{n}\left[\sum_{i=1}^{n}(\mathbf{z}_i-\overline{\mathbf{z}})(\mathbf{z}_{i-1}-\overline{\mathbf{z}})^T \quad \ldots \quad \sum_{i=1}^{n}(\mathbf{z}_i-\overline{\mathbf{z}})\left(\mathbf{z}_{i-p}^T-\overline{\mathbf{z}}\right)\right]$$

$$\mathbf{X}_p\mathbf{X}_p^T = \frac{1}{n}\sum_{i=1}^{n}\begin{bmatrix}(\mathbf{z}_{i-1}-\overline{\mathbf{z}})\\ \vdots \\ (\mathbf{z}_{i-p}-\overline{\mathbf{z}})\end{bmatrix}\left[(\mathbf{z}_{i-1}-\overline{\mathbf{z}})^T \quad \ldots \quad (\mathbf{z}_{i-p}-\overline{\mathbf{z}})^T\right]$$

$$= \frac{1}{n}\sum_{i=1}^{n}\begin{bmatrix}(\mathbf{z}_{i-1}-\overline{\mathbf{z}})(\mathbf{z}_{i-1}-\overline{\mathbf{z}})^T & (\mathbf{z}_{i-1}-\overline{\mathbf{z}})(\mathbf{z}_{i-2}-\overline{\mathbf{z}})^T & \ldots & (\mathbf{z}_{i-1}-\overline{\mathbf{z}})(\mathbf{z}_{i-p}-\overline{\mathbf{z}})^T \\ & \vdots & & \vdots \\ (\mathbf{z}_{i-p}-\overline{\mathbf{z}})(\mathbf{z}_{i-1}-\overline{\mathbf{z}})^T & (\mathbf{z}_{i-p}-\overline{\mathbf{z}})(\mathbf{z}_{i-2}-\overline{\mathbf{z}})^T & \ldots & (\mathbf{z}_{i-p}-\overline{\mathbf{z}})(\mathbf{z}_{i-p}-\overline{\mathbf{z}})^T\end{bmatrix}.$$

Note the use of negative indices. These refer to presample values of which we have $1,\ldots,p_{\max}$. Specifically for the presample notation we can index our $n+p_{\max} := N$ values as $i \in \{-p_{\max}+1,\ldots,0,1,\ldots,n\}$ which correspond respectively to the standard indices $t \in \{1,\ldots,N\}$.

Then for $j \geq k$ we denote $\hat{\Gamma}_{j-k}^{\mathrm{LS}}$ as our LS estimate of Γ_{j-k} and we can write

$$\begin{aligned}\hat{\Gamma}_{j-k}^{\mathrm{LS}} =& \frac{1}{n}\sum_{i=1}^{n}(\mathbf{z}_{i-j}-\overline{\mathbf{z}})(\mathbf{z}_{i-k}-\overline{\mathbf{z}})^T \\ =& \frac{1}{n}\sum_{l=1+p_{\max}}^{n+p_{\max}}(\mathbf{z}_{l-j}-\overline{\mathbf{z}})(\mathbf{z}_{l-k}-\overline{\mathbf{z}})^T \\ =& \frac{1}{n}\sum_{t=p_{\max}-j+1}^{n+p_{\max}-j}(\mathbf{z}_t-\overline{\mathbf{z}})\left(\mathbf{z}_{t-(j-k)}-\overline{\mathbf{z}}\right)^T,\end{aligned}$$

where in the second line we simply change the index from the presample notation to standard indices by using $i=1$ corresponds to observation $t=1+p_{\max}$ in the t notation and in the third line we simply change the index to $t=l-j$. This allows us to easily compare to $\hat{\Gamma}_h$ defined in Quinn

(1980). Specifically we write

$$\begin{aligned}\hat{\Gamma}_{j-k} &= \frac{1}{n+p_{\max}} \sum_{t=1}^{n+p_{\max}-(j-k)} (\mathbf{z}_t - \overline{\mathbf{z}}) \left(\mathbf{z}_{t+(j-k)} - \overline{\mathbf{z}}\right)^T \\ &= \frac{1}{n+p_{\max}} \sum_{t=1}^{p_{\max}-j} (\mathbf{z}_t - \overline{\mathbf{z}}) \left(\mathbf{z}_{t+(j-k)} - \overline{\mathbf{z}}\right)^T + \frac{1}{n+p_{\max}} \sum_{t=p_{\max}-j+1}^{n+p_{\max}-j} (\mathbf{z}_t - \overline{\mathbf{z}}) \left(\mathbf{z}_{t+(j-k)} - \overline{\mathbf{z}}\right)^T \\ &\quad + \frac{1}{n+p_{\max}} \sum_{t=n+p_{\max}-j+1}^{n+p_{\max}-(j-k)} (\mathbf{z}_t - \overline{\mathbf{z}}) \left(\mathbf{z}_{t+(j-k)} - \overline{\mathbf{z}}\right)^T . \end{aligned} \tag{A.1}$$

The middle term in the last equality is simply $\frac{n}{n+p_{\max}}\hat{\Gamma}^{\mathrm{LS}}_{j-k}$. The first and last term are finite and do not grow with n so they are $o(1/n)$. Therefore we can rescale $\hat{\Gamma}_{j-k}$ by $\frac{n+p_{\max}}{n}$ to get

$$\hat{\Gamma}^{\mathrm{LS}}_{j-k} = \frac{n+p_{\max}}{n}\hat{\Gamma}_{j-k} - o\left(\frac{1}{n}\right) ,$$

which for $h = j-k$ gives that $n^{1/2-\delta}\left(\hat{\Gamma}^{LS}_{h,ij} - \Gamma_{h,ij}\right)$ converges almost surely to $0\ \forall\delta > 0$. The same rate holds for $\hat{\Gamma}^{LS}_{-h,ij}$ as $\hat{\Gamma}^{LS}_{-h,ij} = \hat{\Gamma}^{LS,T}_{h,ij}$. Noting that $f(\hat{C}^{\mathrm{LS}}) = \hat{L}_{\mathrm{VAR}}(p)$ when $\hat{C}^{\mathrm{LS}}$ is composed of the least-squares estimates of Γ_h and that $f(C) = L_{\mathrm{VAR}}(p)$, we have by Lemma 1,

$$\begin{aligned} |\hat{L}_{\mathrm{VAR}}(p) - L_{\mathrm{VAR}}(p)| &\leq L \left\|\hat{C}^{\mathrm{LS}} - C\right\|_1 \\ |\hat{L}_{\mathrm{VAR}}(p) - L_{\mathrm{VAR}}(p)| &\leq L(k(k+1)/2 + k^2 p) o_p(n^{-1/2+\delta}) \\ |\hat{L}_{\mathrm{VAR}}(p) - L_{\mathrm{VAR}}(p)| &= o_p(n^{-1/2+\delta}) \end{aligned}$$

□

A.4 Additional simulation results

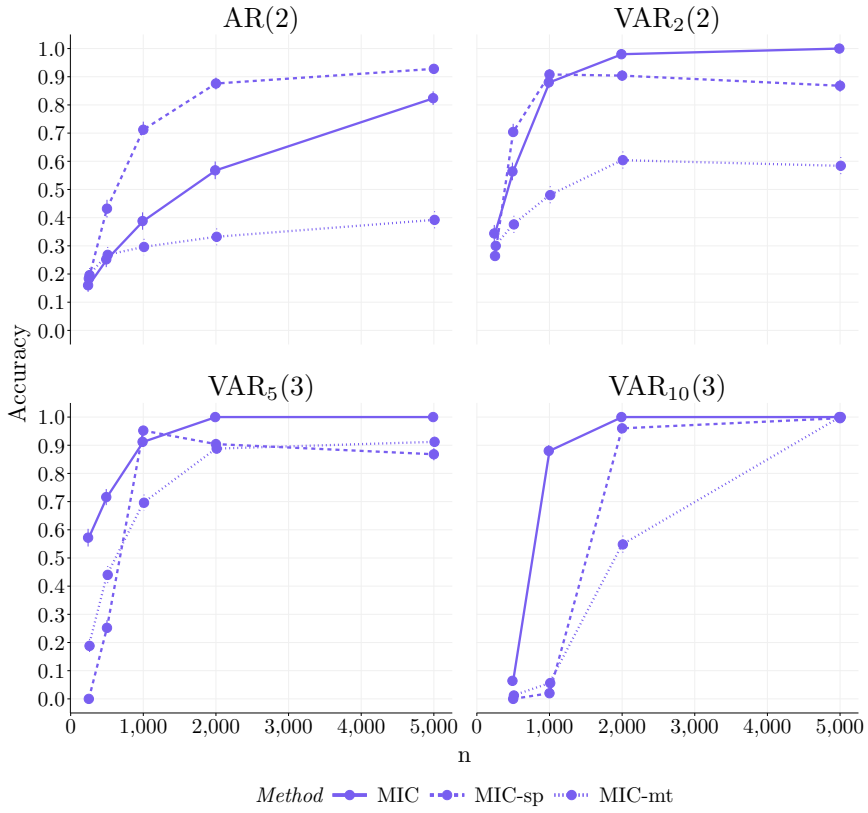


Figure A.1: **Diagonal Gaussian errors**. Simulation results comparing accuracy of different MIC order selection methods. Vertical lines indicate standard errors.

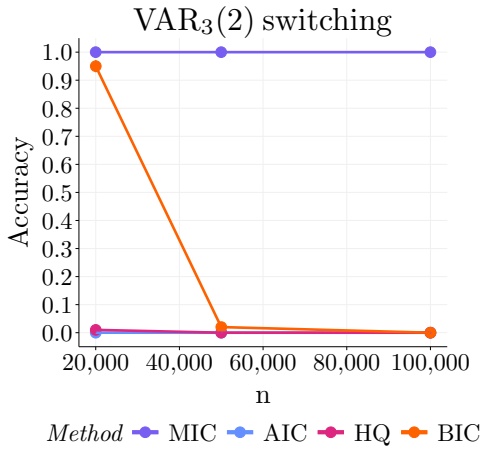


Figure A.2: **$\mathbf{VAR_3(2)}$ switching**. Large sample size simulation results. Vertical lines indicate standard errors.

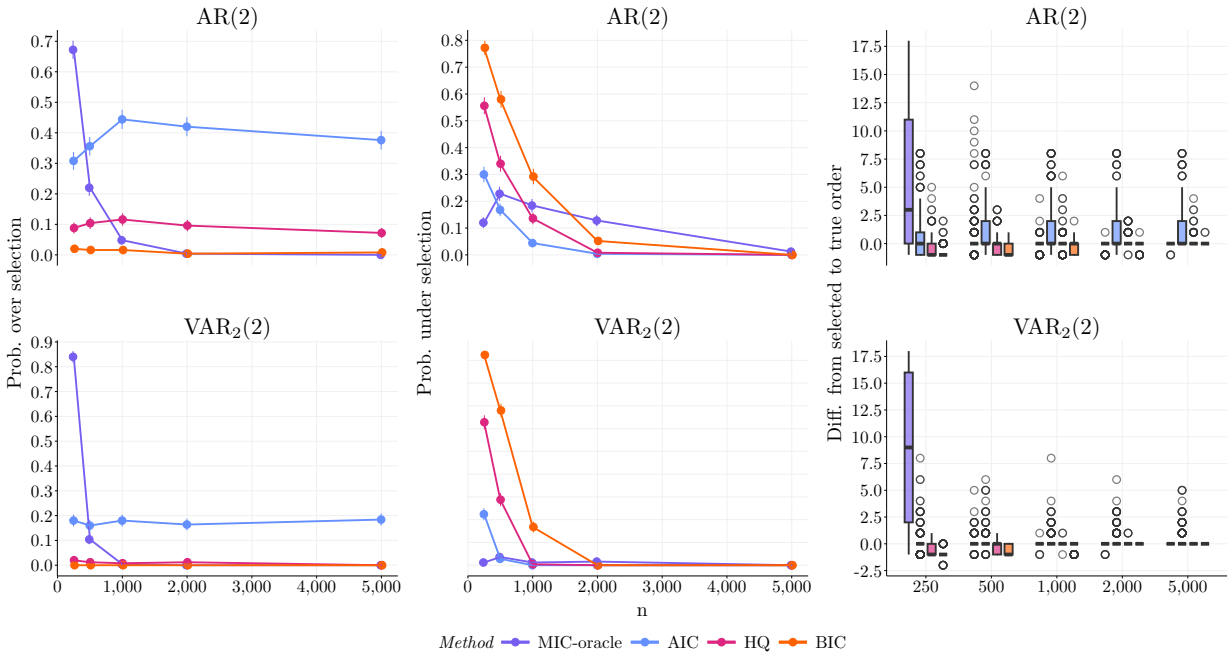


Figure A.3: **Diagonal Gaussian errors**. Simulation results for over and under selection of order with diagonal Gaussian errors. Vertical lines indicate standard errors.

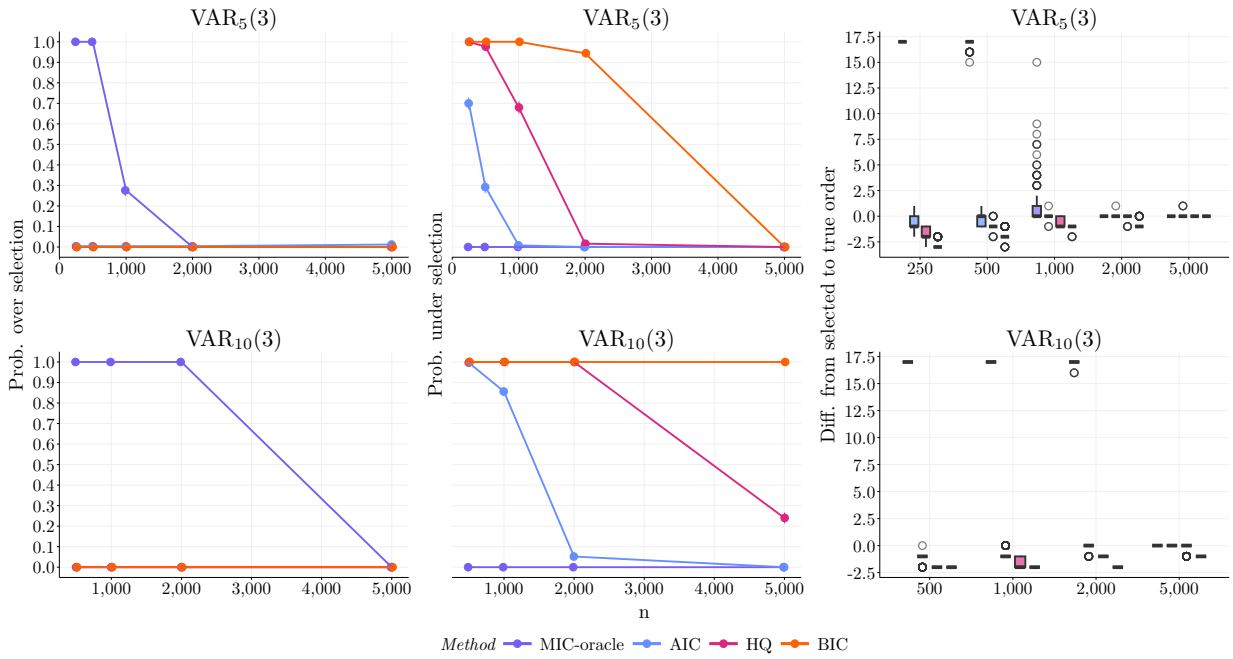


Figure A.4: **Diagonal Gaussian errors**. Simulation results for over and under selection of order with diagonal Gaussian errors. Vertical lines indicate standard errors.

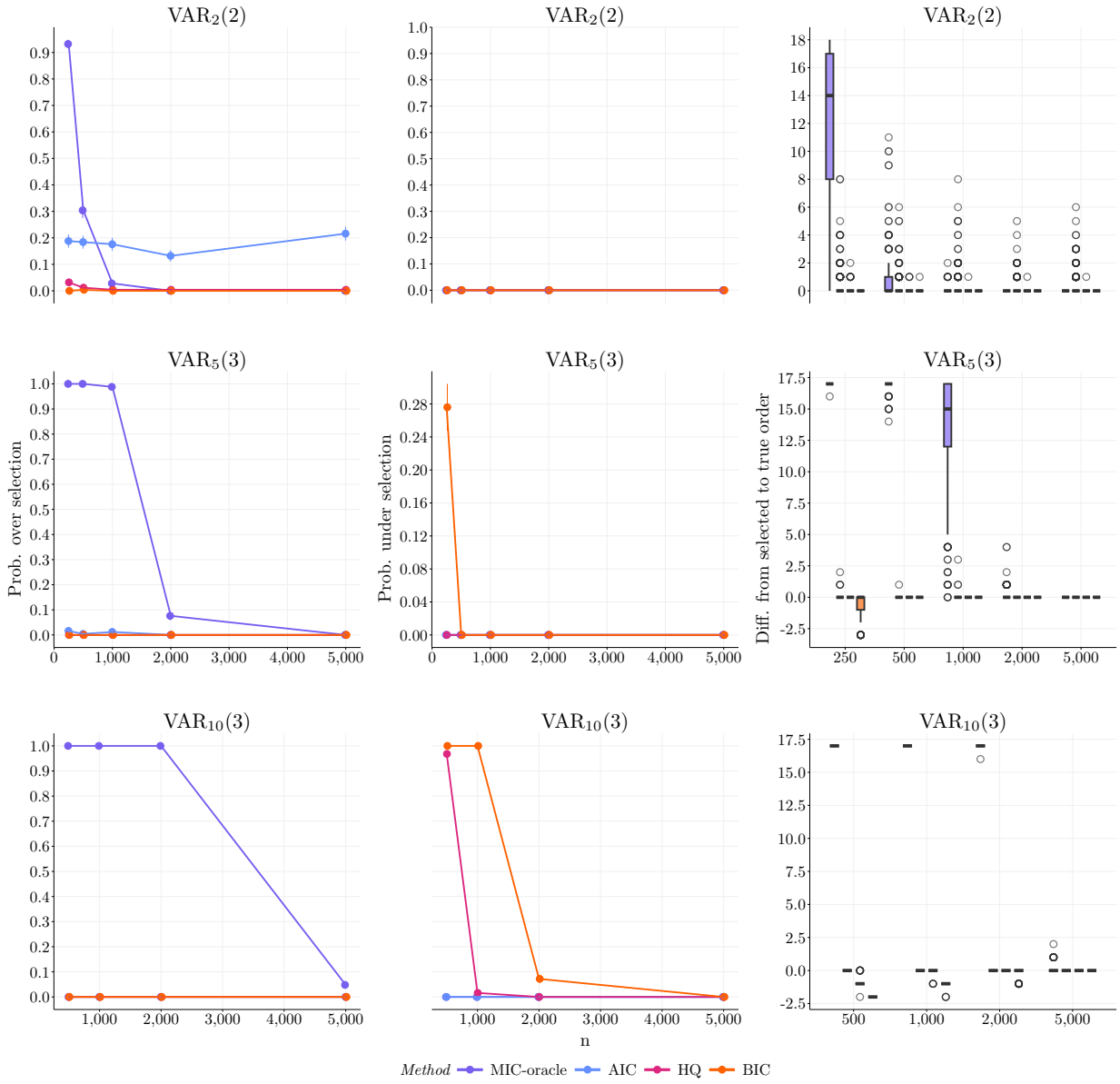


Figure A.5: **Non-diagonal Gaussian errors**. Simulation results for over and under selection of order with non-diagonal Gaussian errors. Vertical lines indicate standard errors.

A.5 Daily realized stock variances

Stock	Description	$k = 16$	$k = 7$
AEX	Amsterdam Exchange Index	✓	
AORD	All Ordinaries Index	✓	
BFX	Belgium Bell 20 Index	✓	
BVSP	BOVESPA Index	✓	
DJI	Dow Jones Industrial Average	✓	
FCHI	Cotation Assist´ee en Continu Index	✓	✓
FTSE	Financial Times Stock Exchange Index 100	✓	✓
GDAXI	Deutscher Aktienindex	✓	✓
HSI	HANG SENG Index	✓	✓
IXIC	Nasdaq stock index	✓	
KS11	Korea Composite Stock Price Index	✓	✓
MXX	IPC Mexico	✓	
N225	Tokyo stock exchange index		✓
RUT	Russel 2000	✓	
SPX	Standard & Poor's 500 market index	✓	✓
SSMI	Swiss market index	✓	
STOXX50E	EURO STOXX 50	✓	

Table A.1: Stocks analyzed in financial application

Appendix B

APPENDICES FOR CHAPTER 3

B.1 Concentration inequality on spectral density estimators

In this section, we prove Theorem 3 which establishes a concentration inequality on a general class of spectral density estimators.

We prove Theorem 3 by applying Theorem 1 and Proposition 2 of Zhang and Zhang (2025) to our setting and combining the conclusions. Although Theorem 1 and Proposition 2 as originally stated apply only to λ over a set of discrete Fourier frequencies, per Remark 1 of Zhang and Zhang (2025), they are easily extended to apply to any subset of the real line. These are the versions of their results we use in our proof below.

Proof of Theorem 3. By Zhang and Zhang (2025, Theorem 1),

$$\mathbb{P}\left\{\max_{\lambda\in[0,2\pi]}\left\|\hat{f}_l(\lambda)-\mathbb{E}\hat{f}_l(\lambda)\right\|_\infty \geq x\right\} \lesssim B_l p^2 \exp\left\{-C_{\alpha_l}\left(\frac{\sqrt{T_l}x}{\sqrt{B_l}\|X_{l,\cdot}\|^2_{\psi_{\alpha_l}}}\right)^{\frac{1}{1+2\alpha_l}}\right\}, \quad l=1,2.$$

Set $x = \|X_{l,\cdot}\|^2_{\psi_{\alpha_l}}(B_l/T_l)^{1/2}(C/C_{\alpha_l})^{1+2\alpha_l}\log^{1+2\alpha_l}(B_l p)$ where C_{α_l} is the constant defined in Zhang and Zhang (2025, Theorem 1) and C is a different constant. Plugging in and simplifying,

$$\mathbb{P}\left\{\max_{\lambda\in[0,2\pi]}\left\|\hat{f}_l(\lambda)-\mathbb{E}\hat{f}_l(\lambda)\right\|_\infty \geq \|X_{l,\cdot}\|^2_{\psi_{\alpha_l}}\left(\frac{B_l}{T_l}\right)^{1/2}\left(\frac{C}{C_{\alpha_l}}\right)^{1+2\alpha_l}\log^{1+2\alpha_l}(B_l p)\right\} \lesssim \frac{B_l p^2}{(B_l p)^C}, \quad l=1,2.$$

Thus for $C \geq 2$ we have that,

$$\max_{\lambda\in[0,2\pi]}\left\|\hat{f}_l(\lambda)-\mathbb{E}\hat{f}_l(\lambda)\right\|_\infty = O_p\left(\|X_{l,\cdot}\|^2_{\psi_{\alpha_l}}\sqrt{\frac{B_l}{T_l}}\log^{1+2\alpha_l}(B_l p)\right), \quad l=1,2. \tag{B.1}$$

By Zhang and Zhang (2025, Proposition 2), the bias of $\hat{f}_l(\lambda)$ is

$$\max_{\lambda\in[0,2\pi]}\left\|\mathbb{E}\hat{f}_l(\lambda)-f_l(\lambda)\right\|_\infty \lesssim \frac{\Phi^2_{l,2}}{B_l}, \quad l=1,2. \tag{B.2}$$

Now, by the triangle inequality,

$$\max_{\lambda\in[0,2\pi]}\left\|\hat{f}_l(\lambda)-f_l(\lambda)\right\|_\infty \le \max_{\lambda\in[0,2\pi]}\left\|\hat{f}_l(\lambda)-\mathbb{E}\hat{f}_l(\lambda)\right\|_\infty + \max_{\lambda\in[0,2\pi]}\left\|\mathbb{E}\hat{f}_l(\lambda)-f_l(\lambda)\right\|_\infty,$$

and combining this with (B.1) and (B.2),

$$\max_{\lambda\in[0,2\pi]}\left\|\hat{f}_l(\lambda)-f_l(\lambda)\right\|_\infty = O_p\left(\|X_{l,\cdot}\|^2_{\psi_{\alpha_l}}\sqrt{\frac{B_l}{T_l}}\log^{1+2\alpha_l}(B_lp)+\frac{\Phi^2_{l,2}}{B_l}\right).$$

Let

$$R_{T_l,p} = \|X_{l,\cdot}\|^2_{\psi_{\alpha_l}}\sqrt{\frac{B_l}{T_l}}\log^{1+2\alpha_l}(B_lp)+\frac{\Phi^2_{l,2}}{B_l},\quad l=1,2. \tag{B.3}$$

For $B_l \asymp \{T_l/\log^{2+4\alpha_l}(p\vee T_l)\}^{\gamma_l}$,

$$\begin{aligned}R_{T_l,p} &\asymp \|X_{l,\cdot}\|^2_{\psi_{\alpha_l}}\frac{\left(\frac{T_l}{\log^{2+4\alpha_l}(p\vee T_l)}\right)^{\gamma_l/2}}{T_l^{1/2}}\log^{1+2\alpha_l}\left\{\left(\frac{T_l}{\log^{2+4\alpha_l}(p\vee T_l)}\right)^{\gamma_l}p\right\}+\Phi^2_{l,2}\left(\frac{T_l}{\log^{2+4\alpha_l}(p\vee T_l)}\right)^{-\gamma_l}\\ &\lesssim \|X_{l,\cdot}\|^2_{\psi_{\alpha_l}}\frac{T_l^{-1/2+\gamma_l/2}}{\log^{\gamma_l(1+2\alpha_l)}(p\vee T_l)}\log^{1+2\alpha_l}(T_l^{\gamma_l}p)+\Phi^2_{l,2}\frac{T_l^{-\gamma_l}}{\log^{-\gamma_l(2+4\alpha_l)}(p\vee T_l)},\end{aligned}$$

where we have dropped the term $\log^{2+4\alpha_l}(p\vee T_l)$ in $\log^{1+2\alpha_l}$ since for $p\vee T_l > 3$, this term is >1. We do not make this $p\vee T_l$ assumption explicit as we are considering asymptotics and high dimensions so it will always be true that $p\vee T_l>3$. This condition is implicit in many other parts of the proof and theorem.

When $\gamma_l \ge 1/3$, we have $-(1-\gamma_l)/2 \ge -\gamma_l$, and the first term dominates. Otherwise, the second term dominates. Thus,

$$R_{T_l,p} \lesssim \begin{cases}\|X_{l,\cdot}\|^2_{\psi_{\alpha_l}}\frac{T_l^{-1/2+\gamma_l/2}}{\log^{\gamma_l(1+2\alpha_l)}(p\vee T_l)}\log^{1+2\alpha_l}(T_l^{\gamma_l}p) & \text{if } \gamma_l\ge 1/3,\\ \Phi^2_{l,2}\frac{T_l^{-\gamma_l}}{\log^{-\gamma_l(2+4\alpha_l)}(p\vee T_l)} & \text{if } \gamma_l<1/3.\end{cases}$$

To simplify further we also used that $\log^{2\alpha_l+1}(T_l^{\gamma_l}p) = (\log(T_l^{\gamma_l}p))^{2\alpha_l+1} = (\gamma_l\log(T_l)+\log(p))^{2\alpha_l+1} \le 2^{\alpha_l+1}\log(p\vee T_l)$. □

It immediately follows that Theorem 3 also applies to $\|\hat{\Sigma}_l-\Sigma_l\|_\infty$. This is stated in the following Corollary.

Corollary 3. *Suppose the conditions of Theorem 3 are satisfied. Then for* $B_l \asymp \left(\frac{T_l}{\log^{4\alpha_l+2}(p\vee T_l)}\right)^{\gamma_l}$,

$$\max_{\lambda\in[0,2\pi]} \left\|\hat{\Sigma}_l(\lambda) - \Sigma_l(\lambda)\right\|_\infty = \begin{cases} O_p\left(2^{\alpha_l+1}\left\|X_{l,\cdot}\right\|^2_{\psi_{\alpha_l}} T_l^{\frac{\gamma_l-1}{2}} \log^{(1-\gamma_l)(2\alpha_l+1)}(p\vee T_l)\right) & \text{for } \gamma_l \geq 1/3 \\ O_p\left(\Phi^2_{l,2} T_l^{-\gamma_l}\log^{\gamma_l(4\alpha_l+2)}(p\vee T_l)\right) & \text{for } \gamma_l < 1/3 \end{cases}$$

Proof. Recall $f_l(\lambda) = A_l(\lambda) + iB_l(\lambda)$ and $\Sigma_l(\lambda) = \begin{bmatrix} A_l(\lambda) & -B_l(\lambda) \\ B_l(\lambda) & A_l(\lambda)\end{bmatrix}$. The subscript l represents the l^{th} condition (1 or 2 in this case) and λ represents the frequency of interest. Note that $\|\Sigma_l(\lambda)\|_\infty = \max(\|A_l(\lambda)\|_\infty, \|B_l(\lambda)\|_\infty) \leq \|f_l(\lambda)\|_\infty$. Similarly $\|\hat{\Sigma}_l(\lambda) - \Sigma_l(\lambda)\|_\infty \leq \|\hat{f}_l(\lambda) - f_l(\lambda)\|_\infty$. Thus, Theorem 3 can be applied to $\|\hat{\Sigma}_l(\lambda) - \Sigma_l(\lambda)\|_\infty$. □

B.2 Convergence rate of SDD estimator

We will use Corollary 1 of Negahban et al. (2012) to prove Theorem 4. Corollary 1 of Negahban et al. (2012) establishes a bound on the error of high-dimensional M-estimators such as SDD. In order to use Corollary 1 we need to establish restricted strong convexity (RSC) of the loss function and decomposability of the penalty. Showing decomposability of the penalty is relatively quick and thus is shown in the proof of Theorem 4. On the other hand, showing the RSC of the loss is more involved and requires an intermediate result on the convergence of the second derivative of the sample D-trace loss. This may be of independent interest and is stated in Lemma 2.

Lemma 2 (Concentration inequality on second derivative of D-trace loss)**.** *Suppose there exists constants* α_1, α_2 *such that Assumption 1 is satisfied for conditions 1 and 2, respectively. Furthermore, suppose the smoothing span for condition* l *is* $B_l \asymp \left(\frac{T_l}{\log^{4\alpha_l+2}(p\vee T_l)}\right)^{\gamma_l}$. *Then,*

$$\left\|0.5\left(\hat{\Sigma}_1\otimes\hat{\Sigma}_2 + \hat{\Sigma}_2\otimes\hat{\Sigma}_1\right) - 0.5\left(\Sigma_1\otimes\Sigma_2 + \Sigma_2\otimes\Sigma_1\right)\right\|_\infty \leq C_f O_p\left(R_{T_1,p} + R_{T_2,p}\right),$$

where $C_f = \max(\|f_1\|_\infty, \|f_2\|_\infty)$ *and* $R_{T_1,p}$, $R_{T_2,p}$ *are the rates from applying Corollary 3 to conditions 1 and 2 respectively.*

Proof. Note that both

$$\|\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 - \Sigma_2 \otimes \Sigma_1\|_\infty, \|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\|_\infty \leq \|\Sigma_1\|_\infty \|\hat{\Sigma}_2 - \Sigma_2\|_\infty + \|\Sigma_2\|_\infty \|\hat{\Sigma}_1 - \Sigma_1\|_\infty + \|\hat{\Sigma}_1 - \Sigma_1\|_\infty \|\hat{\Sigma}_2 - \Sigma_2\|_\infty .$$

Denote $C_f = \max(\|f_1\|_\infty, \|f_2\|_\infty)$ and the convergence rates from applying Corollary 3 to conditions 1 and 2 as $R_{T_1,p}$ and $R_{T_2,p}$. Recall that both of these rates also depend on γ_1, γ_2 and α_1, α_2 which are the smoothing exponents used in conditions 1,2 and the dependence in conditions 1,2 respectively. However, we have excluded these dependencies in the notation of $R_{T_1,p}$ and $R_{T_2,p}$ for convenience. Combining these we get that

$$\begin{aligned}\|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\|_\infty &= C_f \left(O_p\left(R_{T_1,p}\right) + O_p\left(R_{T_2,p}\right)\right) + O_p\left(R_{T_1,p} R_{T_2,p}\right) \\ &= C_f O_p\left(R_{T_1,p} + R_{T_2,p}\right) .\end{aligned}$$

In the last line we dropped the term $O_p\left(R_{T_1,p} R_{T_2,p}\right)$ as it is second order.

The same rate also holds for $\|\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 - \Sigma_2 \otimes \Sigma_1\|_\infty$. From the triangle inequality we also have that

$$\begin{aligned}&\left\|0.5\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right) - 0.5\left(\Sigma_1 \otimes \Sigma_2 + \Sigma_2 \otimes \Sigma_1\right)\right\|_\infty \\ &\qquad \leq 0.5\left\|\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 - \Sigma_2 \otimes \Sigma_1\right\|_\infty + 0.5\left\|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\right\|_\infty\end{aligned}$$

and we conclude

$$\left\|0.5\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right) - 0.5\left(\Sigma_1 \otimes \Sigma_2 + \Sigma_2 \otimes \Sigma_1\right)\right\|_\infty = C_f O_p\left(R_{T_1,p} + R_{T_2,p}\right) . \quad \text{(B.4)}$$

□

In the proof of RSC in Lemma 4, we use the simplification that $\lambda_{\min}(\Sigma_l) = \lambda_{\min}(f_l)$. This is established in Lemma 3.

Lemma 3 (Eigenvalues of expanded spectral density)**.** *Suppose $f \in \mathbb{C}^{p\times p}$ is defined as $f := A + iB$. Further suppose f is Hermitian. Consider the expansion of f to the real space defined as $\Sigma = \begin{bmatrix} A & -B \\ B & A \end{bmatrix}$. Let $\lambda_1, \ldots, \lambda_p$ be the eigenvalues of f. Then the eigenvalues of Σ are $\lambda_1, \lambda_1, \ldots, \lambda_p, \lambda_p$. That is, the eigenvalues of Σ are the same as those in f where each is repeated twice.*

Proof. Since f is Hermitian it's eigenvalues are real. Let λ_i denote an eigenvalue of f and $\boldsymbol{\nu}_i$ it's corresponding eigenvector. Note in general $\boldsymbol{\nu}_i$ is complex valued so we will write it as $\boldsymbol{\nu}_i = \boldsymbol{\nu}_{R,i} + i\boldsymbol{\nu}_{I,i}$. By definition of eigenvalues we have

$$\begin{aligned} f\boldsymbol{\nu}_i &= \lambda_i \boldsymbol{\nu}_i \\ (A+iB)(\boldsymbol{\nu}_{R,i} + i\boldsymbol{\nu}_{I,i}) &= \lambda_i(\boldsymbol{\nu}_{R,i} + i\boldsymbol{\nu}_{I,i}) \\ (A\boldsymbol{\nu}_{R,i} - B\boldsymbol{\nu}_{I,i}) + (B\boldsymbol{\nu}_{R,i} + A\boldsymbol{\nu}_{I,i})i &= \lambda_i\boldsymbol{\nu}_{R,i} + i\lambda_i\boldsymbol{\nu}_{I,i}\,, \end{aligned}$$

which implies that

$$\begin{aligned} (A\boldsymbol{\nu}_{R,i} - B\boldsymbol{\nu}_{I,i}) &= \lambda_i\boldsymbol{\nu}_{R,i} \\ (B\boldsymbol{\nu}_{R,i} + A\boldsymbol{\nu}_{I,i}) &= \lambda_i\boldsymbol{\nu}_{I,i}\,. \end{aligned} \tag{B.5}$$

Consider a scalar δ_j and vector $\boldsymbol{\eta}_j := \begin{bmatrix} \boldsymbol{\eta}_{j,1} & \boldsymbol{\eta}_{j,2} \end{bmatrix}$ where $\boldsymbol{\eta}_{j,1}, \boldsymbol{\eta}_{j,2}$ are p dimensional. Then $(\delta_j, \boldsymbol{\eta}_j)$ will be an eigenvalue, eigenvector pair of Σ if

$$\begin{bmatrix} A & -B \\ B & A \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_{j,1} \\ \boldsymbol{\eta}_{j,2} \end{bmatrix} = \delta_j \begin{bmatrix} \boldsymbol{\eta}_{j,1} \\ \boldsymbol{\eta}_{j,2} \end{bmatrix}\,,$$

which can be written as

$$\begin{bmatrix} A\boldsymbol{\eta}_{j,1} - B\boldsymbol{\eta}_{j,2} \\ B\boldsymbol{\eta}_{j,1} + A\boldsymbol{\eta}_{j,2} \end{bmatrix} = \delta_j \begin{bmatrix} \boldsymbol{\eta}_{j,1} \\ \boldsymbol{\eta}_{j,2} \end{bmatrix}\,.$$

Using equation B.5, one such pair that satisfies this equation is $(\delta_j, \boldsymbol{\eta}_j) = (\lambda_1, \begin{bmatrix} \boldsymbol{\nu}_{R,1} & \boldsymbol{\nu}_{I,1} \end{bmatrix})$ while another is $(\lambda_1, \begin{bmatrix} -\boldsymbol{\nu}_{I,1} & \boldsymbol{\nu}_{R,1} \end{bmatrix})$. This holds for all p eigenvalue, eigenvector pairs in f. Since Σ has $2p$ eigenvalues, eigenvector pairs this concludes the proof.

□

With Lemma 2 and Corollary 3 we are ready to establish RSC of the D-trace loss function. This is stated in Lemma 4. Throughout we will use the notation S_{Δ^*} to denote the set of non-zero entries of $\mathrm{vec}(\Delta^*)$ and the notation $\mathbf{x}_{S_{\Delta^*}}$ and $\mathbf{x}_{S^c_{\Delta^*}}$ to denote the elements in $\mathbf{x}$ corresponding to the non-zero and zero indices of $\mathrm{vec}(\Delta^*)$ respectively. We use this as it is more notationally convenient than writing e.g. $S_{\mathrm{vec}(\Delta^*)}$.

Lemma 4 (RSC of D-trace loss)**.** *Suppose there exists constants α_1, α_2 such that Assumption 1 is satisfied for conditions 1 and 2, respectively. Furthermore, suppose the smoothing span for condition l is $B_l \asymp \left(\frac{T_l}{\log^{4\alpha_l+2}(p \vee T_l)}\right)^{\gamma_l}$. Define $\overline{\mathcal{M}} = \left\{\theta : \mathbb{R}^{4p^2} \middle| \theta_{ij} \neq 0 \text{ for all } (i,j) \in S_{\Delta^*}\right\}$ and $\overline{\mathcal{M}}^{\perp} = \left\{\gamma : \mathbb{R}^{4p^2} \middle| \gamma_{ij} = 0 \text{ for all } (i,j) \in S_{\Delta^*}\right\}$. Then for T_1, T_2 large enough, RSC holds for the sample D-trace loss with $\kappa_L = \lambda_{\min}(f_1)\lambda_{\min}(f_2)/2$ and a tolerance of 0 with high probability for all $\mathbf{m} \in \mathcal{C}(\overline{\mathcal{M}}, \overline{\mathcal{M}}^{\perp}; \Delta^*)$ where $\mathcal{C}(\overline{\mathcal{M}}, \overline{\mathcal{M}}^{\perp}; \Delta^*) := \left\{\mathbf{m} \in \mathbb{R}^{4p^2} \middle| \|\mathbf{m}_{S^c_{\Delta^*}}\|_1 \leq 3\|\mathbf{m}_{S^*_{\Delta}}\|_1\right\}$.*

Proof. To show the RSC condition holds with high probability and with tolerance of 0 ($\tau^2_{\mathcal{L}}(\theta^*) = 0$ in the notation of Negahban et al. (2012)), we must show that $\mathbf{m}^T \left(\nabla^2 L(\Delta, \hat{\Sigma}_1, \hat{\Sigma}_2)\right) \mathbf{m} > \kappa_L \|\mathbf{m}\|_2^2$ for all $\mathbf{m} \in \mathcal{C}(\overline{\mathcal{M}}, \overline{\mathcal{M}}^{\perp}; \Delta^*)$ for some κ_L. Note that the definition of $\mathcal{C}(\overline{\mathcal{M}}, \overline{\mathcal{M}}^{\perp}; \Delta^*)$ follows from Negahban et al. (2012) (17) and by noting that $\left\|\mathrm{vec}(\Delta^*)_{S^c_{\Delta^*}}\right\|_1 = 0$.

First note that for $\mathbf{m} \in \mathcal{C}(\overline{\mathcal{M}}, \overline{\mathcal{M}}^{\perp}; \Delta^*)$, $\|\mathbf{m}\|_1 \leq 4\|\mathbf{m}_{S_{\Delta^*}}\|_1 \leq 4\sqrt{s_{\Delta^*}}\|\mathbf{m}_{S_{\Delta^*}}\|_2$ where $s_{\Delta^*} = |S_{\Delta^*}|$, i.e. s_{Δ^*} is the number of non-zero entries of $\mathrm{vec}(\Delta^*)$. Recall $\nabla^2 L(\Delta, \hat{\Sigma}_1, \hat{\Sigma}_2) = 0.5(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1)$. Showing RSC proceeds the similarly to Wang et al. (2021)

$$\begin{aligned}\mathbf{m}^T(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2)\mathbf{m} &\geq \mathbf{m}^T(\Sigma_1 \otimes \Sigma_2)\mathbf{m} + \mathbf{m}^T\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\right)\mathbf{m} \\ &\geq \lambda_{\min}(\Sigma_1)\lambda_{\min}(\Sigma_2)\|\mathbf{m}\|_2^2 - \left|\mathbf{m}^T\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\right)\mathbf{m}\right| \\ &\geq \lambda_{\min}(\Sigma_1)\lambda_{\min}(\Sigma_2)\|\mathbf{m}\|_2^2 - \|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\|_\infty\|\mathbf{m}\|_1^2 \\ &\geq \lambda_{\min}(\Sigma_1)\lambda_{\min}(\Sigma_2)\|\mathbf{m}\|_2^2 - 16s_{\Delta^*}\|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\|_\infty\|\mathbf{m}\|_2^2 .\end{aligned} \tag{B.6}$$

The last inequality follows from the fact that $\|\mathbf{m}\|_1 = \|\mathbf{m}_{S_{\Delta^*}^c}\|_1 + \|\mathbf{m}_{S_{\Delta^*}}\|_1 \leq 4\|\mathbf{m}_{S_{\Delta^*}}\|_1 \leq 4\sqrt{s_{\Delta^*}}\|\mathbf{m}_{S_{\Delta^*}}\|_2 \leq 4\sqrt{s_{\Delta^*}}\|\mathbf{m}\|_2$. Similarly

$$\mathbf{m}^T(\hat{\Sigma}_2 \otimes \hat{\Sigma}_1)\mathbf{m} \geq \lambda_{\min}(\Sigma_1)\lambda_{\min}(\Sigma_2)\|\mathbf{m}\|_2^2 - 16s_{\Delta^*}\|\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 - \Sigma_2 \otimes \Sigma_1\|_\infty\|\mathbf{m}\|_2^2$$

Recall in the proof Lemma 2, we showed that

$$\|\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 - \Sigma_2 \otimes \Sigma_1\|_\infty, \|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\|_\infty = C_f O_p\left(R_{T_1,p} + R_{T_2,p}\right) ,$$

where $R_{T_1,p}, R_{T_2,p}$ are the rates from applying Corollary 3 to conditions 1 and 2. Using this we have that

$$\frac{1}{2}\mathbf{m}^T(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1)\mathbf{m} \geq \lambda_{\min}(f_1)\lambda_{\min}(f_2)\|\mathbf{m}\|_2^2 - 16s_{\Delta^*}C_f O_p\left(R_{T_1,p} + R_{T_2,p}\right)\|\mathbf{m}\|_2^2 ,$$

where we have also used the fact that f_l is Hermitian and thus by Lemma 3, $\lambda_{\min}(\Sigma_l) = \lambda_{\min}(f_l)$. For large enough T_1, T_2 the second term on the RHS $\leq$ 0.5*(first term on RHS) so we have that with high probability

$$\frac{1}{2}\mathbf{m}^T(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1)\mathbf{m} \geq \frac{\lambda_{\min}(f_1)\lambda_{\min}(f_2)}{2}\|\mathbf{m}\|_2^2 ,$$

where "large enough" depends on p, the smoothing exponents γ_i, the dependence α_i, and the minimum eigenvalues $\lambda_{\min}(f_l)$. In general, we require

$$\frac{\lambda_{\min}(f_1)\lambda_{\min}(f_2)}{2} \leq 16s_{\Delta^*}C_f O_p\left(R_{T_1,p} + R_{T_2,p}\right) .$$

Population quantities such as $\lambda_{\min}(f_1)$ are unknown so what exactly constitutes "large enough" is unknown. However, a larger dimension p, a larger smoothing exponent γ_i (and hence a large smoothing span), or larger α_i (greater dependence in the data) all increase the required T_l for "large enough" to be satisfied. Therefore we conclude that for large enough T_1, T_2, RSC holds with $\kappa_L = \lambda_{\min}(f_1)\lambda_{\min}(f_2)/2$ with high probability.

□

We are now ready to prove Theorem 4.

Proof of Theorem 4. We first proceed by establishing the decomposability of our penalty term, $\|\Delta\|_1$. Decomposability is defined in Definition 1 of Negahban et al. (2012). Recall S_{Δ^*} is the support of $\text{vec}(\Delta^*)$ and that $\overline{\mathcal{M}} = \left\{\theta : \mathbb{R}^{4p^2} \middle| \theta_{ij} \neq 0 \text{ for all } (i,j) \in S_{\Delta^*}\right\}$ and $\overline{\mathcal{M}}^{\perp} = \left\{\gamma : \mathbb{R}^{4p^2} \middle| \gamma_{ij} = 0 \text{ for all } (i,j) \in S_{\Delta^*}\right\}$. Then $\|\Delta^*\|_1$ is decomposable since $\|\theta + \gamma\|_1 = \|\theta\|_1 + \|\gamma\|_1$ for all $\theta \in \overline{\mathcal{M}}$ and $\gamma \in \overline{\mathcal{M}}^{\perp}$

Next we discuss the choice of penalty scale τ_{T_1,T_2}. To use the method in Negahban et al. (2012) we need $\tau_{T_1,T_2} \geq 2\|\nabla L_D(\Delta^*, \hat{\Sigma}_1, \hat{\Sigma}_2)\|_\infty$. We know that $\|\nabla L_D(\Delta^*, \hat{\Sigma}_1, \hat{\Sigma}_2)\|_\infty = 2\|0.5\left(\hat{\Sigma}_1\Delta^*\hat{\Sigma}_2 + \hat{\Sigma}_2\Delta^*\hat{\Sigma}_1\right) - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\|_\infty$. Let $\Gamma = 0.5\left(\Sigma_2 \otimes \Sigma_1 + \Sigma_1 \otimes \Sigma_2\right)$ and $\hat{\Gamma} = 0.5\left(\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 + \hat{\Sigma}_1 \otimes \hat{\Sigma}_2\right)$. Then we write

$$
\begin{aligned}
\|0.5\left(\hat{\Sigma}_1\Delta^*\hat{\Sigma}_2 + \hat{\Sigma}_2\Delta^*\hat{\Sigma}_1\right) - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\|_\infty &= \|\hat{\Gamma}\,\text{vec}(\Delta^*) - \left(\text{vec}(\hat{\Sigma}_2) - \text{vec}(\hat{\Sigma}_1)\right)\|_\infty \\
&= \|(\hat{\Gamma} - \Gamma)\,\text{vec}(\Delta^*) - \text{vec}\left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right) + \text{vec}\left(\Sigma_2 - \Sigma_1\right)\|_\infty \\
&= \|(\hat{\Gamma} - \Gamma)\,\text{vec}(\Delta^*) - \text{vec}\left(\hat{\Sigma}_2 - \Sigma_2\right) + \text{vec}\left(\hat{\Sigma}_1 - \Sigma_1\right)\|_\infty \\
&\leq \|(\hat{\Gamma} - \Gamma)\,\text{vec}(\Delta^*)\|_\infty + \|\hat{\Sigma}_1 - \Sigma_1\|_\infty + \|\hat{\Sigma}_2 - \Sigma_2\|_\infty \\
&\leq \|\hat{\Gamma} - \Gamma\|_\infty\|\Delta^*\|_1 + \|\hat{\Sigma}_1 - \Sigma_1\|_\infty + \|\hat{\Sigma}_2 - \Sigma_2\|_\infty
\end{aligned}
$$

Next we we use the concentration bounds established for $\|\hat{\Gamma} - \Gamma\|_\infty$ in Lemma 2 and $\|\hat{\Sigma}_1 - \Sigma_1\|_\infty, \|\hat{\Sigma}_2 - \Sigma_2\|_\infty$ in Corollary 3 to get that

$$\|0.5\left(\hat{\Sigma}_1\Delta^*\hat{\Sigma}_2+\hat{\Sigma}_2\Delta^*\hat{\Sigma}_1\right)-\left(\hat{\Sigma}_1-\hat{\Sigma}_2\right)\|_\infty = C_fO_p\left(R_{T_1,p}+R_{T_2,p}\right)\|\Delta^*\|_1+C_fO_p\left(R_{T_1,p}\right)+C_fO_p\left(R_{T_2,p}\right)$$
$$= C_fO_p\left(R_{T_1,p}+R_{T_2,p}\right)\left(1+\|\Delta^*\|_1\right)$$

Combining the above results with Lemma 4 which establishes RSC, we can apply Corollary 3 of Negahban et al. (2012) to get that, for smoothing spans $B_l \asymp \left(\frac{T_l}{\log^{4\alpha_l+2}(p\vee T_l)}\right)^{\gamma_l}$ and for $\tau_{T_1,T_2} \geq C_fO_p\left(R_{T_1,p}+R_{T_2,p}\right)\left(1+\|\Delta^*\|_1\right)$ and large enough T_1, T_2 we have with high probability

$$\|\hat{\Delta}-\Delta^*\|_F \leq \frac{6\sqrt{s_{\Delta^*}}\tau_{T_1,T_2}}{\lambda_{\min}(f_1)\lambda_{\min}(f_2)}$$

Since τ_{T_1,T_2} is $C_fO_p\left(R_{T_1,p}+R_{T_2,p}\right)\left(1+\|\Delta^*\|_1\right)$, convergence of $\hat{\Delta}$ is also $\sqrt{s_{\Delta^*}}C_fO_p\left(R_{T_1,p}+R_{T_2,p}\right)\left(1+\|\Delta^*\|_1\right)$.

We also have

$$\|\hat{\Delta}-\Delta^*\|_1 \leq \frac{24s_{\Delta^*}\tau_{T_1,T_2}}{\lambda_{\min}(f_1)\lambda_{\min}(f_2)},$$

which is $s_{\Delta^*}C_fO_p\left(R_{T_1,p}+R_{T_2,p}\right)\left(1+\|\Delta^*\|_1\right)$

□

B.3 Asymptotic distribution of spectral density estimator

The main goal of this section is to prove Proposition 1. To do so, we notice that in Zhang and Zhang (2025) Theorem 3, the asymptotic variances of the spectral density estimator have the general form of e.g., $\mathrm{Cov}(\hat{A}_{ij}(\lambda), \hat{B}_{kl}(\lambda)) = c_{ik}d_{jl} + e_{il}f_{jk}$ where $c_{ik}, d_{jl}, e_{il}, f_{jk}$ are entries from matrices C, D, E, F. For covariances of this form we work out the general structure in terms of matrices C, D, E, F. This is a similar problem to Muirhead (2009) problem 3.3. However, we need a more general result which we show below.

Result 1. *Let $S, U, A, B, C, D \in \mathbb{R}^{p\times p}$. For a matrix X we use lowercase letters with a subscript to denote the i, j entry. That is, $X := \{x_{ij}\}$ Suppose*

$$\mathrm{Cov}(u_{ij}, s_{kl}) = a_{ik}b_{jl} + c_{il}d_{jk}\,.$$

Then

$$\mathrm{Cov}(\mathrm{vec}(U), \mathrm{vec}(S)) = (B \otimes A) + K(C \otimes D)\,,$$

where $\otimes$ denotes the Kronecker product, $K = \sum_{i,j=1}^{p} H_{ij} \otimes H_{ij}^T$ and H_{ij} is the $p \times p$ matrix where $h_{ij} = 1$ and zero everywhere else.

Proof. We establish this result by showing $\mathrm{Cov}(\mathrm{vec}(U), \mathrm{vec}(S))_{ij} = [(B \otimes A) + K(C \otimes D)]_{ij}$. Throughout we let $//$ be integer division (e.g., 6 // 4 = 1) and $\%$ be the modulo operation (e.g., 6 $\%$ 4 = 2). We start by noting that for $i, j \in \{1, \ldots, p^2\}$

$$\begin{aligned}\mathrm{Cov}(\mathrm{vec}(U), \mathrm{vec}(S))_{ij} &= \mathrm{Cov}(u_{(i-1)\%p+1,(i-1)//p+1}, s_{(j-1)\%p+1,(j-1)//p+1}) \\ &= a_{(i-1)\%p+1,(j-1)\%p+1} b_{(i-1)//p+1,(j-1)//p+1} \\ &\quad + c_{(i-1)\%p+1,(j-1)//p+1} d_{(i-1)//p+1,(j-1)\%p+1}\,.\end{aligned}$$

This can be more easily seen by writing out the covariance

$$\mathrm{Cov}(\mathrm{vec}(U), \mathrm{vec}(S)) = \begin{bmatrix} \mathrm{Cov}(u_{11}, s_{11}) & \mathrm{Cov}(u_{11}, s_{21}) & \mathrm{Cov}(u_{11}, s_{31}) & \ldots & \mathrm{Cov}(u_{11}, s_{pp}) \\ \mathrm{Cov}(u_{21}, s_{11}) & \mathrm{Cov}(u_{21}, s_{21}) & \mathrm{Cov}(u_{21}, s_{31}) & \ldots & \mathrm{Cov}(u_{21}, s_{pp} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathrm{Cov}(u_{pp}, s_{11}) & \mathrm{Cov}(u_{pp}, s_{21}) & \mathrm{Cov}(u_{pp}, s_{31}) & \ldots & \mathrm{Cov}(u_{pp}, s_{pp}) \end{bmatrix}.$$

The first index of u_{ij} goes from $1, \ldots, p$ and repeats this p times (e.g., $[1, \ldots p, 1, \ldots, p, \ldots, (p-1), p]$). For i ranging from 1 to p^2, this can be recreated by $(i-1)\%p + 1$. The second index repeats 1 p times, then 2 p times (e.g., $[1, 1, \ldots, 1, 2, 2, \ldots, 2, \ldots, p, p, \ldots, p]$). For i ranging from 1 to p^2 this can recreated by $(i-1)//p + 1$.

Next we work with the right hand side. $[(B \otimes A) + K(C \otimes D)]_{ij} = (B \otimes A)_{ij} + [K(C \otimes D)]_{ij}$. We get that

$$(B \otimes A)_{ij} = b_{(i-1)//p+1,(j-1)//p+1} a_{(i-1)\%p+1,(j-1)\%p+1} \,.$$

Next we note that K has exactly one 1 entry in every row and column and is zero elsewhere. The k^{th} row of K has a 1 in position $p((k-1)\%p) + (k-1)//p + 1$ and is zero everywhere else.

To understand the structure of K better, we discuss the $H_{ij} \otimes H_{ij}^T$ matrices. These matrices are $p^2 \times p^2$ and have only one 1. The Kronecker product in general can be viewed as each element on the LHS multiplied by the entire matrix on the RHS. Thus the Kronecker product can be viewed as a block matrix with dimensions $p \times p$ where each entry is a matrix of size $p \times p$. The matrix on the LHS of the Kronecker product selects the i, j $p \times p$ block that contains the non-zero entry. Then the matrix on the RHS selects the entry of this block that will be non-zero. As an example, if $p = 3$, then we have

$$H_{21} \otimes H_{21}^T = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} .$$

In this example for $p = 3$, the first row of K has a 1 in column 1, the second has a 1 in column 4, the third in column 7, the fourth in column 2, etc. In general, $K[i,]$ has a 1 in position $p((i-1)\%p) + (i-1)//p + 1$ and is zero elsewhere.

We make two additional observations. The first is that for $i \in \{1, \ldots, p^2\}$,

$$(p((i-1)\%p) + (i-1)//p)\,//p + 1 = (i-1)\%p + 1 \,.$$

This holds because for $i \in \{1, \ldots, p^2\}$, $((i-1)//p)//p) = 0$ and $p((i-1)\%p)//p = (i-1)\%p$.

Next, observe that

$$\begin{aligned}[p((i-1)\%p) + (i-1)//p]\%p &= ((i-1)//p)\%p) \\ &= (i-1)//p\,.\end{aligned}$$

The first line holds because $p((i-1)\%p)$ is divisible by p so there is no remainder and $p((i-1)\%p)\%p = 0$. The second holds because when $i \in \{1, \ldots, p^2\}$, $(i-1)//p < p$ so taking the remainder when dividing by p will not change the result.

Then we get

$$\begin{aligned}[K(C \otimes D)]_{ij} &= \langle K_{i\cdot}, (C \otimes D)_{\cdot j} \rangle \\ &= (C \otimes D)_{p((i-1)\%p)+(i-1)//p+1,j} \\ &= (C \otimes D)_{i',j} \qquad\qquad (i' = p((i-1)\%p) + (i-1)//p + 1) \\ &= c_{(i'-1)//p+1,(j-1)//p+1} d_{(i'-1)\%p+1,(j-1)\%p+1} \\ &= c_{(i-1)\%p+1,(j-1)//p+1} d_{(i-1)//p,(j-1)\%p+1}\,.\end{aligned}$$

In the first line we use the definition of matrix multiplication, in the second we use the fact that the i^{th} row of K only has 1 non-zero entry. In the third line we change notation to $i' = p((i-1)\%p) + (i-1)//p + 1$, and in the last we use the two observations stated above where $i' - 1 = p((i-1)\%p) + (i-1)//p$

Combining the results for $(B \otimes A)_{ij}$ and $[K(C \otimes D)]_{ij}$ we get:

$$\begin{aligned}\mathrm{Cov}(\mathrm{vec}(U), \mathrm{vec}(S))_{ij} &= \mathrm{Cov}(u_{(i-1)\%p+1,(i-1)//p+1}, s_{(j-1)\%p+1,(j-1)//p+1}) \\ &= a_{(i-1)\%p+1,(j-1)\%p+1} b_{(i-1)//p+1,(j-1)//p+1} \\ &+ c_{(i-1)\%p+1,(j-1)//p+1} d_{(i-1)//p+1,(j-1)\%p+1} \\ &= (B \otimes A)_{ij} + [K(C \otimes D)]_{ij}\,.\end{aligned}$$

Which shows that $\mathrm{Cov}(\mathrm{vec}(U), \mathrm{vec}(S)) = B \otimes A + K(C \otimes D)$. □

With this result we can now proceed in establishing the form of the asymptotic covariance of the estimator $\hat{f}(\lambda) = \hat{A}(\lambda) + i\hat{B}(\lambda)$.

Proof of Proposition 1. Let $f(\lambda) = A(\lambda) + iB(\lambda)$ denote the true spectral density. T is the number of data points and B is the smoothing parameter used to smooth the periodograms and obtain our spectral density estimator. Applying Theorem 3 of Zhang and Zhang (2025) to the concatenation of all the real and imaginary entries of the estimated spectral density matrix and under $T\Phi_2^4 = o(B^3)$, which lets us substitute e.g. $E\left(\mathrm{vec}(\hat{A}(\lambda))\right)$ with $\mathrm{vec}(A(\lambda))$, we have

$$\sqrt{\frac{T}{B}} \begin{pmatrix} \mathrm{vec}(\hat{A}(\lambda)) - \mathrm{vec}(A(\lambda)) \\ \mathrm{vec}(\hat{B}(\lambda)) - \mathrm{vec}(B(\lambda)) \end{pmatrix} \xrightarrow{d} N(0, V)\,,$$

where V has the block form

$$V = \lim_{T,B\to\infty} \frac{T}{B} \begin{bmatrix} \mathrm{Cov}\left(\mathrm{vec}(\hat{A}(\lambda)), \mathrm{vec}(\hat{A}(\lambda))\right) & \mathrm{Cov}\left(\mathrm{vec}(\hat{A}(\lambda)), \mathrm{vec}(\hat{B}(\lambda))\right) \\ \mathrm{Cov}\left(\mathrm{vec}(\hat{B}(\lambda)), \mathrm{vec}(\hat{A}(\lambda))\right) & \mathrm{Cov}\left(\mathrm{vec}(\hat{B}(\lambda)), \mathrm{vec}(\hat{B}(\lambda))\right) \end{bmatrix}.$$

We simplify each of these three blocks individually. Before beginning we note that

$$\begin{aligned} f_{ik}(\lambda)f_{lj}(\lambda) &= (a_{ik} + ib_{ik})(a_{lj} + ib_{lj}) = a_{ik}a_{lj} + ia_{ik}b_{lj} + ia_{lj}b_{ik} - b_{ik}b_{lj}, \\ f_{il}(\lambda)f_{kj}(\lambda) &= (a_{il} + ib_{il})(a_{kj} + ib_{kj}) = a_{il}a_{kj} + ia_{il}b_{kj} + ia_{kj}b_{il} - b_{il}b_{kj}, \end{aligned}$$

and that $b_{ij} = -b_{ji}$, $a_{ij} = a_{ji}$ because the spectral density matrix is Hermitian. We proceed with the two cases $\lambda \in (0, \pi)$ and $\lambda = \{0, \pi\}$ separately.

Case when $\lambda \in (0, \pi)$. From Zhang and Zhang (2025), top of page 6, we know that

$$\begin{aligned} \lim_{T,B\to\infty} \frac{T}{B} \mathrm{Cov}\left(\hat{A}_{ij}, \hat{A}_{kl}\right) &= \frac{1}{2}\mathrm{Re}\left(f_{ik}(\lambda)f_{lj}(\lambda)\right) + \frac{1}{2}\mathrm{Re}\left(f_{il}(\lambda)f_{kj}(\lambda)\right) \\ &= \frac{1}{2}\left(a_{ik}a_{lj} - b_{ik}b_{lj}\right) + \frac{1}{2}\left(a_{il}a_{kj} - b_{il}b_{kj}\right) \\ &= \frac{1}{2}\left(a_{ik}a_{jl} + b_{ik}b_{jl}\right) + \frac{1}{2}\left(a_{il}a_{jk} + b_{il}b_{jk}\right). \end{aligned}$$

Using Result 1 we get

$$\lim_{T,B\to\infty}\frac{T}{B}\operatorname{Cov}\left(\operatorname{vec}(\hat{A}(\lambda)),\operatorname{vec}(\hat{A}(\lambda))\right)=\frac{1}{2}\left(I_{p^2}+K\right)\left(A(\lambda)\otimes A(\lambda)+B(\lambda)\otimes B(\lambda)\right).$$

Next we have

$$\begin{aligned}\lim_{T,B\to\infty}\frac{T}{B}\operatorname{Cov}\left(\hat{A}_{ij},\hat{B}_{kl}\right)&=\frac{1}{2}\mathbf{Im}\left(f_{il}(\lambda)f_{kj}(\lambda)\right)-\frac{1}{2}\mathbf{Im}\left(f_{ik}(\lambda)f_{lj}(\lambda)\right)\\&=\frac{1}{2}\left(a_{il}b_{kj}+a_{kj}b_{il}\right)-\frac{1}{2}\left(a_{ik}b_{lj}+a_{lj}b_{ik}\right)\\&=\frac{1}{2}\left(-a_{il}b_{jk}+b_{il}a_{jk}\right)-\frac{1}{2}\left(-a_{ik}b_{jl}+b_{ik}a_{jl}\right)\\&=\frac{1}{2}\left(a_{ik}b_{jl}-a_{il}b_{jk}-b_{ik}a_{jl}+b_{il}a_{jk}\right)\end{aligned}$$

which is simplified to

$$\lim_{T,B\to\infty}\frac{T}{B}\operatorname{Cov}\left(\operatorname{vec}(\hat{A}(\lambda)),\operatorname{vec}(\hat{B}(\lambda))\right)=\frac{1}{2}\left(I_{p^2}+K\right)\left(B(\lambda)\otimes A(\lambda)-A(\lambda)\otimes B(\lambda)\right).$$

Lastly,

$$\begin{aligned}\lim_{T,B\to\infty}\frac{T}{B}\operatorname{Cov}\left(\hat{B}_{ij},\hat{B}_{kl}\right)&=\frac{1}{2}\mathbf{Re}\left(f_{ik}(\lambda)f_{lj}(\lambda)\right)-\frac{1}{2}\mathbf{Re}\left(f_{il}(\lambda)f_{kj}(\lambda)\right)\\&=\frac{1}{2}\left(a_{ik}a_{lj}-b_{ik}b_{lj}\right)-\frac{1}{2}\left(a_{il}a_{kj}-b_{il}b_{kj}\right)\\&=\frac{1}{2}\left(a_{ik}a_{jl}-a_{il}a_{jk}+b_{ik}b_{jl}-b_{il}b_{jk}\right).\end{aligned}$$

Again using Result 1 this simplified to

$$\lim_{T,B\to\infty}\frac{T}{B}\operatorname{Cov}\left(\operatorname{vec}(\hat{B}(\lambda)),\operatorname{vec}(\hat{B}(\lambda))\right)=\frac{1}{2}\left(I_{p^2}-K\right)\left(A(\lambda)\otimes A(\lambda)+B(\lambda)\otimes B(\lambda)\right).$$

Combining these we get

$$V=\frac{1}{2}\begin{bmatrix}\left(I_{p^2}+K\right)\left(A(\lambda)\otimes A(\lambda)+B(\lambda)\otimes B(\lambda)\right) & \left(I_{p^2}+K\right)\left(B(\lambda)\otimes A(\lambda)-A(\lambda)\otimes B(\lambda)\right)\\ \left[\left(I_{p^2}+K\right)\left(B(\lambda)\otimes A(\lambda)-A(\lambda)\otimes B(\lambda)\right)\right]^T & \left(I_{p^2}-K\right)\left(A(\lambda)\otimes A(\lambda)+B(\lambda)\otimes B(\lambda)\right)\end{bmatrix}.$$

Case when $\lambda = 0$ or $\lambda = \pi$. When λ is 0 or π, the spectral density and the spectral density estimator are real-valued. Thus we have that $\hat{f}(\lambda) = \hat{A}(\lambda)$ and $f(\lambda) = A(\lambda)$. In this case we are interested in

$$\sqrt{\frac{T}{B}}\left(\text{vec}(\hat{A}(\lambda)) - \text{vec}(A(\lambda))\right) \xrightarrow{d} N(0, V)\,,$$

and that V has the form

$$V = \lim_{T,B\to\infty} \frac{T}{B}\,\text{Cov}\left(\text{vec}(\hat{A}(\lambda)), \text{vec}(\hat{A}(\lambda))\right)\,.$$

From Zhang and Zhang (2025) Theorem 3 (ii) we have

$$\begin{aligned}\lim_{T,B\to\infty} \frac{T}{B}\,\text{Cov}\left(\hat{A}_{ij}, \hat{A}_{kl}\right) &= a_{ik}a_{lj} + a_{il}a_{kj} \\ &= a_{ik}a_{jl} + a_{il}a_{jk}\,,\end{aligned}$$

where we have used that A is symmetric so $a_{ij} = a_{ji}$. This can be simplified using Result 1 as

$$V = \lim_{T,B\to\infty} \frac{T}{B}\,\text{Cov}\left(\text{vec}(\hat{A}(\lambda)), \text{vec}(\hat{A}(\lambda))\right) = (I_{p^2} + K)\,A \otimes A\,.$$

□

B.4 Generalization of inference for high-dimensional estimating equations to arbitrary asymptotic scaling

In the below, we will state all assumptions including those unchanged from Neykov et al. (2018) for compeleteness. In the following we will denote $(\theta, \gamma^*) = \beta^*_\theta$ and $\text{v}^* = [E_{\mathbf{T}}(\beta^*)]^{-1}_{1*}$.

Assumption 5. *This is the same as Neykov et al. (2018) Assumption 1 but is rewritten for completeness.*

There exists a neighborhood $\mathcal{N}_{\theta^*}$ of θ^* such that for all $\theta \in \mathcal{N}_{\theta^*}$

$$\lim_{T\to\infty} \mathbb{P}\left(\|\mathbf{t}\left(Z, \beta^*_\theta\right) - E_{\mathbf{t}}\left(\beta^*_\theta\right)\|_\infty \leq r_1(T, \theta)\right) = 1 \tag{B.7}$$

$$\lim_{T\to\infty} \mathbb{P}\left(\left|\text{v}^{*T}\mathbf{t}\left(Z, \beta^*_\theta\right) - \text{v}^{*T}E_{\mathbf{t}}\left(\beta^*_\theta\right)\right| \leq r_2(T, \theta)\right) = 1 \tag{B.8}$$

$$\lim_{T\to\infty} \mathbb{P}\left(\sup_{\nu\in[0,1]} \left\| \hat{\mathrm{v}}^T \mathbf{T}\left(Z, \tilde{\beta}_\nu\right) - \mathrm{v}^{*T} E_{\mathbf{T}}\left(\beta_\theta^*\right) \right\|_\infty \leq r_3(T,\theta) \right) = 1 \tag{B.9}$$

where $\tilde{\beta}_\nu = \nu\hat{\beta}_\theta + (1-\nu)\beta_\theta^*$, $\sup_{\theta\in\mathcal{N}_{\theta^*}} \max(r_1(T,\theta), r_2(T,\theta), r_3(T,\theta) = o(1)$, *and the following hold:*

$$\sup_{\theta\in\mathcal{N}_{\theta^*}} \|E_{\mathbf{t}}(\beta_\theta^*)\|_\infty < \infty, \qquad \sup_{\theta\in\mathcal{N}_{\theta^*}} \|\mathrm{v}^{*T}\left[E_{\mathbf{T}}(\beta_\theta^*)\right]_{-1}\|_\infty < \infty$$

Assumption 6. *This is the same as Neykov et al. (2018) Assumption 2.*

Let the estimators $\hat{\beta}$ *and* $\hat{\mathrm{v}}$ *satisfy*

$$\lim_{T\to\infty} \mathbb{P}\left(\|\hat{\beta} - \beta^*\|_1 \leq r_4(T) \right) = 1, \qquad \lim_{T\to\infty} \mathbb{P}\left(\|\hat{\mathrm{v}} - \mathrm{v}^*\|_1 \leq r_5(T) \right) = 1 \tag{B.10}$$

where $\max(r_4(T), r_5(T)) = o(1)$.

Assumption 7. *Assume that for* $\sigma^2 = \mathrm{v}^{*T} G \mathrm{v}^*$ *it holds that*

$$\sigma^{-1} s_T S(\beta^*) \xrightarrow{d} N(0,1),$$

where $G = \lim_{T\to\infty} s_T^2 Cov\left[\mathbf{t}(Z,\beta^*)\right]$ *and assume that* $\sigma^2 \geq C > 0$ *for some constant C.*

Assumption 8. *This assumption is the same as Neykov et al. (2018) Assumption 4.*

Suppose there exists a constant $\gamma > 0$ *such that:*

$$\left| \mathrm{v}^T \frac{\partial}{\partial\theta} \left[\mathbf{T}\left(Z,\beta\right)\right]_{*1} \right| \leq \psi(Z)\,,$$

for any v *and* β *satisfying* $\|\mathrm{v} - \mathrm{v}^*\|_1 < \gamma$ *and* $\|\beta - \beta^*\|_1 < \gamma$, *where* $\psi : \mathbb{R}^{n\times q} \mapsto \mathbb{R}$ *is an integrable function with* $\mathbb{E}\psi(Z) < \infty$.

Assumption 9. *This is essentially the same as Neykov et al. (2018) Assumption 5, but has* s_T *instead of* $\sqrt{T}$ *scaling. Assume the convergence rates in Assumptions 5 and 6 satisfy*

$$s_T(r_4(T)r_3(T,\theta^*) + r_5(T)r_1(T,\theta^*)) = o(1)$$

Now that we have established the necessary assumptions we are ready to state Lemma 5 which is a generalization of Lemma 1 from Neykov et al. (2018) and is an essential intermediate result.

Lemma 5. *Suppose Assumptions 5, 6, and 9 hold. Then we have the following expansion*

$$s_T \hat{S}(\hat{\beta}_{\theta^*}) = s_T S(\beta^*) + o_p(1)$$

Proof. We follow the proof of Neykov et al. (2018) Lemma 1 closely. For notational convenience we let $r_1(T) = \sup_{\theta \in \mathcal{N}_{\theta^*}} r_1(T, \theta)$ and $r_3(T) = \sup_{\theta \in \mathcal{N}_{\theta^*}} r_3(T, \theta)$.

By a Taylor expansion of $\hat{S}(\hat{\beta}_\theta)$ around β^*_θ and with a mean-value form of the remainder we get

$$\hat{S}(\hat{\beta}_\theta) = \mathrm{v}^{*T}\mathbf{t}(Z, \beta^*_\theta) + \hat{\mathrm{v}}^T\mathbf{T}(Z, \tilde{\beta}_\nu)(\hat{\beta}_\theta - \beta^*_\theta) + (\hat{\mathrm{v}} - \mathrm{v}^*)^T\mathbf{t}(Z, \beta^*_\theta) \tag{B.11}$$

We control the second term on the RHS of (B.11)

$$\left|\hat{\mathrm{v}}^T\mathbf{T}(Z, \tilde{\beta}_\nu)(\hat{\beta}_\theta - \beta^*_\theta)\right| \leq \left\|\left[\hat{\mathrm{v}}\mathbf{T}(Z, \tilde{\beta}_\nu)\right]_{-1}\right\|_\infty \|\hat{\beta}_\theta - \beta^*_\theta\|_1 \leq O_p\left(r_3(T) + \|\mathrm{v}^{*T}\left[E_{\mathbf{T}}(\beta^*_\theta)\right]_{-1}\|_\infty\right) O_p(r_4(T))$$

where $[\cdot]_{-1}$ denotes removing the first entry of a vector or the first column of a matrix. The first inequality above holds by Hölder's inequality and the fact that $\hat{\beta}_\theta - \beta^*_\theta = (\theta, \hat{\gamma}) - (\theta, \gamma^*) = (0, \hat{\gamma} - \gamma^*)$ so the first entry of the term will be 0 and can be ignored.

The $O_p(r_4(T))$ of the second inequality comes from Assumption 6. The $O_p\left(r_3(T) + \|\mathrm{v}^{*T}\left[E_{\mathbf{T}}(\beta^*_\theta)\right]_{-1}\|_\infty\right)$ term comes from writing $\left\|\left[\hat{\mathrm{v}}\mathbf{T}(Z, \tilde{\beta}_\nu)\right]_{-1}\right\|_\infty = \left\|\left[\hat{\mathrm{v}}\mathbf{T}(Z, \tilde{\beta}_\nu)\right]_{-1} - \mathrm{v}^*\left[E_{\mathbf{T}}(\beta^*_\theta)\right]_{-1} + \mathrm{v}^*\left[E_{\mathbf{T}}(\beta^*_\theta)\right]_{-1}\right\|_\infty$ and using (B.9) from Assumption 5. Note that also from the last line of Assumption 5, we have $\|\mathrm{v}^{*T}\left[E_{\mathbf{T}}(\beta^*_\theta)\right]_{-1}\|_\infty = O(1)$

Now we control the third term on the RHS of (B.11)

$$\left|(\hat{\mathrm{v}} - \mathrm{v}^*)^T\mathbf{t}(Z, \beta^*_\theta)\right| \leq \|\hat{\mathrm{v}} - \mathrm{v}^*\|_1 \|\mathbf{t}(Z, \beta^*_\theta)\|_\infty = O_p(r_5(T)) O_p\left(r_1(T) + \|E_{\mathbf{t}}(\beta^*_\theta)\|_\infty\right)$$

By Assumptions 5 and 6 we know that $\sup_{\theta \in \mathcal{N}_{\theta^*}} \max(r_1(T, \theta), r_2(T, \theta), r_3(T, \theta)) = o(1)$ and $\max(r_4(T), r_5(T)) = o(1)$. Thus terms two and three on the RHS of (B.11) are $o_p(1)$ so we have that

$$\hat{S}(\hat{\beta}_\theta) = S(\beta^*_\theta) + o_p(1) = \mathrm{v}^{*T} E_{\mathbf{t}}\left(\beta^*_\theta\right) + o_p(1) \tag{B.12}$$

The last equality follows by Assumption 5 (B.8),

$$\begin{aligned} S(\beta^*_\theta) + o_p(1) &= \mathrm{v}^{*T}\mathbf{t}\left(Z, \beta^*_\theta\right) - \mathrm{v}^{*T} E_{\mathbf{t}}\left(\beta^*_\theta\right) + \mathrm{v}^{*T} E_{\mathbf{t}}\left(\beta^*_\theta\right) + o_p(1) \\ &= O_p(r_2(T)) + \mathrm{v}^{*T} E_{\mathbf{t}}\left(\beta^*_\theta\right) + o_p(1) \\ &= \mathrm{v}^{*T} E_{\mathbf{t}}\left(\beta^*_\theta\right) + o_p(1) \end{aligned}$$

Note that $E_{\mathbf{t}}\left(\beta^*_{\theta^*}\right) = 0$ which follows from the definition of $\beta^*_{\theta^*}$ as the solution to the population estimating equation. Similarly, $\mathrm{v}^{*T}\left[E_{\mathbf{T}}\left(\beta^*_{\theta^*}\right)\right]_{-1} = \mathbf{0}$ as v^{*T} is defined to be the first row of the inverse of $E_{\mathbf{T}}\left(\beta^*_{\theta^*}\right)$ so $\mathrm{v}^{*T} E_{\mathbf{T}}\left(\beta^*_{\theta^*}\right) = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix}^T$

We plug in $\theta = \theta^*$ to get

$$s_T \hat{S}(\hat{\beta}_{\theta^*}) = s_T S(\beta^*) + s_T O_p(r_4(T) r_3(T, \theta^*) + r_5(T) r_1(T, \theta^*))$$

Using Assumption 9, that $s_T(r_4(T) r_3(T, \theta^*) + r_5(T) r_1(T, \theta^*)) = o(1)$ we get the desired result

$$s_T \hat{S}(\hat{\beta}_{\theta^*}) = s_T S(\beta^*) + o_p(1)$$

□

Lemma 6 (Consistency of de-biased estimates)**.** *Let the map $\theta \mapsto \hat{S}\left(\hat{\beta}_\theta\right)$ be continuous with a single root $\tilde{\theta}$ or non-decreasing. Further, suppose that for any $\epsilon > 0$*

$$\mathrm{v}^{*T}\left[E_{\mathbf{t}}\left(\beta^*_{\theta^* - \epsilon}\right)\right] \mathrm{v}^{*T}\left[E_{\mathbf{t}}\left(\beta^*_{\theta^* + \epsilon}\right)\right] < 0.$$

Suppose Assumptions 5 and 6 are satisfied. Then

$$\lim_{T \to \infty} \mathbb{P}(|\tilde{\theta} - \theta^*| > \epsilon) = 0\,.$$

Proof. The proof is the same as in Neykov et al. (2018) Theorem 1 and is thus omitted.

□

We now prove Theorem 5. Again we closely follow the proof in Neykov et al. (2018) but we use the arbitrary scaling s_T instead of $\sqrt{T}$ scaling.

Proof of Theorem 5. Let $U_T = \frac{s_T}{\sqrt{\mathrm{v}^{*T} G \mathrm{v}^*}}(\tilde{\theta} - \theta^*)$. We can show the statement of the proof holds for U_T as the result for $\hat{U}_T$ follows from consistency of the variance estimate and applying Slutsky's theorem. We use a Taylor expansion of $\hat{S}(\hat{\beta}_{\tilde{\theta}})$ around θ^* with a mean-value form of the remainder to get

$$0 = \hat{S}(\hat{\beta}_{\tilde{\theta}}) = \hat{S}(\hat{\beta}_{\theta^*}) + \hat{\mathrm{v}}^T \left[\mathbf{T}(Z, \hat{\beta}_{\theta^*})\right]_{*1} (\tilde{\theta} - \theta^*) + \frac{1}{2}\hat{\mathrm{v}}^T \frac{\partial}{\partial \theta}\left[\mathbf{T}(Z, \tilde{\beta}_{\nu})\right]_{*1} (\tilde{\theta} - \theta^*)^2$$

where $\tilde{\beta}_{\nu} = \nu\hat{\beta}_{\tilde{\theta}} + (1-\nu)\hat{\beta}_{\theta^*}$ for some $\nu \in [0,1]$. Note that we get the above form since our Taylor expansion is univariate. Thus $\frac{\partial}{\partial \theta}\hat{\mathrm{v}}^T \mathbf{t}(Z,\beta)|_{\beta=\hat{\beta}_{\theta^*}} = \hat{\mathrm{v}}^T \left[\mathbf{T}(Z, \hat{\beta}_{\theta^*})\right]_{*1}$.

We now show that the final term on the RHS of the Taylor expansion is $o_p(s_T^{-1})$. By Assumption 8, which allows us to apply the dominated convergence theorem, the fact that $\hat{\mathrm{v}}$ and $\tilde{\beta}_{\nu}$ are consistent, and Lemma 6 which shows that $\tilde{\theta} - \theta^* = o_p(1)$, we have that

$$\left|\frac{1}{2}\hat{\mathrm{v}}^T \frac{\partial}{\partial \theta}\left[\mathbf{T}(Z, \tilde{\beta}_{\nu})\right]_{*1} (\tilde{\theta} - \theta^*)^2\right| \leq (\tilde{\theta} - \theta^*)^2 O_p(1) = \left|\tilde{\theta} - \theta^*\right| o_p(1)$$

By Lemma 5 and Assumption 7 we have $\frac{s_T}{\sqrt{\mathrm{v}^{*T} G \mathrm{v}^*}}\hat{S}(\hat{\beta}_{\theta^*}) \xrightarrow{d} N(0,1)$. That is $s_T \hat{S}(\hat{\beta}_{\theta^*}) = O_p(1)$. Using this we can rearrange our Taylor expansion so that $s_T \hat{S}(\hat{\beta}_{\theta^*})$ is on the LHS to get that

$$O_p(1) = s_T|\tilde{\theta} - \theta^*| \left(\left|\hat{\mathrm{v}}^T \left[\mathbf{T}(Z, \hat{\beta}_{\theta^*})\right]_{*1}\right| + o_p(1)\right).$$

From Assumption 5, $\left|\hat{\mathrm{v}}^T \left[\mathbf{T}(Z, \hat{\beta}_{\theta^*})\right]_{*1}\right| = 1 + o_p(1)$. Therefore the term $\left(\left|\hat{\mathrm{v}}^T \left[\mathbf{T}(Z, \hat{\beta}_{\theta^*})\right]_{*1}\right| + o_p(1)\right) = 1 + o_p(1) + o_p(1) = 1 + o_p(1)$. Thus we get that

$$\frac{O_p(1)}{1 + o_p(1)} = s_T|\tilde{\theta} - \theta^*|$$
$$O_p(1) = s_T|\tilde{\theta} - \theta^*|,$$

where we used that $1/(1+o_p(1)) = O_p(1)$. Therefore we conclude that $\tilde{\theta} - \theta^* = O_p(s_T^{-1})$ and $\frac{1}{2}\hat{\mathrm{v}}^T \frac{\partial}{\partial \theta}\left[\mathbf{T}(Z, \tilde{\beta}_\nu)\right]_{*1} (\tilde{\theta} - \theta^*)^2 = o_p(s_T^{-1})$.

We are now ready to complete the proof. By Assumption 5 and applying Slutsky's theorem to get that $\hat{\mathrm{v}}^T \left[\mathbf{T}(Z, \hat{\beta}_{\theta^*})\right]_{*1} \to 1$, we rearrange the Taylor expansion again and conclude

$$\frac{s_T(\tilde{\theta} - \theta^*)}{\sqrt{\mathrm{v}^{*T} G \mathrm{v}^*}} = -\frac{s_T \hat{S}(\hat{\beta}_{\theta^*})}{\sqrt{\mathrm{v}^{*T} G \mathrm{v}^*}\hat{\mathrm{v}}^T \left[\mathbf{T}(Z, \hat{\beta}_{\theta^*})\right]_{*1}} + o_p(1) \xrightarrow{d} N(0,1)$$

□

B.5 Asymptotic normality of de-biased differential network estimates

In this section, we prove Theorem 6 by showing that the conditions of Theorem 5 hold for each of the estimating equations considered. Establishing the conditions of Theorem 5 requires several intermediate results that are of independent interest and are therefore stated as Lemmas.

For most of the conditions in Theorem 5 we will show that they hold for $\mathbf{t}_{\mathrm{symm}}(Z, \beta)$ only as the proofs for $\mathbf{t}_{\mathrm{s1Left}}(Z, \beta), \mathbf{t}_{\mathrm{s1Right}}(Z, \beta)$ are very similar. With this in mind, the proofs in this section will often suppress the subscript identifying the type of estimating equation. That is we will use $\mathbf{t}(Z, \beta)$ to represent $\mathbf{t}_{\mathrm{symm}}(Z, \beta)$. We do this for notation convenience. When we wish to show results for each estimating equation, we will include the identifying subscript. For example, in the case of Lemma 9, we will derive the asymptotic distribution for each estimating equation as their variances differ and subscripts are used to identify which results correspond to which estimating equation.

Recall we have assumed for ease of presentation $T_1 = T_2$ and asymptotics will be denoted simply as $T \to \infty$ with the meaning that $T_1, T_2 \to \infty$. All results should still hold as long as $T_1 \asymp T_2$. It is also worth recalling the concentration bounds of the expanded spectral densities in Corollary 3. These hold by Assumption 1 and the correct choice of smoothing spans B_l.

We begin by establishing the consistency of the Dantzig selector estimate of an arbitrary row of the second derviative of the D-trace loss. This is stated in Lemma 7 and will be used to prove Assumption 6.

Lemma 7 (Consistency of Dantzig selectors)**.** *Suppose there exists constants α_1, α_2 such that Assumption 1 is satisfied for conditions 1 and 2 respectively. Furthermore, suppose the smoothing span for condition l is $B_l \asymp \left(\frac{T_l}{\log^{4\alpha_l+2}(p \vee T_l)}\right)^{\gamma_l}$. Consider the quantities $\mathrm{v}^*_{\mathrm{symm}} := [(\Sigma_1 \otimes \Sigma_2 + \Sigma_2 \otimes \Sigma_1)/2]_{i*}$, $\mathrm{v}^*_{\mathrm{s1Left}} := [\Sigma_2 \otimes \Sigma_1]_{i*}$, and $\mathrm{v}^*_{\mathrm{s1Right}} := [\Sigma_1 \otimes \Sigma_2]_{i*}$ with corresponding Dantzig selector estimates of $\hat{\mathrm{v}}_{\mathrm{symm}}, \hat{\mathrm{v}}_{\mathrm{s1Left}}$, and $\hat{\mathrm{v}}_{\mathrm{s1Right}}$ given by*

$$\begin{aligned}
\hat{\mathrm{v}}_{\mathrm{symm}} &= \arg\min_{\mathrm{v}} \|\mathrm{v}\|_1 \text{ such that } \left\|\mathrm{v}^T \left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right)/2 - \mathbf{e}_1\right\|_\infty \leq \rho_{\mathrm{symm}} \\
\hat{\mathrm{v}}_{\mathrm{s1Left}} &= \arg\min_{\mathrm{v}} \|\mathrm{v}\|_1 \text{ such that } \left\|\mathrm{v}^T \left(\hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right) - \mathbf{e}_1\right\|_\infty \leq \rho_{\mathrm{s1Left}} \\
\hat{\mathrm{v}}_{\mathrm{s1Right}} &= \arg\min_{\mathrm{v}} \|\mathrm{v}\|_1 \text{ such that } \left\|\mathrm{v}^T \left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2\right) - \mathbf{e}_1\right\|_\infty \leq \rho_{\mathrm{s1Right}} ,
\end{aligned}$$

where $\rho_{\mathrm{symm}} \geq \left\|\mathrm{v}^*_{\mathrm{symm}}\right\|_1 C_f O_p\left(R_{T_1,p} + R_{T_2,p}\right)$, $\rho_{\mathrm{s1Left}} \geq \left\|\mathrm{v}^*_{\mathrm{s1Left}}\right\|_1 C_f O_p\left(R_{T_1,p} + R_{T_2,p}\right)$, and $\rho_{\mathrm{s1Right}} \geq \left\|\mathrm{v}^*_{\mathrm{s1Right}}\right\|_1 C_f O_p\left(R_{T_1,p} + R_{T_2,p}\right)$. Then for large enough T_1, T_2 and with high probability,

$$\begin{aligned}
\|\hat{\mathrm{v}}_{\mathrm{symm}} - \mathrm{v}^*_{\mathrm{symm}}\|_1 &= s_{\mathrm{v}^*_{\mathrm{symm}}} \left\|\mathrm{v}^*_{\mathrm{symm}}\right\|_1 C_f O_p\left(R_{T_1,p} + R_{T_2,p}\right) \\
\|\hat{\mathrm{v}}_{\mathrm{s1Left}} - \mathrm{v}^*_{\mathrm{s1Left}}\|_1 &= s_{\mathrm{v}^*_{\mathrm{s1Left}}} \left\|\mathrm{v}^*_{\mathrm{s1Left}}\right\|_1 C_f O_p\left(R_{T_1,p} + R_{T_2,p}\right) \\
\|\hat{\mathrm{v}}_{\mathrm{s1Right}} - \mathrm{v}^*_{\mathrm{s1Right}}\|_1 &= s_{\mathrm{v}^*_{\mathrm{s1Right}}} \left\|\mathrm{v}^*_{\mathrm{s1Right}}\right\|_1 C_f O_p\left(R_{T_1,p} + R_{T_2,p}\right) ,
\end{aligned}$$

where $R_{T_1,p}, R_{T_2,p}$ are the convergence rates from applying Theorem 3 to conditions 1 and 2 respectively, $C_f = \max(\|f_1\|_\infty, \|f_2\|_\infty)$, and $s_{\mathrm{v}^*_{\mathrm{symm}}} = \left\|\mathrm{v}^*_{\mathrm{symm}}\right\|_0$, $s_{\mathrm{v}^*_{\mathrm{s1Left}}} = \|\mathrm{v}^*_{\mathrm{s1Left}}\|_0$, $s_{\mathrm{v}^*_{\mathrm{s1Right}}} = \left\|\mathrm{v}^*_{\mathrm{s1right}}\right\|_0$.

Proof. We will only prove this result for the 'symm' case as 's1Left' and 's1Right' are similar. With this in mind, we suppress the subscript 'symm' for notational convenience. It is noted in Candes and Tao (2007) and Bickel et al. (2009) that for any v that satisfies the Dantzig constraint

$$\left\|\mathrm{v}^T \left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right)/2 - \mathbf{e}_1\right\|_\infty \leq \rho\,, \tag{B.13}$$

we have $\|(\hat{\mathrm{v}} - \mathrm{v})_{S_{\mathrm{v}}^c}\|_1 \leq \|(\hat{\mathrm{v}} - \mathrm{v})_{S_{\mathrm{v}}}\|_1$ where $\delta_{S_{\mathrm{v}}}$ selects the elements in δ whose indices are in S_{v}, the set of non-zero indices of v.

Next we show that for the correct choice of ρ, v^* satisfies the Dantzig constraint with high probability. That is, with high probability,

$$\left\| \mathrm{v}^* \left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1 \right) /2 - e_1 \right\|_\infty \leq \rho \,.$$

Begin by noting

$$\begin{aligned} \left\| \mathrm{v}^* \left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1 \right) /2 - e_1 \right\|_\infty &= \left\| \mathrm{v}^* \left((\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1)/2 - (\Sigma_1 \otimes \Sigma_2 + \Sigma_2 \otimes \Sigma_1)/2 \right) \right\|_\infty \\ &\leq \left\| \mathrm{v}^* \left((\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1)/2 - (\Sigma_1 \otimes \Sigma_2 + \Sigma_2 \otimes \Sigma_1)/2 \right) \right\|_1 \\ &\leq \|\mathrm{v}^*\|_1 \left\| (\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1)/2 - (\Sigma_1 \otimes \Sigma_2 + \Sigma_2 \otimes \Sigma_1)/2 \right\|_\infty . \end{aligned}$$

The last line follows from Hölder's inequality. By Lemma 2, we have that

$$\left\| \mathrm{v}^* \left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1 \right) /2 - e_1 \right\|_\infty \leq \|\mathrm{v}^*\|_1 C_f O_p \left(R_{T_1,p} + R_{T_2,p} \right) .$$

So if we set $\rho \geq \|\mathrm{v}^*\|_1 C_f O_p \left(R_{T_1,p} + R_{T_2,p} \right)$ then for large enough T_1, T_2 and with high probability v^* satisfies the Dantzig constraint. That is,

$$\left\| \mathrm{v}^* \left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1 \right) /2 - e_1 \right\|_\infty \leq \rho \,.$$

Next we show that restricted strong convexity is satisfied for our Dantzig selector. Showing this proceeds almost exactly the same as showing RSC for the D-trace loss. The main difference is that we now are interested in showing RSC for vectors $\mathbf{m} \in \mathcal{C}(\overline{\mathcal{M}}, \overline{\mathcal{M}}^\perp; \mathrm{v}^*) := \left\{ \mathbf{m} \in \mathbb{R}^{4p^2} \middle| \|\mathbf{m}_{S_{\mathrm{v}^*}^c}\|_1 \leq \|\mathbf{m}_{S_{\mathrm{v}^*}}\|_1 \right\}$. Recall that we showed that v^* satisfies the Dantzig constraint with high probability. Then for $\delta = \hat{\mathrm{v}} - \mathrm{v}^*$, we know that $\|\delta_{S_{\mathrm{v}^*}^c}\|_1 \leq \|\delta_{S_{\mathrm{v}^*}}\|_1$ since $\hat{\mathrm{v}}$ is the Dantzig selector and we showed v^* satisfies the Dantzig constraint with high probability for the correct choice of ρ. Then it is true that, with high probability, $\delta \in \mathcal{C}(\overline{\mathcal{M}}, \overline{\mathcal{M}}^\perp; \mathrm{v}^*)$. Showing the RSC condition proceeds the same way as showing RSC for the D-trace loss. It is rewritten here for

completeness.

$$\begin{aligned}\mathbf{m}^T(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2)\mathbf{m} &\geq \mathbf{m}^T(\Sigma_1 \otimes \Sigma_2)\mathbf{m} + \mathbf{m}^T\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\right)\mathbf{m} \\ &\geq \lambda_{\min}(\Sigma_1)\lambda_{\min}(\Sigma_2)\|\mathbf{m}\|_2^2 - \left|\mathbf{m}^T\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\right)\mathbf{m}\right| \\ &\geq \lambda_{\min}(\Sigma_1)\lambda_{\min}(\Sigma_2)\|\mathbf{m}\|_2^2 - \|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\|_\infty\|\mathbf{m}\|_1^2 \\ &\geq \lambda_{\min}(\Sigma_1)\lambda_{\min}(\Sigma_2)\|\mathbf{m}\|_2^2 - 4s_{\mathrm{v}^*}\|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\|_\infty\|\mathbf{m}\|_2^2\,.\end{aligned} \tag{B.14}$$

Note that in the last line we get $4s_{\mathrm{v}^*}$ compared to $16s_{\Delta^*}$ in the D-trace RSC. This is because we have $\|\mathbf{m}\|_1 = \|\mathbf{m}_{S^c_{\mathrm{v}^*}}\|_1 + \|\mathbf{m}_{S_{\mathrm{v}^*}}\|_1 \leq 2\|\mathbf{m}_{S_{\mathrm{v}^*}}\|_1 \leq 2\sqrt{s_{\mathrm{v}^*}}\|\mathbf{m}_{S_{\mathrm{v}^*}}\|_2 \leq 2\sqrt{s_{\mathrm{v}^*}}\|\mathbf{m}\|_2$. Again, we get the exact same inequality for $\mathbf{m}^T(\hat{\Sigma}_2 \otimes \hat{\Sigma}_1)\mathbf{m}$. Therefore, for large enough T_1, T_2 we have that with high probability

$$\mathbf{m}^T 0.5(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1)\mathbf{m} \geq \frac{\lambda_{\min}(f_1)\lambda_{\min}(f_2)}{2}\|\mathbf{m}\|_2^2\,,$$

and we conclude that RSC holds with $\kappa_L = \lambda_{\min}(f_1)\lambda_{\min}(f_2)/2$.

To find the convergence rate of $\hat{\mathrm{v}}$ to v^* in ℓ_1 norm, we proceed similar to the proof of Theorem 7.1 in Bickel et al. (2009).

Define $\delta = \hat{\mathrm{v}} - \mathrm{v}^*$. We showed that for high probability v^* satisfies the Dantzig constraint andby definition $\hat{\mathrm{v}}$ satisfies the RSC condition. Thus with high probability and for large enough T_1, T_2,

$$\begin{aligned}\left\|\delta^T\frac{\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right)}{2}\right\|_\infty &= \left\|(\hat{\mathrm{v}} - \mathrm{v}^*)^T\frac{\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right)}{2}\right\|_\infty \\ &= \left\|\hat{\mathrm{v}}^T\frac{\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right)}{2} - e_1 + e_1 - \mathrm{v}^{*,T}\frac{\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right)}{2}\right\|_\infty \\ &\leq \left\|\hat{\mathrm{v}}^T\frac{\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right)}{2} - e_1\right\|_\infty + \left\|\mathrm{v}^{*,T}\frac{\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right)}{2} - e_1\right\|_\infty \\ &\leq 2\rho\,.\end{aligned}$$

Proceeding with showing convergence of $\hat{\mathrm{v}}$ to v^* in L1 norm we get

$$\begin{aligned}
\delta^T \frac{\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right)}{2} \delta &\leq \left| \delta^T \frac{\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right)}{2} \delta \right| \\
&\leq \left\| \delta^T \frac{\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right)}{2} \right\|_\infty \|\delta\|_1 \\
&\leq 2\rho \|\delta\|_1 \\
&\leq 4\rho \|\delta_{S_{\mathrm{v}^*}}\|_1 \\
&\leq 4\rho \sqrt{s_{\mathrm{v}^*}} \|\delta_{S_{\mathrm{v}^*}}\|_2 \,.
\end{aligned}$$

From the RSC condition we get that

$$\frac{\lambda_{\min}(f_1)\lambda_{\min}(f_2)}{2} \|\delta\|_2^2 \leq \delta^T \frac{\left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1\right)}{2} \delta \leq 4\rho \sqrt{s_{\mathrm{v}^*}} \|\delta_{S_{\mathrm{v}^*}}\|_2 \,.$$

Noting that $\|\delta_{S_{\mathrm{v}^*}}\|_2^2 \leq \|\delta\|_2^2$ we get

$$\frac{\lambda_{\min}(f_1)\lambda_{\min}(f_2)}{2} \|\delta_{S_{\mathrm{v}^*}}\|_2^2 \leq 4\rho \sqrt{s_{\mathrm{v}^*}} \|\delta_{S_{\mathrm{v}^*}}\|_2 \,,$$

which implies

$$\|\delta_{S_{\mathrm{v}^*}}\|_2 \leq \frac{8\rho \sqrt{s_{\mathrm{v}^*}}}{\lambda_{\min}(f_1)\lambda_{\min}(f_2)} \,.$$

Using the fact that $\|\hat{\mathrm{v}} - \mathrm{v}^*\|_1 = \|\delta\|_1 \leq 2\|\delta_{S_{\mathrm{v}^*}}\|_1 \leq 2\sqrt{s_{\mathrm{v}^*}} \|\delta_{S_{\mathrm{v}^*}}\|_2$ and the fact that $\rho = \|\mathrm{v}^*\|_1 C_f O_p \left(R_{T_1,p} + R_{T_2,p}\right)$ we get

$$\|\hat{\mathrm{v}} - \mathrm{v}^*\|_1 \leq \frac{16 s_{\mathrm{v}^*} \|\mathrm{v}^*\|_1 C_f O_p \left(R_{T_1,p} + R_{T_2,p}\right)}{\lambda_{\min}(f_1)\lambda_{\min}(f_2)} \,,$$

from which we conclude

$$\|\hat{\mathrm{v}} - \mathrm{v}^*\|_1 = s_{\mathrm{v}^*} \|\mathrm{v}^*\|_1 C_f O_p \left(R_{T_1,p} + R_{T_2,p}\right) \,.$$

□

Next we compute the asymptotic distribution of $\text{vec}\left(\hat{\Sigma}_i - \Sigma_i\right)$ which is subsequently used to establish the asymptotic distribution of the de-biased estimating equations in Lemma 9 and in turn Assumption 7.

Lemma 8 (Asymptotic distribution of $\hat{\Sigma}$)**.** *Suppose the conditions of Proposition 1 are satisfied. Let* $\hat{\Sigma}(\lambda) = \begin{bmatrix} \hat{A}(\lambda) & -\hat{B}(\lambda) \\ \hat{B}(\lambda) & \hat{A}(\lambda) \end{bmatrix}$ *and* $\Sigma(\lambda) = \begin{bmatrix} A(\lambda) & -B(\lambda) \\ B(\lambda) & A(\lambda) \end{bmatrix}$*. Then*

(i) If $\lambda \in (0, \pi)$*,*

$$\sqrt{\frac{T}{B}}\left(\text{vec}(\hat{\Sigma}(\lambda)) - \text{vec}(\Sigma(\lambda))\right) \to N(0, V)\,,$$

where $V = PV_fP^T$*,* V_f *is the asymptotic variance from Proposition 1, and* $P = \begin{bmatrix} P_1^T & P_2^T \end{bmatrix}^T$ *with*

$$P_1 = \begin{bmatrix} I_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} & 0_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} \\ 0_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} & I_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} \\ 0_{p\times p} & I_{p\times p} & \dots & 0_{p\times p} & 0_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} \\ 0_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} & 0_{p\times p} & I_{p\times p} & \dots & 0_{p\times p} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_{p\times p} & 0_{p\times p} & \dots & I_{p\times p} & 0_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} \\ 0_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} & 0_{p\times p} & 0_{p\times p} & \dots & I_{p\times p} \end{bmatrix}.$$

We define $I_{p\times p}$ *to be the identity matrix of dimension* $\mathbb{R}^{p\times p}$*. We also define* $0_{p\times p}$ *to be the matrix of 0s of dimension* $\mathbb{R}^{p\times p}$*. The first row of block matrices in* $P_1 \in \mathbb{R}^{2p^2\times 2p^2}$ *consists of a* $I_{p\times p}$ *block followed by* $2p-1$ $0_{p\times p}$ *blocks. The second row consists of* p $0_{p\times p}$ *blocks followed by an* $I_{p\times p}$ *block and* $p-1$ $0_{p\times p}$ *blocks.* $P_2 \in \mathbb{R}^{2p^2\times 2p^2}$ *is similar to* P_1

$$P_2 = \begin{bmatrix} 0_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} & -I_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} \\ I_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} & 0_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} \\ 0_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} & 0_{p\times p} & -I_{p\times p} & \dots & 0_{p\times p} \\ 0_{p\times p} & I_{p\times p} & \dots & 0_{p\times p} & 0_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} & 0_{p\times p} & 0_{p\times p} & \dots & -I_{p\times p} \\ 0_{p\times p} & 0_{p\times p} & \dots & I_{p\times p} & 0_{p\times p} & 0_{p\times p} & \dots & 0_{p\times p} \end{bmatrix}.$$

(ii) If $\lambda = 0$ or π then

$$\sqrt{\frac{T}{B}}\left(\text{vec}(\hat{\Sigma}(\lambda)) - \text{vec}(\Sigma(\lambda))\right) \to N(0, V)\,,$$

where V is the asymptotic variance from Proposition 1 ((ii)).

Proof. We will solve first for the case where $\lambda \in (0, \pi)$. Note that for each condition l

$$\text{vec}(\hat{\Sigma}_l(\lambda)) - \text{vec}(\Sigma_l(\lambda)) = \begin{pmatrix} \text{vec}\left(\begin{bmatrix} \hat{A}_l(\lambda) \\ \hat{B}_l(\lambda) \end{bmatrix}\right) - \text{vec}\left(\begin{bmatrix} A_l(\lambda) \\ B_l(\lambda) \end{bmatrix}\right) \\ \text{vec}\left(\begin{bmatrix} -\hat{B}_l(\lambda) \\ \hat{A}_l(\lambda) \end{bmatrix}\right) - \text{vec}\left(\begin{bmatrix} -B_l(\lambda) \\ A_l(\lambda) \end{bmatrix}\right) \end{pmatrix} = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \begin{pmatrix} \text{vec}(\hat{A}(\lambda)) - \text{vec}(A(\lambda)) \\ \text{vec}(\hat{B}(\lambda)) - \text{vec}(B(\lambda)) \end{pmatrix}.$$

Applying Proposition 1 and the multivariate delta method establishes this result. When $\lambda = 0$ or $\lambda = \pi$ we get that the spectral density and its estimates are real-valued so the expansion to the real space is not necessary. This means that we do not need the pre-multiplication by $\begin{bmatrix} P_1^T & P_2^T \end{bmatrix}^T$. In this case we get that

$$\text{vec}(\hat{\Sigma}_l(\lambda)) - \text{vec}(\Sigma_l(\lambda)) = \text{vec}(\hat{A}_l(\lambda)) - \text{vec}(A_l(\lambda))\,.$$

□

Lemma 9 (Asymptotic distribution of de-biased estimating equations)**.** *Suppose that Assumption 1 is satisfied for both conditions with values α_1, α_2. Define $\alpha = \max(\alpha_1, \alpha_2)$. Furthermore suppose*

that $T_1 = T_2 := T$, that the smoothing span for both conditions is $B \asymp \left(\frac{T}{\log^{4\alpha+2}(p \vee T)}\right)^{\gamma}$ for $\gamma > 1/3$ and that the conditions of Proposition 1 are satisfied. Then

$$\sigma_{\text{symm}}^{-1}\sqrt{\frac{T}{B}}\mathrm{v}_{\text{symm}}^{*T}\,\text{vec}\left[\left(\hat{\Sigma}_2\Delta^*\hat{\Sigma}_1 + \hat{\Sigma}_1\Delta^*\hat{\Sigma}_2\right)/2 - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right] \rightsquigarrow N(0,1)$$
$$\sigma_{\text{s1Left}}^{-1}\sqrt{\frac{T}{B}}\mathrm{v}_{\text{s1Left}}^{*T}\,\text{vec}\left[\hat{\Sigma}_1\Delta^*\hat{\Sigma}_2 - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right] \rightsquigarrow N(0,1)$$
$$\sigma_{\text{s1Right}}^{-1}\sqrt{\frac{T}{B}}\mathrm{v}_{\text{s1Right}}^{*T}\,\text{vec}\left[\hat{\Sigma}_2\Delta^*\hat{\Sigma}_1 - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right] \rightsquigarrow N(0,1)\,,$$

where

$$\sigma_{\text{symm}}^2 = \mathrm{v}_{\text{symm}}^{*T}\left(M_{1,\text{symm}}V_1M_{1,\text{symm}}^T + M_{2,\text{symm}}V_2M_{2,\text{symm}}^T\right)\mathrm{v}_{\text{symm}}^*$$
$$\sigma_{\text{s1Left}}^2 = \mathrm{v}_{\text{s1Left}}^{*T}\left(M_{1,\text{s1Left}}V_1M_{1,\text{s1Left}}^T + M_{2,\text{s1Left}}V_2M_{2,\text{s1Left}}^T\right)\mathrm{v}_{\text{s1Left}}^*$$
$$\sigma_{\text{s1Right}}^2 = \mathrm{v}_{\text{s1Right}}^{*T}\left(M_{1,\text{s1Right}}V_1M_{1,\text{s1Right}}^T + M_{2,\text{s1Right}}V_2M_{2,\text{s1Right}}^T\right)\mathrm{v}_{\text{s1Right}}^*\,.$$

The notation V_l is used to represent the asymptotic variances from Lemma 8 for condition l. The other components of these variances are defined as

$$M_{1,\text{symm}} = \frac{\left(\Sigma_2\Delta^{*T}\otimes I_{2p}\right) + \left(I_{2p}\otimes\Sigma_2\Delta^*\right)}{2} + I_{4p^2} \qquad M_{2,\text{symm}} = \frac{\left(\Sigma_1^T\Delta^{*T}\otimes I_{2p}\right) + \left(I_{2p}\otimes\Sigma_1\Delta^*\right)}{2} - I_{4p^2}$$
$$M_{1,\text{s1Left}} = \Sigma_2^T\Delta^{*T}\otimes I_{2p} + I_{4p^2} \qquad M_{2,\text{s1Left}} = I_{2p}\otimes\Sigma_1\Delta^* - I_{4p^2}$$
$$M_{1,\text{s1Right}} = I_{2p}\otimes\Sigma_2\Delta^* + I_{4p^2} \qquad M_{2,\text{s1Right}} = \Sigma_1^T\Delta^{*T}\otimes I_{2p} - I_{4p^2}\,,$$

and

$$\mathrm{v}_{\text{symm}}^{*T} = \left[\frac{1}{2}\left(\Sigma_1\otimes\Sigma_2 + \Sigma_2\otimes\Sigma_1\right)\right]_{1*}^{-1}$$
$$\mathrm{v}_{\text{s1Left}}^{*T} = \left[\Sigma_2\otimes\Sigma_1\right]_{1*}^{-1}$$
$$\mathrm{v}_{\text{s1Right}}^{*T} = \left[\Sigma_1\otimes\Sigma_2\right]_{1*}^{-1}\,.$$

Proof. **Symmetric estimating equation**. We begin by expanding the symmetric estimating equation $\text{vec}\left[\left(\hat{\Sigma}_2\Delta^*\hat{\Sigma}_1 + \hat{\Sigma}_1\Delta^*\hat{\Sigma}_2\right)/2 - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right]$ into a form that allows us to find the asymptotic distribution.

$$
\begin{aligned}
\operatorname{vec}\left[\frac{1}{2}\left(\hat{\Sigma}_2\Delta^*\hat{\Sigma}_1+\hat{\Sigma}_1\Delta^*\hat{\Sigma}_2\right)-\left(\hat{\Sigma}_2-\hat{\Sigma}_1\right)\right] &= \operatorname{vec}\left[\hat{\Sigma}_2\Delta^*\left(\hat{\Sigma}_1-\Sigma_1\right)\right]/2+\operatorname{vec}\left[\hat{\Sigma}_1\Delta^*\left(\hat{\Sigma}_2-\Sigma_2\right)\right]/2 \\
&\quad+\operatorname{vec}\left[\hat{\Sigma}_2\Delta^*\Sigma_1\right]/2+\operatorname{vec}\left[\hat{\Sigma}_1\Delta^*\Sigma_2\right]/2-\operatorname{vec}\left[\hat{\Sigma}_2-\hat{\Sigma}_1\right] \\
&= \operatorname{vec}\left[\left(\hat{\Sigma}_2-\Sigma_2\right)\Delta^*\left(\hat{\Sigma}_1-\Sigma_1\right)\right]/2 \\
&\quad+\operatorname{vec}\left[\left(\hat{\Sigma}_1-\Sigma_1\right)\Delta^*\left(\hat{\Sigma}_2-\Sigma_2\right)\right]/2 \\
&\quad+\operatorname{vec}\left[\hat{\Sigma}_2\Delta^*\Sigma_1\right]/2+\operatorname{vec}\left[\hat{\Sigma}_1\Delta^*\Sigma_2\right]/2 \\
&\quad+\operatorname{vec}\left[\Sigma_2\Delta^*\left(\hat{\Sigma}_1-\Sigma_1\right)\right]/2 \\
&\quad+\operatorname{vec}\left[\Sigma_1\Delta^*\left(\hat{\Sigma}_2-\Sigma_2\right)\right]/2-\operatorname{vec}\left[\hat{\Sigma}_2-\hat{\Sigma}_1\right] \\
&\quad+\operatorname{vec}\left[\Sigma_2-\Sigma_1\right]-\operatorname{vec}\left[\Sigma_2-\Sigma_1\right] \\
&= \operatorname{vec}\left[\left(\hat{\Sigma}_2-\Sigma_2\right)\Delta^*\left(\hat{\Sigma}_1-\Sigma_1\right)\right]/2 \\
&\quad+\operatorname{vec}\left[\left(\hat{\Sigma}_1-\Sigma_1\right)\Delta^*\left(\hat{\Sigma}_2-\Sigma_2\right)\right]/2 \\
&\quad+\operatorname{vec}\left[\hat{\Sigma}_2\Delta^*\Sigma_1\right]/2+\operatorname{vec}\left[\hat{\Sigma}_1\Delta^*\Sigma_2\right]/2-\operatorname{vec}\left[\Sigma_2-\Sigma_1\right] \\
&\quad+\operatorname{vec}\left[\Sigma_2\Delta^*\left(\hat{\Sigma}_1-\Sigma_1\right)\right]/2+\operatorname{vec}\left[\Sigma_1\Delta^*\left(\hat{\Sigma}_2-\Sigma_2\right)\right]/2 \\
&\quad-\operatorname{vec}\left[\left(\hat{\Sigma}_2-\Sigma_2\right)-\left(\hat{\Sigma}_1-\Sigma_1\right)\right] \\
&= \operatorname{vec}\left[\left(\hat{\Sigma}_2-\Sigma_2\right)\Delta^*\left(\hat{\Sigma}_1-\Sigma_1\right)\right]/2 \\
&\quad+\operatorname{vec}\left[\left(\hat{\Sigma}_1-\Sigma_1\right)\Delta^*\left(\hat{\Sigma}_2-\Sigma_2\right)\right]/2 \\
&\quad+\operatorname{vec}\left[\Sigma_2\Delta^*\Sigma_1\right]/2+\operatorname{vec}\left[\Sigma_1\Delta^*\Sigma_2\right]/2-\operatorname{vec}\left[\Sigma_2-\Sigma_1\right] \\
&\quad+\operatorname{vec}\left[\left(\hat{\Sigma}_2-\Sigma_2\right)\Delta^*\Sigma_1\right]/2+\operatorname{vec}\left[\Sigma_1\Delta^*\left(\hat{\Sigma}_2-\Sigma_2\right)\right]/2 \\
&\quad-\operatorname{vec}\left[\hat{\Sigma}_2-\Sigma_2\right] \\
&\quad+\operatorname{vec}\left[\left(\hat{\Sigma}_1-\Sigma_1\right)\Delta^*\Sigma_2\right]/2+\operatorname{vec}\left[\Sigma_2\Delta^*\left(\hat{\Sigma}_1-\Sigma_1\right)\right]/2 \\
&\quad+\operatorname{vec}\left[\hat{\Sigma}_1-\Sigma_1\right].
\end{aligned}
$$

Now we will show that max norm of the first line in the last equation is $o_p\left(\sqrt{\frac{B}{T}}\right)$ and so is

$o_p(1)$ when we scale by $\sqrt{\frac{T}{B}}$.

$$
\begin{aligned}
&\left\|\text{vec}\left[\left(\hat{\Sigma}_2-\Sigma_2\right)\Delta^*\left(\hat{\Sigma}_1-\Sigma_1\right)\right]/2+\text{vec}\left[\left(\hat{\Sigma}_1-\Sigma_1\right)\Delta^*\left(\hat{\Sigma}_2-\Sigma_2\right)\right]/2\right\|_\infty \\
&=\left\|\frac{1}{2}\left(\left(\hat{\Sigma}_1-\Sigma_1\right)^T\otimes\left(\hat{\Sigma}_2-\Sigma_2\right)+\left(\hat{\Sigma}_2-\Sigma_2\right)^T\otimes\left(\hat{\Sigma}_1-\Sigma_1\right)\right)\text{vec}\left(\Delta^*\right)\right\|_\infty \\
&\leq\frac{1}{2}\left\|\left(\hat{\Sigma}_1-\Sigma_1\right)^T\otimes\left(\hat{\Sigma}_2-\Sigma_2\right)+\left(\hat{\Sigma}_2-\Sigma_2\right)^T\otimes\left(\hat{\Sigma}_1-\Sigma_1\right)\right\|_\infty\left\|\text{vec}\left(\Delta^*\right)\right\|_1 \\
&\leq\frac{1}{2}\left(\left\|\hat{\Sigma}_1-\Sigma_1\right\|_\infty\left\|\hat{\Sigma}_2-\Sigma_2\right\|_\infty+\left\|\hat{\Sigma}_2-\Sigma_2\right\|_\infty\left\|\hat{\Sigma}_1-\Sigma_1\right\|_\infty\right)\left\|\text{vec}\left(\Delta^*\right)\right\|_1 \\
&=O_p\left(R_{T_1,p}R_{T_2,p}\left\|\text{vec}\left(\Delta^*\right)\right\|_1\right)\,.
\end{aligned}
$$

Since Assumption 1 is satisfied for both conditions, in the last line we applied Corollary 3 to both conditions and write the rates as $R_{T_1,p}, R_{T_2,p}$. To show that this term is $o_p\left(\sqrt{\frac{T}{B}}\right)$ it suffices to show that $\sqrt{\frac{T}{B}}O_p\left(R_{T_1,p}R_{T_2,p}\left\|\text{vec}\left(\Delta^*\right)\right\|_1\right)=o_p(1)$. Note the choice of smoothing span is $B\asymp\left(\frac{T}{\log^{4\alpha+2}(p\vee T)}\right)^\gamma$ for $\gamma>1/3$ in order to satisfy the conditions in Proposition 1. Thus we have that

$$
\sqrt{\frac{T}{B}}=\left(\frac{T}{\frac{T^\gamma}{\log^{\gamma(4\alpha+2)}(p\vee T)}}\right)^{1/2}=T^{\frac{1-\gamma}{2}}\log^{\gamma(2\alpha+1)}(p\vee T)\,.
$$

Since we have assumed the same T and smoothing span in each group we can simplify $R_{T_1,p}R_{T_2,p}$ as follows

$$
\begin{aligned}
R_{T_1,p}R_{T_2,p}&=\left\|X_{1,\cdot}\right\|_{\psi_\alpha}^2\left\|X_{2,\cdot}\right\|_{\psi_\alpha}^2\left(2^{\alpha+1}T^{\frac{\gamma-1}{2}}\log^{(1-\gamma)(2\alpha+1)}(p\vee T)\right)^2 \\
&=2^{2\alpha+2}\left\|X_{1,\cdot}\right\|_{\psi_\alpha}^2\left\|X_{2,\cdot}\right\|_{\psi_\alpha}^2T^{\gamma-1}\log^{2(1-\gamma)(2\alpha+1)}(p\vee T)\,.
\end{aligned}
$$

Combining this with $\sqrt{\frac{T}{B}}$ we get

$$
\begin{aligned}
\sqrt{\frac{T}{B}}O_p\left(R_{T_1,p}R_{T_2,p}\left\|\text{vec}\left(\Delta^*\right)\right\|_1\right)&=O_p\left(2^{2\alpha+2}\left\|X_{1,\cdot}\right\|_{\psi_\alpha}^2\left\|X_{2,\cdot}\right\|_{\psi_\alpha}^2\left\|\text{vec}\left(\Delta^*\right)\right\|_1T^{\frac{\gamma-1}{2}}\log^{(2\alpha+1)(2-\gamma)}\left(p\vee T\right)\right) \\
&=o_p(1)\,.
\end{aligned}
$$

Going back to our expanded form, by definition of Δ^*, the second line is 0. Then we will simplify lines 3 & 4 and 5 & 6 of the expanded symmetric estimating equation using the fact that $\text{vec}(ABC) = (I \otimes AB)\,\text{vec}(C) = (C^T B^T \otimes I)\,\text{vec}(A)$. Using this we get

$$
\begin{aligned}
&= o_p\left(\sqrt{\frac{B}{T}}\right) \\
&\quad + 0 \\
&\quad + \left[\frac{\left(\Sigma_1^T \Delta^{*,T} \otimes I_{2p}\right) + \left(I_{2p} \otimes \Sigma_1 \Delta^*\right)}{2} - I_{4p^2}\right] \text{vec}\left(\hat{\Sigma}_2 - \Sigma_2\right) \\
&\quad + \left[\frac{\left(\Sigma_2^T \Delta^{*,T} \otimes I_{2p}\right) + \left(I_{2p} \otimes \Sigma_2 \Delta^*\right)}{2} + I_{4p^2}\right] \text{vec}\left(\hat{\Sigma}_1 - \Sigma_1\right) .
\end{aligned}
$$

We define

$$
\begin{aligned}
M_{2,\text{symm}} &= \frac{\left(\Sigma_1^T \Delta^{*,T} \otimes I_{2p}\right) + \left(I_{2p} \otimes \Sigma_1 \Delta^*\right)}{2} - I_{4p^2} \\
M_{1,\text{symm}} &= \frac{\left(\Sigma_2^T \Delta^{*,T} \otimes I_{2p}\right) + \left(I_{2p} \otimes \Sigma_2 \Delta^*\right)}{2} + I_{4p^2} .
\end{aligned}
$$

We further denote the asymptotic variance of the appropriately scaled versions of $\text{vec}\left(\hat{\Sigma}_2 - \Sigma_2\right)$ and $\text{vec}\left(\hat{\Sigma}_1 - \Sigma_1\right)$ as V_2 and V_1 respectively. Assuming that data in conditions 1 and 2 is independent we have from Proposition 1 that

$$
\begin{aligned}
&\sqrt{\frac{T}{B}} \text{v}_{\text{symm}}^{*T} \text{vec}\left[\frac{\left(\hat{\Sigma}_2 \Delta^* \hat{\Sigma}_1 + \hat{\Sigma}_1 \Delta^* \hat{\Sigma}_2\right)}{2} - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right] \\
&\qquad \rightsquigarrow N\left(0, \text{v}_{\text{symm}}^{*T}\left(M_{1,\text{symm}} V_1 M_{1,\text{symm}}^T + M_{2,\text{symm}} V_2 M_{2,\text{symm}}^T\right) \text{v}_{\text{symm}}^*\right) .
\end{aligned}
$$

s1Left estimating equation. Similar to the symmetric estimating equation case, we simplify the s1Left estimating equation

$$
\text{vec}\left[\hat{\Sigma}_1 \Delta^* \hat{\Sigma}_2 - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right] .
$$

$$
\begin{aligned}
\operatorname{vec}\left[\hat{\Sigma}_1\Delta^*\hat{\Sigma}_2 - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right] &= \operatorname{vec}\left[\hat{\Sigma}_1\Delta^*\left(\hat{\Sigma}_2 - \Sigma_2\right)\right] + \operatorname{vec}\left[\hat{\Sigma}_1\Delta^*\Sigma_2\right] - \operatorname{vec}\left[\hat{\Sigma}_2 - \hat{\Sigma}_1\right] \\
&= \operatorname{vec}\left[\left(\hat{\Sigma}_1 - \Sigma_1\right)\Delta^*\left(\hat{\Sigma}_2 - \Sigma_2\right)\right] + \operatorname{vec}\left[\Sigma_1\Delta^*\left(\hat{\Sigma}_2 - \Sigma_2\right)\right] \\
&\quad + \operatorname{vec}\left[\hat{\Sigma}_1\Delta^*\Sigma_2\right] - \operatorname{vec}\left[\hat{\Sigma}_2 - \hat{\Sigma}_1\right] \\
&= \operatorname{vec}\left[\left(\hat{\Sigma}_1 - \Sigma_1\right)\Delta^*\left(\hat{\Sigma}_2 - \Sigma_2\right)\right] + \operatorname{vec}\left[\Sigma_1\Delta^*\left(\hat{\Sigma}_2 - \Sigma_2\right)\right] \\
&\quad + \operatorname{vec}\left[\Sigma_1\Delta^*\Sigma_2\right] + \operatorname{vec}\left[\left(\hat{\Sigma}_1 - \Sigma_1\right)\Delta^*\Sigma_2\right] \\
&\quad - \operatorname{vec}\left[\hat{\Sigma}_2 - \hat{\Sigma}_1\right] - \operatorname{vec}\left[\Sigma_2 - \Sigma_1\right] + \operatorname{vec}\left[\Sigma_2 - \Sigma_1\right] \\
&= \operatorname{vec}\left[\left(\hat{\Sigma}_2 - \Sigma_2\right)\Delta^*\left(\hat{\Sigma}_1 - \Sigma_1\right)\right] \\
&\quad + \operatorname{vec}\left[\Sigma_1\Delta^*\Sigma_2\right] - \operatorname{vec}\left[\Sigma_2 - \Sigma_1\right] \\
&\quad + \left(\Sigma_2^T\Delta^{*T} \otimes I_{2p} + I_{4p^2}\right)\operatorname{vec}\left[\hat{\Sigma}_1 - \Sigma_1\right] \\
&\quad + \left(I_{2p} \otimes \Sigma_1\Delta^* - I_{4p^2}\right)\operatorname{vec}\left[\hat{\Sigma}_2 - \Sigma_2\right] .
\end{aligned}
$$

Similar to the symmetric case we have that the first term is $o_p\left(\sqrt{\frac{T}{B}}\right)$, and the second term is 0. Then we define

$$
\begin{aligned}
M_{1,\text{s1Left}} &= \Sigma_2^T\Delta^{*T} \otimes I_{2p} + I_{4p^2} \\
M_{2,\text{s1Left}} &= I_{2p} \otimes \Sigma_1\Delta^* - I_{4p^2} ,
\end{aligned}
$$

and we get that

$$
\sqrt{\tfrac{T}{B}}\mathrm{v}_{\text{s1Left}}^{*T}\operatorname{vec}\left[\hat{\Sigma}_1\Delta^*\hat{\Sigma}_2 - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right] \rightsquigarrow N\left(0, \mathrm{v}_{\text{s1Left}}^{*T}\left(M_{1,\text{S1Left}}V_1M_{1,\text{S1Left}}^T + M_{2,\text{S1Left}}V_2M_{2,\text{S1Left}}^T\right)\mathrm{v}_{\text{s1Left}}^*\right) .
$$

s1Right estimating equation. Lastly, we simplify the s1Right estimating equation

$$
\operatorname{vec}\left[\hat{\Sigma}_2\Delta^*\hat{\Sigma}_1 - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right] .
$$

We have that

$$\begin{aligned}
\operatorname{vec}\left[\hat{\Sigma}_2\Delta^*\hat{\Sigma}_1 - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right] &= \operatorname{vec}\left[\hat{\Sigma}_2\Delta^*\left(\hat{\Sigma}_1 - \Sigma_1\right)\right] + \operatorname{vec}\left[\hat{\Sigma}_2\Delta^*\Sigma_1\right] - \operatorname{vec}\left[\hat{\Sigma}_2 - \hat{\Sigma}_1\right] \\
&= \operatorname{vec}\left[\left(\hat{\Sigma}_2 - \Sigma_2\right)\Delta^*\left(\hat{\Sigma}_1 - \Sigma_1\right)\right] + \operatorname{vec}\left[\Sigma_2\Delta^*\left(\hat{\Sigma}_1 - \Sigma_1\right)\right] \\
&\quad + \operatorname{vec}\left[\hat{\Sigma}_2\Delta^*\Sigma_1\right] - \operatorname{vec}\left[\hat{\Sigma}_2 - \hat{\Sigma}_1\right] \\
&= \operatorname{vec}\left[\left(\hat{\Sigma}_2 - \Sigma_2\right)\Delta^*\left(\hat{\Sigma}_1 - \Sigma_1\right)\right] + \operatorname{vec}\left[\Sigma_2\Delta^*\left(\hat{\Sigma}_1 - \Sigma_1\right)\right] \\
&\quad + \operatorname{vec}\left[\Sigma_2\Delta^*\Sigma_1\right] + \operatorname{vec}\left[\left(\hat{\Sigma}_2 - \Sigma_2\right)\Delta^*\Sigma_1\right] - \operatorname{vec}\left[\hat{\Sigma}_2 - \hat{\Sigma}_1\right] \\
&\quad - \operatorname{vec}\left[\Sigma_2 - \Sigma_1\right] + \operatorname{vec}\left[\Sigma_2 - \Sigma_1\right] \\
&= \operatorname{vec}\left[\left(\hat{\Sigma}_2 - \Sigma_2\right)\Delta^*\left(\hat{\Sigma}_1 - \Sigma_1\right)\right] \\
&\quad + \operatorname{vec}\left[\Sigma_2\Delta^*\Sigma_1\right] - \operatorname{vec}\left[\Sigma_2 - \Sigma_1\right] \\
&\quad + \left(I_{2p} \otimes \Sigma_2\Delta^* + I_{4p^2}\right)\operatorname{vec}\left[\hat{\Sigma}_1 - \Sigma_1\right] \\
&\quad + \left(\Sigma_1^T\Delta^{*T} \otimes I_{2p} - I_{4p^2}\right)\operatorname{vec}\left[\hat{\Sigma}_2 - \Sigma_2\right] .
\end{aligned}$$

Similar to the symmetric case we have that the first term is $o_p\left(\sqrt{\frac{T}{B}}\right)$, and the second term is 0. Then we define

$$\begin{aligned}
M_{1,\text{s1Right}} &= I_{2p} \otimes \Sigma_2\Delta^* + I_{4p^2} \\
M_{2,\text{s1Right}} &= \Sigma_1^T\Delta^{*T} \otimes I_{2p} - I_{4p^2} ,
\end{aligned}$$

and we get that

$$\sqrt{\tfrac{T}{B}}\mathrm{v}_{\text{s1Left}}^{*T}\operatorname{vec}\left[\hat{\Sigma}_2\Delta^*\hat{\Sigma}_1 - \left(\hat{\Sigma}_2 - \hat{\Sigma}_1\right)\right] \rightsquigarrow N\left(0, \mathrm{v}_{\text{s1Left}}^{*T}\left(M_{1,\text{s1Right}}V_1M_{1,\text{s1Right}}^T + M_{2,\text{s1Right}}V_2M_{2,\text{s1Right}}^T\right)\mathrm{v}_{\text{s1Left}}^*\right) .$$

□

Lemma 10. *Suppose the conditions of Theorem 4 are satisfied. Then*

$$\begin{aligned}\hat{\sigma}^2_{\text{symm}} &\xrightarrow{p} \sigma^2_{\text{symm}}\\ \hat{\sigma}^2_{\text{s1Left}} &\xrightarrow{p} \sigma^2_{\text{s1Left}}\\ \hat{\sigma}^2_{\text{s1Right}} &\xrightarrow{p} \sigma^2_{\text{s1Right}}\end{aligned}$$

where

$$\begin{aligned}\sigma^2_{\text{symm}} &= \mathrm{v}^{*T}_{\text{symm}} \left(M_{1,\text{symm}} V_1 M^T_{1,\text{symm}} + M_{2,\text{symm}} V_2 M^T_{2,\text{symm}}\right) \mathrm{v}^*_{\text{symm}}\\ \sigma^2_{\text{s1Left}} &= \mathrm{v}^{*T}_{\text{s1Left}} \left(M_{1,\text{s1Left}} V_1 M^T_{1,\text{s1Left}} + M_{2,\text{s1Left}} V_2 M^T_{2,\text{s1Left}}\right) \mathrm{v}^*_{\text{s1Left}}\\ \sigma^2_{\text{s1Right}} &= \mathrm{v}^{*T}_{\text{s1Right}} \left(M_{1,\text{s1Right}} V_1 M^T_{1,\text{s1Right}} + M_{2,\text{s1Right}} V_2 M^T_{2,\text{s1Right}}\right) \mathrm{v}^*_{\text{s1Right}} ,\end{aligned}$$

and

$$\begin{aligned}\hat{\sigma}^2_{\text{symm}} &= \hat{\mathrm{v}}^T_{\text{symm}} \left(\hat{M}_{1,\text{symm}} \hat{V}_1 \hat{M}^T_{1,\text{symm}} + \hat{M}_{2,\text{symm}} \hat{V}_2 \hat{M}^T_{2,\text{symm}}\right) \hat{\mathrm{v}}_{\text{symm}}\\ \hat{\sigma}^2_{\text{s1Left}} &= \hat{\mathrm{v}}^T_{\text{s1Left}} \left(\hat{M}_{1,\text{s1Left}} \hat{V}_1 \hat{M}^T_{1,\text{s1Left}} + \hat{M}_{2,\text{s1Left}} \hat{V}_2 \hat{M}^T_{2,\text{s1Left}}\right) \hat{\mathrm{v}}_{\text{s1Left}}\\ \hat{\sigma}^2_{\text{s1Right}} &= \hat{\mathrm{v}}^T_{\text{s1Right}} \left(\hat{M}_{1,\text{s1Right}} \hat{V}_1 \hat{M}^T_{1,\text{s1Right}} + \hat{M}_{2,\text{s1Right}} \hat{V}_2 \hat{M}^T_{2,\text{s1Right}}\right) \hat{\mathrm{v}}_{\text{s1Right}} .\end{aligned}$$

Proof. We will only show that $\hat{\sigma}^2_{\text{symm}}$ is consistent as the s1Left, s1Right cases follow using similar techniques. Specifically we wish to show that

$$\hat{\sigma}^2_{\text{symm}} = \hat{\mathrm{v}}^T_{\text{symm}} \left(\hat{M}_{1,\text{symm}} \hat{V}_1 \hat{M}^T_{1,\text{symm}} + \hat{M}_{2,\text{symm}} \hat{V}_2 \hat{M}^T_{2,\text{symm}}\right) \hat{\mathrm{v}}_{\text{symm}} ,$$

is a consistent estimate of

$$\sigma^2_{\text{symm}} = \mathrm{v}^{*T}_{\text{symm}} \left(M_{1,\text{symm}} V_1 M^T_{1,\text{symm}} + M_{2,\text{symm}} V_2 M^T_{2,\text{symm}}\right) \mathrm{v}^*_{\text{symm}} ,$$

where

$$M_{1,\text{symm}} = \frac{\left(\Sigma_2 \Delta^{*T} \otimes I_{2p}\right) + \left(I_{2p} \otimes \Sigma_2 \Delta^*\right)}{2} + I_{4p^2} \qquad M_{2,\text{symm}} = \frac{\left(\Sigma_1^T \Delta^{*T} \otimes I_{2p}\right) + \left(I_{2p} \otimes \Sigma_1 \Delta^*\right)}{2} - I_{4p^2} ,$$

and V_1, V_2 are the asymptotic variances of $\text{vec}\left(\hat{\Sigma}_1 - \Sigma_1\right), \text{vec}\left(\hat{\Sigma}_2 - \Sigma_2\right)$, the forms of which are given in Lemma 8. Recall that $A_l = \text{Re}(f_l)$ and $B_l = \text{Im}(f_l)$ and f_l is the spectral density in condition l. In general, $M_{l,\text{symm}}, V_l$ are functions of $\Sigma_1, \Sigma_2, \Delta^*, A_l, B_l$. Then to generate $\hat{M}_{1,\text{symm}}, \hat{M}_{2,\text{symm}}, \hat{V}_1, \hat{V}_2$ we replace the population quantites with their hatted versions. Specifically the expanded smoothed periodograms, $\hat{\Sigma}_l$, will estimate Σ_l; the SDD estimator, $\hat{\Delta}$, will estimate Δ^*; the real and imaginary parts of the smoothed periodograms ($\hat{f}_l$), $\hat{A}_l$ and $\hat{B}_l$, will estimate A_l and B_l respectively. We proceed by establishing consistency of each of the pieces of $\hat{\sigma}^2$. Once we have each of the required pieces we prove consistency of $\hat{\sigma}^2$.

We will frequently use the identity that $\left\|I \otimes \hat{X} - I \otimes X\right\|_\infty = \left\|\hat{X} - X\right\|_\infty$. This holds because the Kronecker product of X with the identity matrix is simply a block diagonal matrix with X in each block. We will also use the fact that $\|X\|_1 = \|\text{vec}\,(X)\|_1$ and $\|X\|_\infty = \|\text{vec}\,(X)\|_\infty$.

Convergence of $\hat{\mathbf{M}}_{\mathbf{l},\text{symm}}$ to $\mathbf{M}_{\mathbf{l},\text{symm}}$. We begin by noting that $\hat{M}_{1,\text{symm}} = \frac{\left(\hat{\Sigma}_2^T \hat{\Delta}^T \otimes I_{2p}\right) + \left(I_{2p} \otimes \hat{\Sigma}_2 \hat{\Delta}\right)}{2} + I_{4p^2}$, then

$$
\begin{aligned}
\left\|\hat{M}_{1,\text{symm}} - M_{1,\text{symm}}\right\|_1 &= \left\| \frac{\left(\hat{\Sigma}_2^T \hat{\Delta}^T \otimes I_{2p}\right) + \left(I_{2p} \otimes \hat{\Sigma}_2 \hat{\Delta}\right)}{2} + I_{4p^2} - \frac{\left(\Sigma_2^T \Delta^{*,T} \otimes I_{2p}\right) + \left(I_{2p} \otimes \Sigma_2 \Delta^*\right)}{2} - I_{4p^2} \right\|_1 \\
&\leq \frac{2p}{2} \left\|\hat{\Sigma}_2 \hat{\Delta} - \Sigma_2 \Delta^*\right\|_1 + \frac{2p}{2} \left\|\hat{\Sigma}_2^T \hat{\Delta}^T - \Sigma_2^T \Delta^{*T}\right\|_1 \\
&= 2p \left\|\hat{\Sigma}_2 \hat{\Delta} - \Sigma_2 \Delta^*\right\|_1 \\
&\leq 8p^3 \left\|\hat{\Sigma}_2 \hat{\Delta} - \Sigma_2 \Delta^*\right\|_\infty \\
&= 8p^3 \left\|\left(I_{2p} \otimes \hat{\Sigma}_2\right) \text{vec}\left(\hat{\Delta}\right) - \left(I_{2p} \otimes \Sigma_2\right) \text{vec}\left(\Delta^*\right)\right\|_\infty \\
&= 8p^3 \left\|\left(I_{2p} \otimes \hat{\Sigma}_2\right) \text{vec}\left(\hat{\Delta}\right) - \left(I_{2p} \otimes \Sigma_2\right) \text{vec}\left(\Delta^*\right) - \left(I_{2p} \otimes \hat{\Sigma}_2\right) \text{vec}\left(\Delta^*\right) + \left(I_{2p} \otimes \hat{\Sigma}_2\right) \text{vec}\left(\Delta^*\right)\right\|_\infty \\
&= 8p^3 \left\|\left(I_{2p} \otimes \hat{\Sigma}_2\right) \text{vec}\left(\hat{\Delta} - \Delta^*\right) + \left(\left(I_{2p} \otimes \hat{\Sigma}_2\right) - \left(I_{2p} \otimes \Sigma_2\right)\right) \text{vec}\left(\Delta^*\right)\right\|_\infty \\
&\leq 8p^3 \left\|I_{2p} \otimes \hat{\Sigma}_2\right\|_\infty \left\|\text{vec}\left(\hat{\Delta} - \Delta^*\right)\right\|_1 + 8p^3 \left\|I_{2p} \otimes \hat{\Sigma}_2 - I_{2p} \otimes \Sigma_2\right\|_\infty \left\|\text{vec}\left(\Delta^*\right)\right\|_1 \\
&\leq 8p^3 \left\|\hat{\Sigma}_2 - \Sigma_2\right\|_\infty \left\|\text{vec}\left(\hat{\Delta} - \Delta^*\right)\right\|_1 + 8p^3 \left\|\Sigma_2\right\|_\infty \left\|\text{vec}\left(\hat{\Delta} - \Delta^*\right)\right\|_1 \\
&+ 8p^3 \left\|\hat{\Sigma}_2 - \Sigma_2\right\|_\infty \left\|\text{vec}\left(\Delta^*\right)\right\|_1 ,
\end{aligned}
$$

where in the second line we used the fact that $I_{2p} \in \mathbb{R}^{2p \times 2p}$ and so $I_{2p} \otimes \hat{\Sigma}_2 \hat{\Delta}$ is block diagonal with $2p$ blocks of $\hat{\Sigma}_2 \hat{\Delta}$ so $\left\|I_{2p} \otimes \hat{\Sigma}_2 \hat{\Delta}\right\|_1 = 2p \left\|\hat{\Sigma}_2 \hat{\Delta}\right\|_1$. In the third line we use the fact that Σ_l

and Δ and their estimators are symmetric. Next we use the fact that $\|\hat{\Sigma}_2 - \Sigma_2\|_\infty = O_p(R_{T_2,p})$ since the conditions of Corollary 3 are satisfied in Theorem 4 and that $\left\|\text{vec}\left(\hat{\Delta} - \Delta^*\right)\right\|_1 = O_p(s_{\Delta^*}(R_{T_1,p} + R_{T_2,p})C_f(1 + \|\Delta^*\|_1))$ from Theorem 4. We can simplify these rates and similarly follow the same steps for condition 2 to show that

$$\left\|\hat{M}_{1,\text{symm}} - M_{1,\text{symm}}\right\|_1 = O_p\left(8p^3 \|\Sigma_2\|_\infty s_{\Delta^*}(R_{T_1,p} + R_{T_2,p})C_f(1 + \|\Delta^*\|_1) + 8p^3 \|\Delta^*\|_1 R_{T_2,p}\right)$$
$$\left\|\hat{M}_{2,\text{symm}} - M_{2,\text{symm}}\right\|_1 = O_p\left(8p^3 \|\Sigma_1\|_\infty s_{\Delta^*}(R_{T_1,p} + R_{T_2,p})C_f(1 + \|\Delta^*\|_1) + 8p^3 \|\Delta^*\|_1 R_{T_1,p}\right) .$$

Convergence of $\hat{\mathbf{V}}_\mathbf{1}$ to $\mathbf{V}_\mathbf{1}$. Next we will establish the rate of $\left\|\hat{V}_l - V_l\right\|_\infty$. Recall that we have $\hat{f}_l = \hat{A}_l + i\hat{B}_l$ and

$$\hat{V}_l = \frac{1}{2}\begin{bmatrix} P_1 \\ P_2 \end{bmatrix}\begin{bmatrix} (I_{p^2} + K)(\hat{A}_l \otimes \hat{A}_l + \hat{B}_l \otimes \hat{B}_l) & (I_{p^2} + K)(\hat{B}_l \otimes \hat{A}_l - \hat{A}_l \otimes \hat{B}_l) \\ [(I_{p^2} + K)(\hat{B}_l \otimes \hat{A}_l - \hat{A}_l \otimes \hat{B}_l)]^T & (I_{p^2} - K)(\hat{A}_l \otimes \hat{A}_l + \hat{B}_l \otimes \hat{B}_l) \end{bmatrix}\begin{bmatrix} P_1^T & P_2^T \end{bmatrix} .$$

We can remove the left and right multiplication of the $\begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$ as each row and column only has one 1. In other words, these matrices simply select one element from the matrix so their multiplication does not change the max norm. Similarly the rows and columns of $I_{p^2} + K$ have either one entry with a value of 2 or two entries each with a value of 1 so the max norm with these matrices is at most the max norm without them multiplied by 2.

Thus we have

$$\left\|\hat{V}_l - V_l\right\|_\infty = \left\|\begin{bmatrix} (\hat{A}_l \otimes \hat{A}_l + \hat{B}_l \otimes \hat{B}_l) - (A_l \otimes A_l + B_l \otimes B_l) & (\hat{B}_l \otimes \hat{A}_l - \hat{A}_l \otimes \hat{B}_l) - (B_l \otimes A_l - A_l \otimes B_l) \\ [(\hat{B}_l \otimes \hat{A}_l - \hat{A}_l \otimes \hat{B}_l) - (B_l \otimes A_l - A_l \otimes B_l)]^T & (\hat{A}_l \otimes \hat{A}_l + \hat{B}_l \otimes \hat{B}_l) - (A_l \otimes A_l + B_l \otimes B_l) \end{bmatrix}\right\|_\infty ,$$

which is equivalent to the maximum of the $(1,1)$ and $(1,2)$ entries. We will consider each of these entries separately. Consider just the (1,1) entry,

$$\left\|(\hat{A}_l \otimes \hat{A}_l + \hat{B}_l \otimes \hat{B}_l) - (A_l \otimes A_l + B_l \otimes B_l)\right\|_\infty \leq \left\|\hat{A}_l \otimes \hat{A}_l - A_l \otimes A_l\right\|_\infty + \left\|\hat{B}_l \otimes \hat{B}_l - B_l \otimes B_l\right\|_\infty .$$

Studying just the first entry on the left hand side we get

$$\begin{aligned}
\left\|\hat{A}_l \otimes \hat{A}_l - A_l \otimes A_l\right\|_\infty &= \left\|\left(\hat{A}_l - A_l\right) \otimes \hat{A}_l + A_l \otimes \left(\hat{A}_l - A_l\right)\right\|_\infty \\
&= \left\|\left(\hat{A}_l - A_l\right) \otimes \left(\hat{A}_l - A_l\right) + \left(\hat{A}_l - A_l\right) \otimes A_l + A_l \otimes \left(\hat{A}_l - A_l\right)\right\|_\infty \\
&\leq \left\|\left(\hat{A}_l - A_l\right) \otimes \left(\hat{A}_l - A_l\right)\right\|_\infty + 2\left\|\hat{A}_l - A_l\right\|_\infty \|A_l\|_\infty \\
&\leq \left\|\hat{A}_l - A_l\right\|_\infty^2 + 2\left\|\hat{A}_l - A_l\right\|_\infty \|A_l\|_\infty ,
\end{aligned}$$

where we have used the fact that $\|X \otimes Y\|_\infty \leq \|X\|_\infty \|Y\|_\infty$ since the Kronecker product multiplies every entry in X with every entry in Y. Using that $\|\hat{\Sigma}_l - \Sigma_l\|_\infty = O_p(R_{T_l,p})$ and that $\left\|\hat{A}_l - A_l\right\|_\infty, \left\|\hat{B}_l - B_l\right\|_\infty \leq \|\hat{\Sigma}_l - \Sigma_l\|_\infty$ and also that $\|A_l\|_\infty, \|B_l\|_\infty < \|\Sigma_l\|_\infty$ and repeating the same process for $\left\|\hat{B}_l \otimes \hat{B}_l - B_l \otimes B_l\right\|_\infty$ we get that

$$\begin{aligned}
\left\|\hat{A}_l \otimes \hat{A}_l - A_l \otimes A_l\right\|_\infty &= O_p\left(2\|\Sigma_l\|_\infty R_{T_l,p}\right) \\
\left\|\hat{B}_l \otimes \hat{B}_l - B_l \otimes B_l\right\|_\infty &= O_p\left(2\|\Sigma_l\|_\infty R_{T_l,p}\right) \\
\left\|(\hat{A}_l \otimes \hat{A}_l + \hat{B}_l \otimes \hat{B}_l) - (A_l \otimes A_l + B_l \otimes B_l)\right\|_\infty &= O_p\left(4\|\Sigma_l\|_\infty R_{T_l,p}\right) .
\end{aligned}$$

Next we consider the $(2,1)$ entry,

$$\left\|(\hat{B}_l \otimes \hat{A}_l - \hat{A}_l \otimes \hat{B}_l) - (B_l \otimes A_l - A_l \otimes B_l)\right\|_\infty \leq \left\|\hat{B}_l \otimes \hat{A}_l - B_l \otimes A_l\right\|_\infty + \left\|\hat{A}_l \otimes \hat{B}_l - A_l \otimes B_l\right\|_\infty .$$

Again considering just the first entry on the left hand side we get

$$\begin{aligned}
\left\|\hat{B}_l \otimes \hat{A}_l - B_l \otimes A_l\right\|_\infty &= \left\|\left(\hat{B}_l - B_l\right) \otimes \hat{A}_l + B_l \otimes \hat{A}_l - B_l \otimes A_l\right\|_\infty \\
&= \left\|\left(\hat{B}_l - B_l\right) \otimes \hat{A}_l + B_l \otimes \left(\hat{A}_l - A_l\right)\right\|_\infty \\
&= \left\|\left(\hat{B}_l - B_l\right) \otimes \left(\hat{A}_l - A_l\right) + \left(\hat{B}_l - B_l\right) \otimes A_l + B_l \otimes \left(\hat{A}_l - A_l\right)\right\|_\infty \\
&\leq \left\|\hat{B}_l - B_l\right\|_\infty \left\|\hat{A}_l - A_l\right\|_\infty + \left\|\hat{B}_l - B_l\right\|_\infty \|A_l\|_\infty + \left\|\hat{A}_l - A_l\right\|_\infty \|B_l\|_\infty \ .
\end{aligned}$$

We again use that $\|\hat{\Sigma}_l - \Sigma_l\|_\infty = O_p(R_{T_l,p})$ and that $\left\|\hat{A}_l - A_l\right\|_\infty, \left\|\hat{B}_l - B_l\right\|_\infty \leq \|\hat{\Sigma}_l - \Sigma_l\|_\infty$ and also that $\|A_l\|_\infty, \|B_l\|_\infty < \|\Sigma_l\|_\infty$ and repeat the same process for $\left\|\hat{A}_l \otimes \hat{B}_l - A_l \otimes B_l\right\|_\infty$ to see

$$\begin{aligned}
\left\|\hat{B}_l \otimes \hat{A}_l - B_l \otimes A_l\right\|_\infty &= O_p\left(2\|\Sigma_l\|_\infty R_{T_l,p}\right) \\
\left\|\hat{A}_1 \otimes \hat{B}_l - A_l \otimes B_l\right\|_\infty &= O_p\left(2\|\Sigma_l\|_\infty R_{T_l,p}\right) \\
\left\|(\hat{B}_l \otimes \hat{A}_l - \hat{A}_l \otimes \hat{B}_l) - (B_l \otimes A_l - A_l \otimes B_l)\right\|_\infty &= O_p\left(4\|\Sigma_l\|_\infty R_{T_l,p}\right) \ .
\end{aligned}$$

Using these we conclude that

$$\begin{aligned}
\left\|\hat{V}_1 - V_1\right\|_\infty &= O_p\left(4\|\Sigma_1\|_\infty R_{T_1,p}\right) \\
\left\|\hat{V}_2 - V_2\right\|_\infty &= O_p\left(4\|\Sigma_2\|_\infty R_{T_2,p}\right) \ .
\end{aligned}$$

Convergence of $\mathbf{I_{4p^2} \otimes \hat{M}_{l,\mathrm{symm}} \hat{V}_l}$ to $\mathbf{I_{4p^2} \otimes M_{l,\mathrm{symm}} V_l}$. We will find the rate of $\left\|I_{4p^2} \otimes \hat{M}_{l,\mathrm{symm}}\hat{V}_l - I_{4p^2} \otimes M_{l,\mathrm{symm}}V_l\right\|_\infty$. For ease of notation we will suppress the symm subscript in the below.

$$
\begin{aligned}
&\left\| I_{4p^2} \otimes \hat{M}_l \hat{V}_l - I_{4p^2} \otimes M_l V_l \right\|_\infty \\
&= \left\| \hat{M}_l \hat{V}_l - M_l V_l \right\|_\infty \\
&= \left\| \left(\hat{V}_l^T \otimes I_{4p^2} \right) \operatorname{vec}\left(\hat{M}_l \right) - \left(V_l^T \otimes I_{4p^2} \right) \operatorname{vec}\left(M_l \right) \right\|_\infty \\
&= \left\| \left(\hat{V}_l^T \otimes I_{4p^2} - V_l^T \otimes I_{4p^2} \right) \operatorname{vec}\left(\hat{M}_l \right) - \left(V_l^T \otimes I_{4p^2} \right) \operatorname{vec}\left(M_l - \hat{M}_l \right) \right\|_\infty \\
&= \left\| \left(\hat{V}_l^T \otimes I_{4p^2} - V_l^T \otimes I_{4p^2} \right) \operatorname{vec}\left(\hat{M}_l - M_l \right) + \left(\hat{V}_l^T \otimes I_{4p^2} - V_l^T \otimes I_{4p^2} \right) \operatorname{vec}\left(M_l \right) - \left(V_l^T \otimes I_{4p^2} \right) \operatorname{vec}\left(M_l - \hat{M}_l \right) \right\|_\infty \\
&\leq \left\| \hat{V}_l - V_l \right\|_\infty \left\| \hat{M}_l - M_l \right\|_1 + \left\| \hat{V}_l - V_l \right\|_\infty \left\| M_l \right\|_1 + \left\| V_l \right\|_\infty \left\| \hat{M}_l - M_l \right\|_1 .
\end{aligned}
$$

For $l = 1$ we have that

$$
\left\| \hat{M}_{1,\text{symm}} - M_{1,\text{symm}} \right\|_1 = O_p\left(8p^3 \left\| \Sigma_2 \right\|_\infty s_{\Delta^*}(R_{T_1,p} + R_{T_2,p}) C_f (1 + \left\| \Delta^* \right\|_1) + 8p^3 \left\| \Delta^* \right\|_1 R_{T_2,p} \right) ,
$$

is slower than $\left\| \hat{V}_1 - V_1 \right\|_\infty = O_p\left(4 \left\| \Sigma_1 \right\|_\infty R_{T_1,p} \right)$ and applying the same method for condition 2,

$$
\begin{aligned}
\left\| I_{4p^2} \otimes \hat{M}_1 \hat{V}_1 - I_{4p^2} \otimes M_1 V_1 \right\|_\infty &= \left\| \hat{M}_1 \hat{V}_1 - M_1 V_1 \right\|_\infty \\
&= O_p\left(8p^3 \left\| V_1 \right\|_\infty \left(\left\| \Sigma_2 \right\|_\infty s_{\Delta^*}(R_{T_1,p} + R_{T_2,p}) C_f (1 + \left\| \Delta^* \right\|_1) + \left\| \Delta^* \right\|_1 R_{T_2,p} \right) \right) \\
\left\| I_{4p^2} \otimes \hat{M}_2 \hat{V}_2 - I_{4p^2} \otimes M_2 V_2 \right\|_\infty &= \left\| \hat{M}_2 \hat{V}_2 - M_2 V_2 \right\|_\infty \\
&= O_p\left(8p^3 \left\| V_2 \right\|_\infty \left(\left\| \Sigma_1 \right\|_\infty s_{\Delta^*}(R_{T_1,p} + R_{T_2,p}) C_f (1 + \left\| \Delta^* \right\|_1) + \left\| \Delta^* \right\|_1 R_{T_1,p} \right) \right)
\end{aligned}
$$

Convergence of $\hat{\mathbf{M}}_\mathbf{1} \hat{\mathbf{V}}_\mathbf{1} \hat{\mathbf{M}}_\mathbf{1}^\mathbf{T} + \hat{\mathbf{M}}_\mathbf{2} \hat{\mathbf{V}}_\mathbf{2} \hat{\mathbf{M}}_\mathbf{2}^\mathbf{T}$ to $\mathbf{M}_\mathbf{1} \mathbf{V}_\mathbf{1} \mathbf{M}_\mathbf{1}^\mathbf{T} + \mathbf{M}_\mathbf{2} \mathbf{V}_\mathbf{2} \mathbf{M}_\mathbf{2}^\mathbf{T}$. For ease of notation we will again suppress the symm subscript in the below. Next we consider

$$
\left\| \hat{M}_1 \hat{V}_1 \hat{M}_1^T + \hat{M}_2 \hat{V}_2 \hat{M}_2^T - M_1 V_1 M_1^T - M_2 V_2 M_2^T \right\|_\infty \leq \left\| \hat{M}_1 \hat{V}_1 \hat{M}_1^T - M_1 V_1 M_1^T \right\|_\infty + \left\| \hat{M}_2 \hat{V}_2 \hat{M}_2^T - M_2 V_2 M_2^T \right\|_\infty
$$

Considering just the first term on the left hand side

$$
\begin{aligned}
&\left\|\hat{M}_1\hat{V}_1\hat{M}_1^T - M_1V_1M_1^T\right\|_\infty \\
&= \left\|\left(I_{4p^2}\otimes\hat{M}_1\hat{V}_1\right)\operatorname{vec}\left(\hat{M}_1^T\right) - \left(I_{4p^2}\otimes M_1V_1\right)\operatorname{vec}\left(M_1^T\right)\right\|_\infty \\
&= \left\|\left(I_{4p^2}\otimes\hat{M}_1\hat{V}_1 - I_{4p^2}\otimes M_1V_1\right)\operatorname{vec}\left(\hat{M}_1^T\right) + \left(I_{4p^2}\otimes M_1V_1\right)\operatorname{vec}\left(\hat{M}_1^T - M_1^T\right)\right\|_\infty \\
&\leq \left\|\left(I_{4p^2}\otimes\hat{M}_1\hat{V}_1 - I_{4p^2}\otimes M_1V_1\right)\operatorname{vec}\left(\hat{M}_1^T\right)\right\|_\infty + \left\|\left(I_{4p^2}\otimes M_1V_1\right)\operatorname{vec}\left(\hat{M}_1^T - M_1^T\right)\right\|_\infty \\
&\leq \left\|\hat{M}_1\hat{V}_1 - M_1V_1\right\|_\infty \left\|\operatorname{vec}\left(\hat{M}_1^T - M_1^T\right)\right\|_1 + \left\|\hat{M}_1\hat{V}_1 - M_1V_1\right\|_\infty \left\|\operatorname{vec}\left(M_1^T\right)\right\|_1 \\
&+ \left\|M_1V_1\right\|_\infty \left\|\operatorname{vec}\left(\hat{M}_1^T - M_1^T\right)\right\|_1 .
\end{aligned}
$$

Note that we have

$$
\left\|\hat{M}_1\hat{V}_1 - M_1V_1\right\|_\infty = O_p\left(8p^3\left\|V_1\right\|_\infty\left(\left\|\Sigma_2\right\|_\infty s_{\Delta^*}(R_{T_1,p}+R_{T_2,p})C_f(1+\left\|\Delta^*\right\|_1) + \left\|\Delta^*\right\|_1 R_{T_2,p}\right)\right) ,
$$

and

$$
\left\|\hat{M}_1 - M_1\right\|_1 = O_p\left(8p^3\left\|\Sigma_2\right\|_\infty s_{\Delta^*}(R_{T_1,p}+R_{T_2,p})C_f(1+\left\|\Delta^*\right\|_1) + 8p^3\left\|\Delta^*\right\|_1 R_{T_2,p}\right) .
$$

We can use the same approach for $\left\|\hat{M}_2\hat{V}_2\hat{M}_2^T - M_2V_2M_2^T\right\|_\infty$ so we conclude that

$$
\begin{aligned}
&\left\|\hat{M}_1\hat{V}_1\hat{M}_1^T - M_1V_1M_1^T\right\|_\infty \\
&\quad = O_p\left(8p^3\left(\left\|M_1V_1\right\|_\infty + \left\|M_1\right\|_1\left\|V_1\right\|_\infty\right)\left(\left\|\Sigma_2\right\|_\infty s_{\Delta^*}(R_{T_1,p}+R_{T_2,p})C_f(1+\left\|\Delta^*\right\|_1) + \left\|\Delta^*\right\|_1 R_{T_2,p}\right)\right) \\
&\quad = O_p\left(8p^3C_{v,1}\left(\left\|\Sigma_2\right\|_\infty s_{\Delta^*}(R_{T_1,p}+R_{T_2,p})C_f(1+\left\|\Delta^*\right\|_1) + \left\|\Delta^*\right\|_1 R_{T_2,p}\right)\right) \\
&\left\|\hat{M}_2\hat{V}_2\hat{M}_2^T - M_2V_2M_2^T\right\|_\infty \\
&\quad = O_p\left(8p^3\left(\left\|M_2V_2\right\|_\infty + \left\|M_2\right\|_1\left\|V_2\right\|_\infty\right)\left(\left\|\Sigma_1\right\|_\infty s_{\Delta^*}(R_{T_1,p}+R_{T_2,p})C_f(1+\left\|\Delta^*\right\|_1) + \left\|\Delta^*\right\|_1 R_{T_1,p}\right)\right) \\
&\quad = O_p\left(8p^3C_{v,2}\left(\left\|\Sigma_1\right\|_\infty s_{\Delta^*}(R_{T_1,p}+R_{T_2,p})C_f(1+\left\|\Delta^*\right\|_1) + \left\|\Delta^*\right\|_1 R_{T_1,p}\right)\right) \\
&\left\|\hat{M}_1\hat{V}_1\hat{M}_1^T + \hat{M}_2\hat{V}_2\hat{M}_2^T - M_1V_1M_1^T - M_2V_2M_2^T\right\|_\infty \\
&\quad = O_p\left(8p^3 s_{\Delta^*}(R_{T_1,p}+R_{T_2,p})C_f(1+\left\|\Delta^*\right\|_1)\left(C_{v,1}\left\|\Sigma_2\right\|_\infty + C_{v,2}\left\|\Sigma_1\right\|_\infty\right) + 8p^3\left\|\Delta^*\right\|_1\left(C_{v,1}R_{T_2,p} + C_{v,2}R_{T_1,p}\right)\right) ,
\end{aligned}
$$

where we have defined $C_{v,1} = \left\|M_1V_1\right\|_\infty + \left\|M_1\right\|_1\left\|V_1\right\|_\infty$ and $C_{v,2} = \left\|M_2V_2\right\|_\infty + \left\|M_2\right\|_1\left\|V_2\right\|_\infty$.

Convergence of plug-in variance estimator. Now we are ready to show that our plug-in estimate of the variance is consistent. The true variance is given by $\sigma^2 = \mathrm{v}^{*T} G \mathrm{v}^*$ while our plug-in estimate is given by $\hat{\sigma}^2 = \hat{\mathrm{v}}^T \hat{G} \hat{\mathrm{v}}$. Recall that $\hat{\mathrm{v}}^T$ is the Dantzig selector for the first column of $\mathbf{T}_{\mathrm{symm}}(Z, \Delta)$ and $\hat{G} = \hat{M}_{1,\mathrm{symm}} \hat{V}_1 \hat{M}_{1,\mathrm{symm}}^T + \hat{M}_{2,\mathrm{symm}} \hat{V}_2 \hat{M}_{2,\mathrm{symm}}^T$ while $G = M_{1,\mathrm{symm}} V_1 M_{1,\mathrm{symm}}^T + M_{2,\mathrm{symm}} V_2 M_{2,\mathrm{symm}}^T$. To show that $\hat{\sigma}^2$ is consistent we will show that

$$\left|\hat{\sigma}^2 - \sigma^2\right| = \left|\hat{\mathrm{v}}^T \hat{G} \hat{\mathrm{v}} - \mathrm{v}^{*T} G \mathrm{v}^*\right| = o_p(1)\,.$$

We proceed as follows

$$\begin{aligned}
&\left|\hat{\mathrm{v}}^T \hat{G} \hat{\mathrm{v}} - \mathrm{v}^{*T} G \mathrm{v}^*\right| \\
&= \left|\hat{\mathrm{v}}^T \hat{G} \hat{\mathrm{v}} - \mathrm{v}^{*T} \hat{G} \hat{\mathrm{v}} + \mathrm{v}^{*T} \hat{G} \hat{\mathrm{v}} - \mathrm{v}^{*T} G \mathrm{v}^*\right| \\
&= \left|\left(\hat{\mathrm{v}}^T - \mathrm{v}^{*T}\right) \hat{G} \hat{\mathrm{v}} + \mathrm{v}^{*T} \hat{G} \hat{\mathrm{v}} - \mathrm{v}^{*T} G \mathrm{v}^*\right| \\
&= \left|\left(\hat{\mathrm{v}}^T - \mathrm{v}^{*T}\right) \hat{G} \left(\hat{\mathrm{v}} - \mathrm{v}^*\right) + \left(\hat{\mathrm{v}}^T - \mathrm{v}^{*T}\right) \hat{G} \mathrm{v}^* + \mathrm{v}^{*T} \hat{G} \hat{\mathrm{v}} - \mathrm{v}^{*T} G \mathrm{v}^*\right| \\
&= \left|\left(\hat{\mathrm{v}}^T - \mathrm{v}^{*T}\right) \hat{G} \left(\hat{\mathrm{v}} - \mathrm{v}^*\right) + \left(\hat{\mathrm{v}}^T - \mathrm{v}^{*T}\right) \hat{G} \mathrm{v}^* + \mathrm{v}^{*T} \hat{G} \left(\hat{\mathrm{v}} - \mathrm{v}^{*T}\right) + \mathrm{v}^{*T} \left(\hat{G} - G\right) \mathrm{v}^*\right| \\
&\leq \left|\left(\hat{\mathrm{v}}^T - \mathrm{v}^{*T}\right) \hat{G} \left(\hat{\mathrm{v}} - \mathrm{v}^*\right)\right| + \left|\left(\hat{\mathrm{v}}^T - \mathrm{v}^{*T}\right) \hat{G} \mathrm{v}^*\right| + \left|\mathrm{v}^{*T} \hat{G} \left(\hat{\mathrm{v}} - \mathrm{v}^{*T}\right)\right| + \left|\mathrm{v}^{*T} \left(\hat{G} - G\right) \mathrm{v}^*\right| \\
&\leq \left\|\hat{\mathrm{v}} - \mathrm{v}^*\right\|_1 \left\|\hat{G} \left(\hat{\mathrm{v}} - \mathrm{v}^*\right)\right\|_\infty + \left\|\hat{\mathrm{v}} - \mathrm{v}^*\right\|_1 \left\|\hat{G} \mathrm{v}^*\right\|_\infty + \left\|\hat{\mathrm{v}} - \mathrm{v}^*\right\|_1 \left\|\mathrm{v}^{*T} \hat{G}\right\|_\infty + \left\|\mathrm{v}^*\right\|_1 \left\|\left(\hat{G} - G\right) \mathrm{v}^*\right\|_\infty \\
&\leq \left\|\hat{\mathrm{v}} - \mathrm{v}^*\right\|_1^2 \left\|\hat{G}\right\|_\infty + 2 \left\|\hat{\mathrm{v}} - \mathrm{v}^*\right\|_1 \left\|\left(\hat{G} - G\right) \mathrm{v}^* + G \mathrm{v}^*\right\|_\infty + \left\|\mathrm{v}^*\right\|_1^2 \left\|\hat{G} - G\right\|_\infty \\
&\leq \left\|\hat{\mathrm{v}} - \mathrm{v}^*\right\|_1^2 \left\|\hat{G} - G\right\|_\infty + \left\|\hat{\mathrm{v}} - \mathrm{v}^*\right\|_1^2 \|G\|_\infty + 2 \left\|\hat{\mathrm{v}} - \mathrm{v}^*\right\|_1 \left\|\hat{G} - G\right\|_\infty \left\|\mathrm{v}^*\right\|_1 + 2 \left\|\hat{\mathrm{v}} - \mathrm{v}^*\right\|_1 \|G \mathrm{v}^*\|_\infty \\
&+ \left\|\mathrm{v}^*\right\|_1^2 \left\|\hat{G} - G\right\|_\infty .
\end{aligned}$$

Recall that we have shown that

$$\begin{aligned}
&\left\|\hat{G} - G\right\|_\infty \\
&\quad = O_p\left(8p^3 s_{\Delta^*} (R_{T_1,p} + R_{T_2,p}) C_f (1 + \|\Delta^*\|_1) \left(C_{v,1} \|\Sigma_2\|_\infty + C_{v,2} \|\Sigma_1\|_\infty\right) + 8p^3 \|\Delta^*\|_1 \left(C_{v,1} R_{T_2,p} + C_{v,2} R_{T_1,p}\right)\right) ,
\end{aligned}$$

which in general will be slower than $\left\|\hat{\mathrm{v}} - \mathrm{v}^*\right\|_1 = O_p\left(s_{\mathrm{v}^*} \left\|\mathrm{v}^*\right\|_1 C_f \left(R_{T_1,p} + R_{T_2,p}\right)\right)$. Therefore, the last term is the slowest and we conclude that

$$
\begin{aligned}
&\left|\hat{\mathrm{v}}^T \hat{G} \hat{\mathrm{v}} - \mathrm{v}^{*T} G \mathrm{v}^*\right| \\
&\quad = O_p(8p^3 \left\|\mathrm{v}^*\right\|_1^2 s_{\Delta^*} \left(R_{T_1,p} + R_{T_2,p}\right) C_f (1 + \left\|\Delta^*\right\|_1)(C_{v,1} \left\|\Sigma_2\right\|_\infty + \\
&\qquad\qquad C_{v,2} \left\|\Sigma_1\right\|_\infty) + 8p^3 \left\|\mathrm{v}^*\right\|_1^2 \left\|\Delta^*\right\|_1 (C_{v,2} R_{T_2,p} + C_{v,1} R_{T_1,p})) \\
&\quad = o_p(1) \,.
\end{aligned}
$$

□

This concludes the intermediate results needed and we proceed in proving Theorem 6.

B.5.1 Proving Theorem 6

To prove Theorem 6, we proceed by showing the assumptions for Theorem 5 are satisfied.

Showing Assumption 5 is satisified. We begin by showing Assumption 5 which requires three concentration conditions and two expectation conditions. We must show that there exists a neighborhood $\mathcal{N}_{\theta^*}$ of θ^* such that for all $\theta \in \mathcal{N}_{\theta^*}$ the assumptions are satisfied. We will fix $|\theta - \theta^*| < \epsilon$ for some $\epsilon > 0$. We show each condition below.

To show $\lim_{T\to\infty} \mathbb{P}\left(\left\|\mathbf{t}(Z, \Delta_\theta^*) - E_{\mathbf{t}}(\Delta_\theta^*)\right\|_\infty \leq r_1(T, \theta)\right) = 1$, we define $\Gamma = 0.5\left(\Sigma_2 \otimes \Sigma_1 + \Sigma_1 \otimes \Sigma_2\right)$ and $\hat{\Gamma} = 0.5\left(\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 + \hat{\Sigma}_1 \otimes \hat{\Sigma}_2\right)$. Then we have that

$$
\begin{aligned}
\left\|\mathbf{t}(Z, \Delta_\theta^*) - E_{\mathbf{t}}(\Delta_\theta^*)\right\|_\infty &= \left\|\left(\hat{\Gamma} - \Gamma\right) \mathrm{vec}\left(\Delta_\theta^*\right) - \left(\mathrm{vec}\left(\hat{\Sigma}_2 - \Sigma_2\right) - \mathrm{vec}\left(\hat{\Sigma}_1 - \Sigma_1\right)\right)\right\|_\infty \\
&\leq \left\|\left(\hat{\Gamma} - \Gamma\right) \mathrm{vec}\left(\Delta_\theta^*\right)\right\|_\infty + \left\|\mathrm{vec}\left(\hat{\Sigma}_2 - \Sigma_2\right)\right\|_\infty + \left\|\mathrm{vec}\left(\hat{\Sigma}_1 - \Sigma_1\right)\right\|_\infty \\
&\leq \left\|\hat{\Gamma} - \Gamma\right\|_\infty \left\|\mathrm{vec}\left(\Delta_\theta^*\right)\right\|_1 + \left\|\mathrm{vec}\left(\hat{\Sigma}_2 - \Sigma_2\right)\right\|_\infty + \left\|\mathrm{vec}\left(\hat{\Sigma}_1 - \Sigma_1\right)\right\|_\infty \\
&\leq \left\|\hat{\Gamma} - \Gamma\right\|_\infty \left(\left\|\Delta^*\right\|_1 + \epsilon\right) + \left\|\mathrm{vec}\left(\hat{\Sigma}_2 - \Sigma_2\right)\right\|_\infty + \left\|\mathrm{vec}\left(\hat{\Sigma}_1 - \Sigma_1\right)\right\|_\infty \\
&= C_f O_p(R_{T_1,p} + R_{T_2,p}) \left(\left\|\Delta^*\right\|_1 + \epsilon\right) + O_p(R_{T_1,p} + R_{T_2,p}) \\
&= C_f O_p(R_{T_1,p} + R_{T_2,p}) \left(\left\|\Delta^*\right\|_1 + \epsilon\right) \,.
\end{aligned}
$$

Where we have used Corollary 3 in the second to last line. Since the conditions in Theorem 4 are satisfied, so too are the conditions of Corollary 3. Next we establish that $\lim_{T\to\infty}\mathbb{P}\left(\left|\mathrm{v}^{*T}\mathbf{t}(Z,\Delta_\theta^*)-\mathrm{v}^{*T}E_{\mathbf{t}}(\Delta_\theta^*)\right|\leq r_2(T,\theta)\right)=1$. This is easily shown by

$$\begin{aligned}\left|\mathrm{v}^{*T}\mathbf{t}(Z,\Delta_\theta^*)-\mathrm{v}^{*T}E_{\mathbf{t}}(\Delta_\theta^*)\right| &\leq \|\mathrm{v}^*\|_1\left\|\mathbf{t}(Z,\Delta_\theta^*)-E_{\mathbf{t}}(\Delta_\theta^*)\right\|_\infty \\ &= \|\mathrm{v}^*\|_1 C_f O_p(R_{T_1,p}+R_{T_2,p})\left(\|\Delta^*\|_1+\epsilon\right).\end{aligned}$$

For $\lim_{T\to\infty}\mathbb{P}\left(\sup_{\nu\in[0,1]}\left\|\hat{\mathrm{v}}^T\mathbf{T}\left(Z,\tilde{\Delta}_\nu\right)-\mathrm{v}^{*T}E_{\mathbf{T}}\left(\Delta_\theta^*\right)\right\|_\infty\leq r_3(T,\theta)\right)=1$, note that $\tilde{\Delta}_\nu=\nu\hat{\Delta}_\theta+(1-\nu)\Delta_\theta^*$ and that $\mathbf{T}\left(Z,\Delta\right)=\left(\hat{\Sigma}_2\otimes\hat{\Sigma}_1+\hat{\Sigma}_1\otimes\hat{\Sigma}_2\right)/2$ is Δ free. Also, by definition $\mathrm{v}^{*T}E_{\mathbf{T}}\left(\Delta_\theta^*\right)=\mathbf{e}_1$. So we have

$$\begin{aligned}\sup_{\nu\in[0,1]}\left\|\hat{\mathrm{v}}^T\mathbf{T}\left(Z,\tilde{\Delta}_\nu\right)-\mathrm{v}^{*T}E_{\mathbf{T}}\left(\Delta_\theta^*\right)\right\|_\infty &= \left\|\hat{\mathrm{v}}^T\hat{\Gamma}-\mathbf{e}_1\right\|_\infty \\ &\leq \rho:=\|\mathrm{v}^*\|_1 C_f O_p\left(R_{T_1,p}+R_{T_2,p}\right),\end{aligned}$$

where the last inequality holds by definition as $\hat{\mathrm{v}}$ is the Dantzig selector for the first row of $\hat{\Gamma}$ and the definition of ρ in the statement of the Theorem. Based on these rates we can conclude

$$\sup_{\theta\in\mathcal{N}_{\theta^*}}\max(r_1(T,\theta),r_2(T,\theta),r_3(T,\theta))=o(1)$$

The first expectation condition is satisfied because

$$\begin{aligned}\left\|E_{\mathbf{t}}\left(\Delta_\theta^*\right)\right\|_\infty &= \left\|\Gamma\,\mathrm{vec}\left(\Delta_\theta^*\right)-\left(\mathrm{vec}\left(\Sigma_2\right)-\mathrm{vec}\left(\Sigma_1\right)\right)-\left[\Gamma\,\mathrm{vec}\left(\Delta^*\right)-\left(\mathrm{vec}\left(\Sigma_2\right)-\mathrm{vec}\left(\Sigma_1\right)\right)\right]\right\|_\infty \\ &= \left\|\Gamma\begin{pmatrix}\theta-\theta^*\\0\\\vdots\\0\end{pmatrix}\right\|_\infty \\ &= \left\|\Gamma_{*1}\right\|_\infty\epsilon<\infty.\end{aligned}$$

In the first line we subtracted $0 = \Gamma \operatorname{vec}(\Delta^*) - (\operatorname{vec}(\Sigma_2) - \operatorname{vec}(\Sigma_1))$. Thus we have

$$\sup_{\theta \in \mathcal{N}_{\theta^*}} \left\| E_{\mathbf{t}}\left(\Delta_\theta^*\right) \right\|_\infty < \infty \,.$$

For the second condition we have that

$$\left\| \mathrm{v}^{*T} \left[E_{\mathbf{T}}\left(\Delta_\theta^*\right) \right]_{-1} \right\|_\infty = 0 \,,$$

as v^* is by definition the first column of the inverse of $E_{\mathbf{T}}\left(\Delta_\theta^*\right)$ so we conclude that

$$\sup_{\theta \in \mathcal{N}_{\theta^*}} \left\| \mathrm{v}^{*T} \left[E_{\mathbf{T}}\left(\Delta_\theta^*\right) \right]_{-1} \right\|_\infty < \infty \,.$$

Showing Assumption 6 is satisfied. Showing Assumption 6 requires ℓ_1 consistency of the parameter estimates. In this case we need to show that our SDD estimator, $\hat{\Delta}$, is ℓ_1 consistent. It also requires ℓ_1 consistency of the Dantzig selector estimate $\hat{\mathrm{v}}$.

The conditions of Theorem 4 are satisfied so we have that

$$\|\hat{\Delta} - \Delta^*\|_1 = O_p\left(s_{\Delta^*} \left(R_{T_1,p} + R_{T_2,p} \right) C_f \left(1 + \|\Delta^*\|_1 \right) \right) \,.$$

Furthermore, the conditions of Lemma 7 are satsified so we have that

$$\|\hat{\mathrm{v}} - \mathrm{v}^*\|_1 = O_p\left(s_{\mathrm{v}^*} \|\mathrm{v}^*\|_1 C_f \left(R_{T_1,p} + R_{T_2,p} \right) \right) \,.$$

Thus we conclude this assumption holds with

$$\begin{aligned} r_4(T) &= s_{\Delta^*} \left(R_{T_1,p} + R_{T_2,p} \right) C_f \left(1 + \|\Delta^*\|_1 \right) \\ r_5(T) &= s_{\mathrm{v}^*} \|\mathrm{v}^*\|_1 C_f \left(R_{T_1,p} + R_{T_2,p} \right) \,. \end{aligned}$$

Technical conditions from Theorem 5. Here we show the two technical conditions from Theorem 5. They are $\theta \mapsto \hat{S}(\hat{\Delta}_\theta)$ is continuous with a single root $\tilde{\theta}$ or is non-decreasing. We also need

to show that for any $\epsilon > 0$,

$$\mathrm{v}^{*T}\left[E_{\mathbf{t}}\left(\Delta^*_{\theta^*-\epsilon}\right)\right]\mathrm{v}^{*T}\left[E_{\mathbf{t}}\left(\Delta^*_{\theta^*+\epsilon}\right)\right],$$

or that θ^* is the unique root of $\mathrm{v}^{*T}E_{\mathbf{t}}\left(\Delta^*_\theta\right)$. The map $\theta \mapsto \hat{S}(\hat{\Delta}_\theta)$ is continuous as it is linear and has a unique root except when $\hat{\mathrm{v}}^T\left[\hat{\Gamma}\right]_{*1} = 0$ but by the definition of the Dantzig selector $\left|\hat{\mathrm{v}}^T\left[\hat{\Gamma}\right]_{*1} - 1\right| \leq \rho$ so for appropriate ρ the unique root will exist. To show the second condition, write

$$\begin{aligned}
\mathrm{v}^{*T}E_{\mathbf{t}}\left(\Delta^*_\theta\right) &= \mathrm{v}^{*T}\left[\Gamma\,\mathrm{vec}\left(\Delta^*_\theta\right) - \mathrm{vec}\left(\Sigma_2 - \Sigma_1\right)\right] \\
&= \mathrm{v}^{*T}\left[\Gamma\,\mathrm{vec}\left(\Delta^*_\theta\right) - \mathrm{vec}\left(\Sigma_2 - \Sigma_1\right)\right] + \mathrm{v}^{*T}\left[\Gamma\,\mathrm{vec}\left(\Delta^*\right) - \mathrm{vec}\left(\Sigma_2 - \Sigma_1\right)\right] \\
&= \mathrm{v}^{*T}\Gamma\left(\mathrm{vec}\left(\Delta^*_\theta\right) - \mathrm{vec}\left(\Delta^*\right)\right) \\
&= \theta - \theta^*,
\end{aligned}$$

where in the second line we added 0 since $\mathrm{v}^{*T}\left[\Gamma\,\mathrm{vec}\left(\Delta^*\right) - \mathrm{vec}\left(\Sigma_2 - \Sigma_1\right)\right] = 0$. In the last line we use the fact that by definition $\mathrm{v}^{*T}\Gamma = \begin{bmatrix}1 & 0 & \dots & 0\end{bmatrix}$ and $\mathrm{vec}\left(\Delta^*_\theta\right) - \mathrm{vec}\left(\Delta^*\right) = \begin{bmatrix}\theta - \theta^* & 0 & \dots & 0\end{bmatrix}^T$. Solving for $\mathrm{v}^{*T}E_{\mathbf{t}}\left(\Delta^*_\theta\right) = 0$ shows that $\theta = \theta^*$ is the unique root.

Showing Assumption 7 is satisified. Assumption 7 requires

$$\sigma^{-1}s_T S\left(\beta^*\right) \rightsquigarrow N(0,1),$$

where $\sigma^2 = \mathrm{v}^{*T}G\mathrm{v}^*$, $G = \lim_{T\to\infty} s_T^2\,\mathrm{Cov}(\mathbf{t}(Z,\beta^*))$, $\mathrm{v}^* := \left[E_{\mathbf{T}}(\beta^*)\right]^{-1}$, $E_{\mathbf{T}}(\beta^*) = \lim_{T\to\infty} E\left(\frac{\partial}{\partial\beta}\mathbf{t}(Z,\beta)\Big|_{\beta^*}\right)$. By Lemma 9 this is satisfied for each of the three estimating equation types.

Showing Assumption 8 is satisfied. Here we show Assumption 8. Assumption 8 is a bound on $\mathrm{v}^T\frac{\partial}{\partial\theta}\left[\mathbf{T}(Z,(\theta,\gamma^T)^T)\right]_{*1}$. Recall for our estimator, we have

$$\mathbf{T}(Z,\Delta) = \frac{\partial}{\partial\Delta}t(Z;\Delta) = (\hat{\Sigma}_2\otimes\hat{\Sigma}_1 + \hat{\Sigma}_1\otimes\hat{\Sigma}_2)/2,$$

which is Δ free so

$$\frac{\partial}{\partial\theta}\mathbf{T}(Z,\Delta) = 0\,,$$

and Assumption 8 holds.

Showing Assumption 9 is satisfied. Assumption 9 says that the convergence rates in Assumptions 5 and 6 satisfy

$$s_T(r_4(T)r_3(T,\theta^*) + r_5(T)r_1(T,\theta^*)) = o(1)\,.$$

Below we restate the rates we have from above.

$$\begin{aligned}
r_1(T,\theta^*) &= (R_{T_1,p} + R_{T_2,p})\, C_f\, (\|\Delta^*\|_1 + \epsilon) \\
r_3(T,\theta^*) &= \|\mathrm{v}^*\|_1\, C_f\, (R_{T_1,p} + R_{T_2,p}) \\
r_4(T) &= s_{\Delta^*}\, (R_{T_1,p} + R_{T_2,p})\, C_f\, (1 + \|\Delta^*\|_1) \\
r_5(T) &= s_{\mathrm{v}^*}\, \|\mathrm{v}^*\|_1\, C_f\, (R_{T_1,p} + R_{T_2,p})\ .
\end{aligned}$$

Then if we assume that the sample sizes T_1, T_2 are equal to T for both samples and the smoothing spans B_1, B_2 are equal to B, we conclude that

$$\begin{aligned}
r_1(T,\theta^*)r_5(T) &= 4R_{T,p}^2 C_f^2\, (\|\Delta^*\|_1 + \epsilon)\, s_{\mathrm{v}^*}\, \|\mathrm{v}^*\|_1 \\
r_3(T,\theta^*)r_4(T) &= 4R_{T,p}^2\, \|\mathrm{v}^*\|_1\, C_f^2 s_{\Delta^*}\, (1 + \|\Delta^*\|_1)\ .
\end{aligned}$$

Using the same method as we did to show that the second order terms in the asymptotic distribution of the symmetric estimating equation were $o_p\left(\sqrt{\frac{B}{T}}\right)$ in the proof of Lemma 9 we can easily show that $s_T(r_4(T)r_3(T,\theta^*) + r_5(T)r_1(T,\theta^*)) = o(1)$ for $s_T = \sqrt{\frac{T}{B}}$.

Consistent estimate of σ^2. Lastly, by Lemma 10, our plug-in varaince estimator $\hat{\sigma}^2$ is consistent for σ^2.

We have shown all conditions for Theorem 5 which concludes the proof.

B.6 Additional Simulation Details and Results

For simulation setting 1, the transition matrix is the same as in (Sun et al., 2018). Specifically, it consists of 18 blocks of dimension 3×3, where each block is $\begin{bmatrix} 0.5 & 0.9 & 0 \\ 0 & 0.5 & 0.9 \\ 0 & 0 & 0.5 \end{bmatrix}$. In both simulation setting 2 and setting 3, 60% of the coefficients in the 3×3 block were randomly drawn from either a $\mathrm{Uniform}(-0.5, -0.2)$ or a $\mathrm{Uniform}(0.2, 0.5)$ each with equal probability. In the second and third setting, 40% and 5% respectively of the entries of the larger block were randomly drawn from a $\mathrm{Uniform}(-0.5, -0.2)$ and $\mathrm{Uniform}(0.2, 0.5)$ each with equal probability. This corresponded to 60% and 95% sparsity respectively. The number of non-zero entries in the difference in expanded inverse spectral densities, which we will refer to as edges, varies by frequency but is almost always 22 for Sim-Sun, 14 for Sim-Dense, and 28 for Sim-Sparse.

Let TP, FN, TN, FP denote the true positive, false negative, true negative, and false positive edges identified by either SDD or the naïve method, respectively. A value greater than 1×10^{-6} in absolute value was considered an edge. The metrics are defined as follows

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \qquad \mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{4p^2} \qquad \mathrm{RRMSE} = \sqrt{\frac{\sum_{i,j}(\hat{\Delta}_{i,j} - \Delta_{i,j})^2}{\sum_{i,j}(\Delta_{i,j})^2}},$$

where $\hat{\Delta}$ represents the difference estimator, either SDD or the naïve difference, Δ is the true difference. The denominator of the accuracy measure is $4p^2$ as expanding a $p \times p$ spectral density to the real space gives a $2p \times 2p$ matrix which has $4p^2$ entries. We also report the number of average number of true edges and the average number of estimated edges across frequencies.

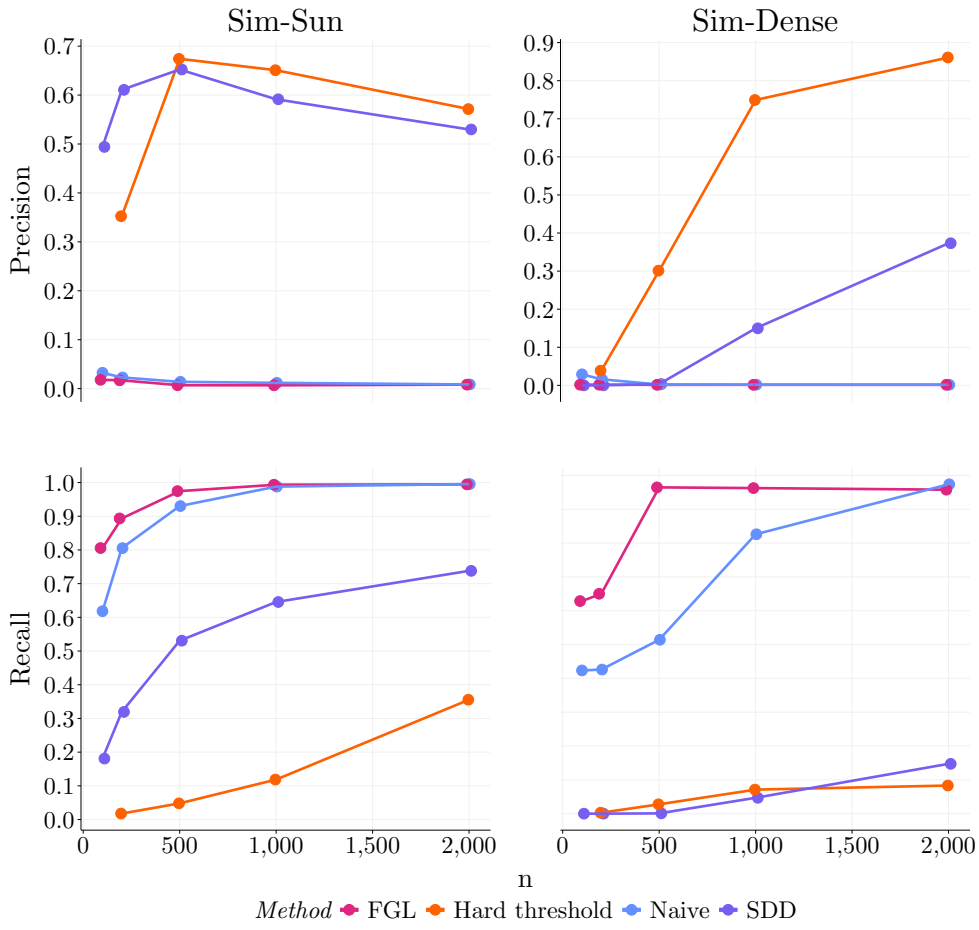


Figure B.1: Sim-Sun (left) and Sim-Dense (right) Precision and Recall. Results are reported as mean (dots) and SE (vertical lines) where the mean and SE are taken across all frequencies and iterations for a given sample size T. Note that SE may appear as 0 due to the large number of frequencies and iterations.

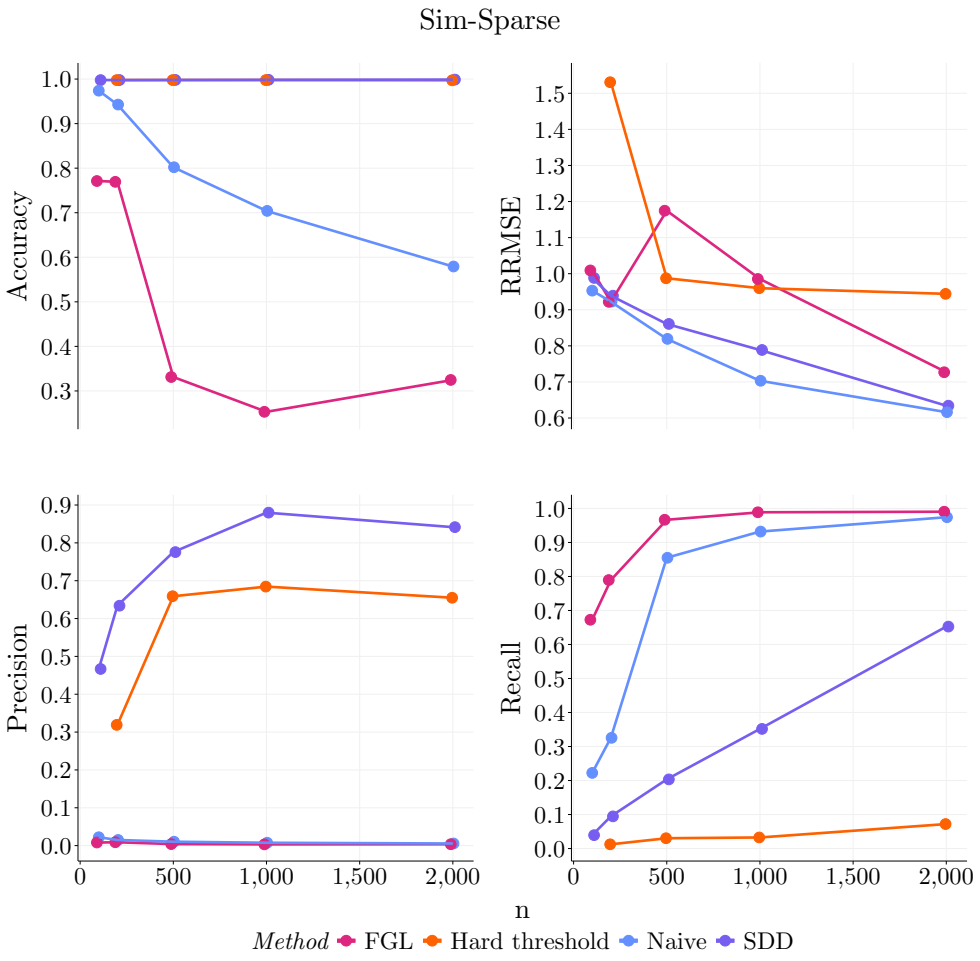


Figure B.2: Sim-Sparse. Results are reported as mean (dots) and SE (vertical lines) where the mean and SE are taken across all frequencies and iterations for a given sample size T. Note that SE may appear as 0 due to the large number of frequencies and iterations.

Table B.1: Sim-Sun. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies and iterations for a given sample size T. SEs are rounded to two decimal places so a SE of 0.00 indicates SE < 0.005.

		SDD				
T	# True edges	# Est edges	Precision	Recall	Accuracy	RRMSE
100	21.6 (0.04)	8.28 (0.13)	0.49 (0.01)	0.18 (0.00)	1.00 (0.00)	0.95 (0.00)
200	21.8 (0.02)	12.9 (0.12)	0.61 (0.01)	0.32 (0.00)	1.00 (0.00)	0.89 (0.00)
500	21.9 (0.01)	23.7 (0.23)	0.65 (0.00)	0.53 (0.00)	1.00 (0.00)	0.73 (0.00)
1000	21.9 (0.01)	35.5 (0.38)	0.59 (0.00)	0.65 (0.00)	1.00 (0.00)	0.60 (0.00)
2000	21.9 (0.01)	58.8 (0.91)	0.53 (0.00)	0.74 (0.00)	1.00 (0.00)	0.49 (0.00)
		Naïve				
		559 (4.89)	0.03 (0.00)	0.62 (0.00)	0.95 (0.00)	0.95 (0.00)
		983 (5.60)	0.02 (0.00)	0.81 (0.00)	0.92 (0.00)	0.90 (0.00)
		1779 (10.5)	0.01 (0.00)	0.93 (0.00)	0.85 (0.00)	0.78 (0.00)
		2306 (15.2)	0.01 (0.00)	0.99 (0.00)	0.80 (0.00)	0.69 (0.00)
		3144 (21.0)	0.01 (0.00)	1.00 (0.00)	0.73 (0.00)	0.57 (0.00)
		FGL				
		2015 (26.4)	0.02 (0.00)	0.81 (0.00)	0.83 (0.00)	0.91 (0.00)
		2524 (24.3)	0.02 (0.00)	0.89 (0.00)	0.79 (0.00)	0.83 (0.00)
		5104 (42.9)	0.01 (0.00)	0.97 (0.00)	0.56 (0.00)	0.68 (0.00)
		5930 (45.4)	0.01 (0.00)	0.99 (0.00)	0.49 (0.00)	0.64 (0.00)
		6329 (45.9)	0.01 (0.00)	0.99 (0.00)	0.46 (0.00)	0.60 (0.00)
		Hard threshold				
		-	-	-	-	-
		0.68 (0.01)	0.35 (0.01)	0.02 (0.00)	1.00 (0.00)	1.50 (0.01)
		1.52 (0.10)	0.67 (0.01)	0.05 (0.00)	1.00 (0.00)	1.00 (0.00)
		18.2 (2.22)	0.65 (0.01)	0.12 (0.00)	1.00 (0.00)	0.95 (0.00)
		249 (7.91)	0.57 (0.01)	0.36 (0.01)	0.98 (0.00)	1.06 (0.01)

Table B.2: Sim-Dense. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies and iterations for a given sample size T. SEs are rounded to two decimal places so a SE of 0.00 indicates SE < 0.005.

		SDD				
T	# True edges	# Est edges	Precision	Recall	Accuracy	RRMSE
100	13.7 (0.03)	0.76 (0.03)	0.00 (0.00)	0.00 (0.00)	1.00 (0.00)	1.00 (0.00)
200	13.9 (0.02)	0.76 (0.02)	0.00 (0.00)	0.00 (0.00)	1.00 (0.00)	1.00 (0.00)
500	14.0 (0.01)	1.64 (0.05)	0.00 (0.00)	0.00 (0.00)	1.00 (0.00)	1.00 (0.00)
1000	14.0 (0.01)	3.37 (0.07)	0.15 (0.00)	0.05 (0.00)	1.00 (0.00)	0.99 (0.00)
2000	14.0 (0.01)	6.52 (0.10)	0.37 (0.01)	0.15 (0.00)	1.00 (0.00)	0.93 (0.00)
		Naïve				
		244 (2.27)	0.03 (0.00)	0.42 (0.00)	0.98 (0.00)	1.00 (0.00)
		562 (5.08)	0.02 (0.00)	0.43 (0.00)	0.95 (0.00)	0.99 (0.00)
		3466 (7.23)	0.00 (0.00)	0.51 (0.00)	0.70 (0.00)	1.20 (0.00)
		6522 (9.08)	0.00 (0.00)	0.83 (0.00)	0.44 (0.00)	1.57 (0.01)
		9111 (13.9)	0.00 (0.00)	0.97 (0.00)	0.22 (0.00)	2.24 (0.01)
		FGL				
		5296 (31.3)	0.00 (0.00)	0.63 (0.00)	0.55 (0.00)	2.41 (0.02)
		5338 (23.0)	0.00 (0.00)	0.65 (0.00)	0.54 (0.00)	2.08 (0.01)
		10055 (19.1)	0.00 (0.00)	0.96 (0.00)	0.14 (0.00)	4.11 (0.01)
		9661 (22.5)	0.00 (0.00)	0.96 (0.00)	0.17 (0.00)	3.04 (0.01)
		9234 (24.6)	0.00 (0.00)	0.96 (0.00)	0.21 (0.00)	2.30 (0.01)
		Hard threshold				
		-	-	-	-	-
		0.70 (0.01)	0.04 (0.00)	0.00 (0.00)	1.00 (0.00)	3.04 (0.02)
		0.83 (0.01)	0.30 (0.01)	0.03 (0.00)	1.00 (0.00)	1.13 (0.00)
		1.08 (0.01)	0.75 (0.01)	0.07 (0.00)	1.00 (0.00)	0.93 (0.00)
		1.17 (0.01)	0.86 (0.00)	0.08 (0.00)	1.00 (0.00)	0.89 (0.00)

Table B.3: Sim-Sparse. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies and iterations for a given sample size T. SEs are rounded to two decimal places so a SE of 0.00 indicates SE < 0.005.

		SDD				
T	# True edges	# Est edges	Precision	Recall	Accuracy	RRMSE
100	27.4 (0.06)	1.66 (0.03)	0.47 (0.01)	0.04 (0.00)	1.00 (0.00)	0.99 (0.00)
200	27.7 (0.03)	3.22 (0.04)	0.63 (0.01)	0.09 (0.00)	1.00 (0.00)	0.94 (0.00)
500	27.9 (0.02)	6.71 (0.07)	0.78 (0.00)	0.20 (0.00)	1.00 (0.00)	0.86 (0.00)
1000	27.9 (0.02)	11.4 (0.11)	0.88 (0.00)	0.35 (0.00)	1.00 (0.00)	0.79 (0.00)
2000	27.9 (0.02)	22.5 (0.13)	0.84 (0.00)	0.65 (0.00)	1.00 (0.00)	0.63 (0.00)
		Naïve				
		289 (1.63)	0.02 (0.00)	0.22 (0.00)	0.97 (0.00)	0.95 (0.00)
		658 (3.20)	0.01 (0.00)	0.33 (0.00)	0.94 (0.00)	0.92 (0.00)
		2328 (4.67)	0.01 (0.00)	0.86 (0.00)	0.80 (0.00)	0.82 (0.00)
		3475 (6.41)	0.01 (0.00)	0.93 (0.00)	0.70 (0.00)	0.70 (0.00)
		4933 (8.46)	0.01 (0.00)	0.97 (0.00)	0.58 (0.00)	0.62 (0.00)
		FGL				
		2679 (23.2)	0.01 (0.00)	0.67 (0.00)	0.77 (0.00)	1.01 (0.00)
		2707 (14.4)	0.01 (0.00)	0.79 (0.00)	0.77 (0.00)	0.92 (0.00)
		7826 (36.1)	0.00 (0.00)	0.97 (0.00)	0.33 (0.00)	1.17 (0.00)
		8738 (19.6)	0.00 (0.00)	0.99 (0.00)	0.25 (0.00)	0.99 (0.00)
		7904 (23.1)	0.00 (0.00)	0.99 (0.00)	0.32 (0.00)	0.73 (0.00)
		Hard threshold				
		-	-	-	-	-
		0.70 (0.01)	0.32 (0.01)	0.01 (0.00)	1.00 (0.00)	1.53 (0.01)
		0.86 (0.01)	0.66 (0.01)	0.03 (0.00)	1.00 (0.00)	0.99 (0.00)
		0.90 (0.01)	0.68 (0.01)	0.03 (0.00)	1.00 (0.00)	0.96 (0.00)
		8.41 (0.49)	0.65 (0.01)	0.07 (0.00)	1.00 (0.00)	0.94 (0.00)

Table B.4: Sim-Sun. Difference Between SDD and Naïve Method for Each Metric. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies and iterations for a given sample size T. SEs are rounded to two decimal places so a SE of 0.00 indicates SE < 0.005.

T	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-550 (4.88)	0.46 (0.01)	-0.44 (0.00)	0.05 (0.00)	0.00 (0.00)
200	-970 (5.61)	0.59 (0.01)	-0.49 (0.00)	0.08 (0.00)	-0.01 (0.00)
500	-1756 (10.4)	0.64 (0.00)	-0.40 (0.00)	0.15 (0.00)	-0.05 (0.00)
1000	-2271 (15.1)	0.58 (0.00)	-0.34 (0.00)	0.19 (0.00)	-0.08 (0.00)
2000	-3086 (20.6)	0.52 (0.00)	-0.26 (0.00)	0.26 (0.00)	-0.08 (0.00)

Table B.5: Sim-Sun. Difference Between SDD and Hard Thresholding Method for Each Metric. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies and iterations for a given sample size T. SEs are rounded to two decimal places so a SE of 0.00 indicates SE < 0.005.

T	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-	-	-	-	-
200	12.2 (0.12)	0.26 (0.01)	0.30 (0.00)	0.00 (0.00)	-0.61 (0.01)
500	22.2 (0.24)	-0.02 (0.01)	0.48 (0.00)	-0.00 (0.00)	-0.28 (0.00)
1000	17.3 (2.23)	-0.06 (0.01)	0.53 (0.01)	0.00 (0.00)	-0.35 (0.01)
2000	-190 (7.93)	-0.04 (0.01)	0.38 (0.01)	0.02 (0.00)	-0.57 (0.01)

Table B.6: Sim-Sun. Difference Between SDD and FGL for Each Metric. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies and iterations for a given sample size T. SEs are rounded to two decimal places so a SE of 0.00 indicates SE < 0.005.

T	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-2007 (26.4)	0.48 (0.01)	-0.62 (0.00)	0.17 (0.00)	0.05 (0.00)
200	-2511 (24.3)	0.59 (0.01)	-0.57 (0.00)	0.21 (0.00)	0.06 (0.00)
500	-5081 (42.9)	0.65 (0.00)	-0.44 (0.00)	0.43 (0.00)	0.05 (0.00)
1000	-5894 (45.5)	0.58 (0.00)	-0.35 (0.00)	0.50 (0.00)	-0.04 (0.00)
2000	-6270 (46.2)	0.52 (0.00)	-0.26 (0.00)	0.54 (0.00)	-0.11 (0.00)

Table B.7: Sim-Dense. Difference Between SDD and Naïve Method for Each Metric. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies and iterations for a given sample size T. SEs are rounded to two decimal places so a SE of 0.00 indicates SE < 0.005.

T	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-243 (2.26)	-0.03 (0.00)	-0.42 (0.00)	0.02 (0.00)	-0.00 (0.00)
200	-561 (5.08)	-0.02 (0.00)	-0.43 (0.00)	0.05 (0.00)	0.01 (0.00)
500	-3464 (7.22)	0.00 (0.00)	-0.51 (0.00)	0.30 (0.00)	-0.20 (0.00)
1000	-6518 (9.09)	0.15 (0.00)	-0.78 (0.00)	0.56 (0.00)	-0.58 (0.01)
2000	-9105 (14.0)	0.37 (0.01)	-0.83 (0.00)	0.78 (0.00)	-1.31 (0.01)

Table B.8: Sim-Dense. Difference Between SDD and Hard Thresholding Method for Each Metric. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies and iterations for a given sample size T. SEs are rounded to two decimal places so a SE of 0.00 indicates SE < 0.005.

T	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-	-	-	-	-
200	0.05 (0.02)	-0.04 (0.00)	-0.00 (0.00)	-0.00 (0.00)	-2.04 (0.02)
500	0.82 (0.05)	-0.30 (0.01)	-0.03 (0.00)	-0.00 (0.00)	-0.13 (0.00)
1000	2.29 (0.07)	-0.60 (0.01)	-0.02 (0.00)	-0.00 (0.00)	0.06 (0.00)
2000	5.35 (0.11)	-0.49 (0.01)	0.06 (0.00)	-0.00 (0.00)	0.04 (0.00)

Table B.9: Sim-Dense. Difference Between SDD and FGL for Each Metric. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies and iterations for a given sample size T. SEs are rounded to two decimal places so a SE of 0.00 indicates SE < 0.005.

T	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-5295 (31.3)	-0.00 (0.00)	-0.63 (0.00)	0.45 (0.00)	-1.41 (0.02)
200	-5338 (23.0)	-0.00 (0.00)	-0.65 (0.00)	0.46 (0.00)	-1.08 (0.01)
500	-10053 (19.2)	0.00 (0.00)	-0.96 (0.00)	0.86 (0.00)	-3.11 (0.01)
1000	-9658 (22.5)	0.15 (0.00)	-0.92 (0.00)	0.83 (0.00)	-2.05 (0.01)
2000	-9228 (24.7)	0.37 (0.01)	-0.81 (0.00)	0.79 (0.00)	-1.37 (0.01)

Table B.10: Sim-Sparse. Difference Between SDD and Naïve Method for Each Metric. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies and iterations for a given sample size T. SEs are rounded to two decimal places so a SE of 0.00 indicates SE < 0.005.

n	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-287 (1.63)	0.44 (0.01)	-0.18 (0.00)	0.02 (0.00)	0.04 (0.00)
200	-655 (3.20)	0.62 (0.01)	-0.23 (0.00)	0.06 (0.00)	0.02 (0.00)
500	-2321 (4.68)	0.77 (0.00)	-0.65 (0.00)	0.20 (0.00)	0.04 (0.00)
1000	-3464 (6.41)	0.87 (0.00)	-0.58 (0.00)	0.29 (0.00)	0.09 (0.00)
2000	-4911 (8.45)	0.84 (0.00)	-0.32 (0.00)	0.42 (0.00)	0.02 (0.00)

Table B.11: Sim-Sparse. Difference Between SDD and Hard Thresholding Method for Each Metric. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies and iterations for a given sample size T. SEs are rounded to two decimal places so a SE of 0.00 indicates SE < 0.005.

n	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-	-	-	-	-
200	2.52 (0.04)	0.31 (0.01)	0.08 (0.00)	0.00 (0.00)	-0.59 (0.01)
500	5.85 (0.07)	0.12 (0.01)	0.17 (0.00)	0.00 (0.00)	-0.13 (0.00)
1000	10.5 (0.11)	0.20 (0.01)	0.32 (0.00)	0.00 (0.00)	-0.17 (0.00)
2000	14.1 (0.49)	0.19 (0.01)	0.58 (0.00)	0.00 (0.00)	-0.31 (0.00)

Table B.12: Sim-Sparse. Difference Between SDD and FGL for Each Metric. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies and iterations for a given sample size T. SEs are rounded to two decimal places so a SE of 0.00 indicates SE < 0.005.

n	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-2677 (23.2)	0.46 (0.01)	-0.63 (0.00)	0.23 (0.00)	-0.02 (0.00)
200	-2704 (14.4)	0.63 (0.01)	-0.69 (0.00)	0.23 (0.00)	0.02 (0.00)
500	-7819 (36.1)	0.77 (0.00)	-0.76 (0.00)	0.67 (0.00)	-0.31 (0.00)
1000	-8727 (19.6)	0.88 (0.00)	-0.64 (0.00)	0.75 (0.00)	-0.20 (0.00)
2000	-7881 (23.1)	0.84 (0.00)	-0.34 (0.00)	0.67 (0.00)	-0.09 (0.00)

Appendix C

APPENDICES FOR CHAPTER 4

C.1 Additional simulation information

In this section we provide more detail on the simulation settings. We use $Z_t \in \mathbb{R}^p$ to denote the data observed at time $t \in \{1/T, \ldots, 1\}$. Recall that parameters such as change-point locations and transition matrices are fixed across simulation seeds.

C.1.1 $\boldsymbol{NLAR_3(4)}$

The $\mathrm{NLAR}_3(4)$ model with 2 change-points and 3 regimes is given by

$$\mathrm{NLAR}_3(4) : Z_t = \begin{cases} (0.4 - \exp\left(-50Z_{t-1/T}^2\right))Z_{t-3/T} + (0.5 - 0.1\exp\left(-50Z_{t-2/T}^2\right))Z_{t-4/T} + \epsilon_t & \text{if} \quad \tau_0 < t \leq \tau_1 \\ (0.4 - \exp\left(-50Z_{t-2/T}^2\right))Z_{t-2/T} + (0.5 - 0.1\exp\left(-50Z_{t-4/T}^2\right))Z_{t-1/T} + \epsilon_t & \text{if} \quad \tau_1 < t \leq \tau_2 \\ (0.4 - \exp\left(-50Z_{t-3/T}^2\right))Z_{t-2/T} + (0.5 - 0.1\exp\left(-50Z_{t-3/T}^2\right))Z_{t-4/T} + \epsilon_t & \text{if} \quad \tau_2 < t \leq \tau_3 \end{cases} . \tag{C.1}$$

Note that multiplication, exponentiation, and raising to a power are carried out element-wise. That is $Z_{t-1/T}Z_{t-2/T} = \begin{bmatrix} Z_{t-1/T,1}Z_{t-2/T,1} & \ldots & Z_{t-1/T,p}Z_{t-2/T,p} \end{bmatrix}$. A realization of this process for one data seed is shown in Figure C.1

C.1.2 $\boldsymbol{VAR_3(2)}$

The $\mathrm{VAR}_3(2)$ model with 3 change-points and thus 4 regimes can be written as

$$\mathrm{VAR}_3(2) : Z_t = \sum_{j=1}^{4} \left(\sum_{l=1/T}^{2/T} A_{j,l} Z_{t-l} \right) I\left(\tau_{j-1} < t \leq \tau_j\right) + \epsilon_t \,. \tag{C.2}$$

Note that in this setting the outer summation over $j = 1, \ldots, 4$ indexes regime while the inner summation indexes the lags of the VAR processes. We use $l = 1/T,\ l = 2/T$ since we using the

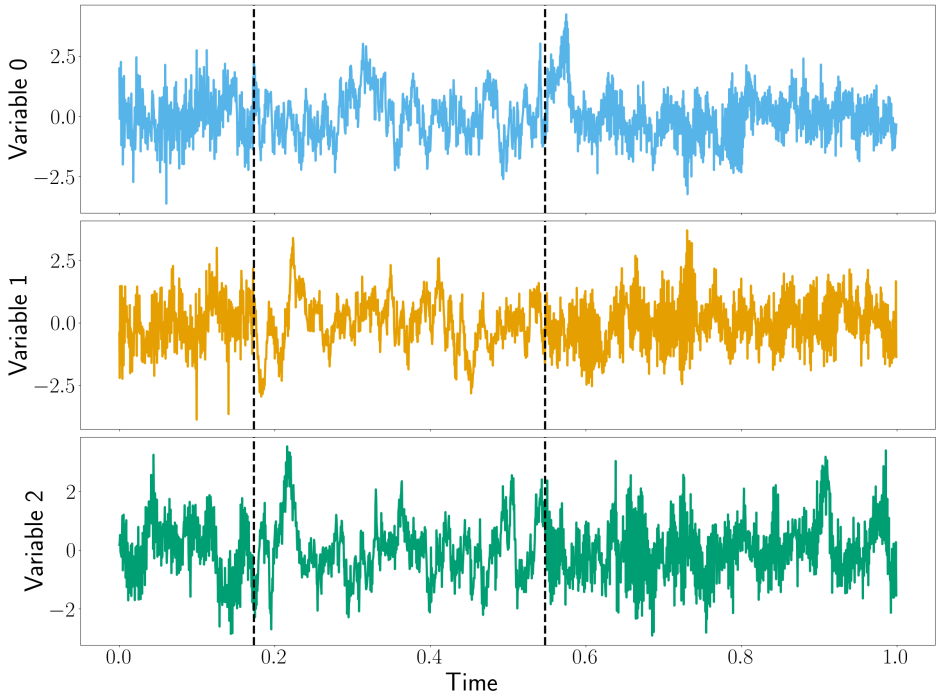


Figure C.1: Realization of $\mathrm{NLAR}_3(4)$ for one data seed. Dashed lines indicate true change-points.

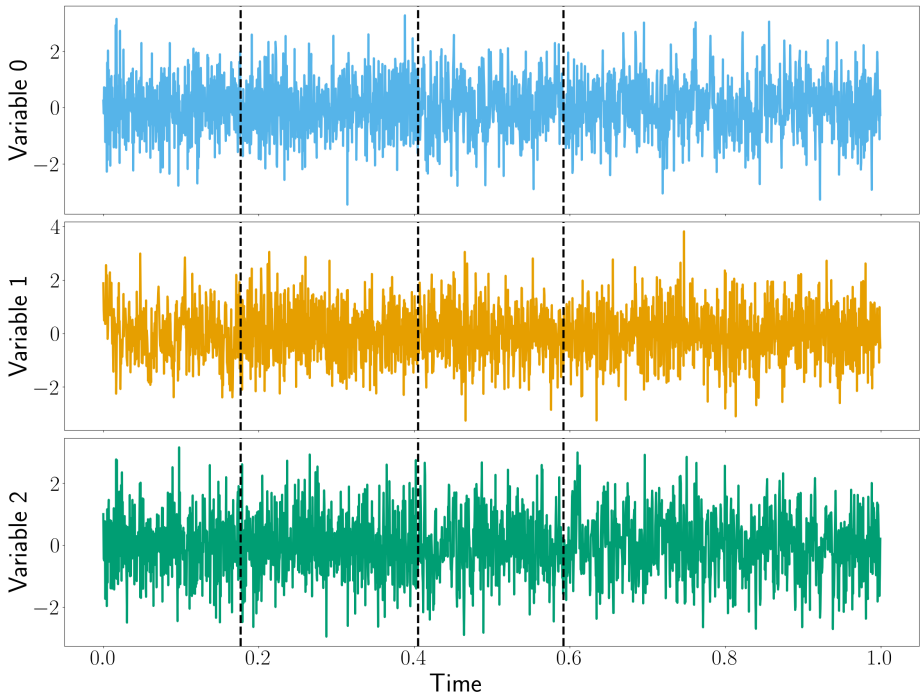


Figure C.2: Realization of $\mathrm{VAR}_3(2)$ for one data seed. Dashed lines indicate true change-points.

time indexing from 0 to 1. Thus, $l = 1/T,\ l = 2/T$ selects the lag-1 and lag-2 data. The matrices $A_{j,l}$ are simulated as in Section 4.3.1. Recall that $\tau_0 = -\infty, \tau_4 = \infty$. A realization of this process for one data seed is shown in Figure C.2

C.1.3 ***NLARU$_2$(4)***

The $\mathrm{NLARU}_2(4)$ model with 1 change-points and 2 regimes is given by

$$\mathrm{NLARU}_2(4) : Z_t = \begin{cases} 0.6(3 - (Z_{t-1/T}Z_{t-2/T} - 0.5)^3)/(1 + (Z_{t-1/T}Z_{t-2/T} - 0.5)^4) & \text{if} \quad \tau_0 < t \leq \tau_1 \\ 0.6(3 - (Z_{t-3/T}Z_{t-4/T} - 0.5)^3)/(1 + (Z_{t-3/T}Z_{t-4/T} - 0.5)^4) & \text{if} \quad \tau_1 < t \leq \tau_2 \end{cases}. \tag{C.3}$$

A realization of this process for one data seed is shown in Figure C.3

C.1.4 Simulation results with misspecified lag order

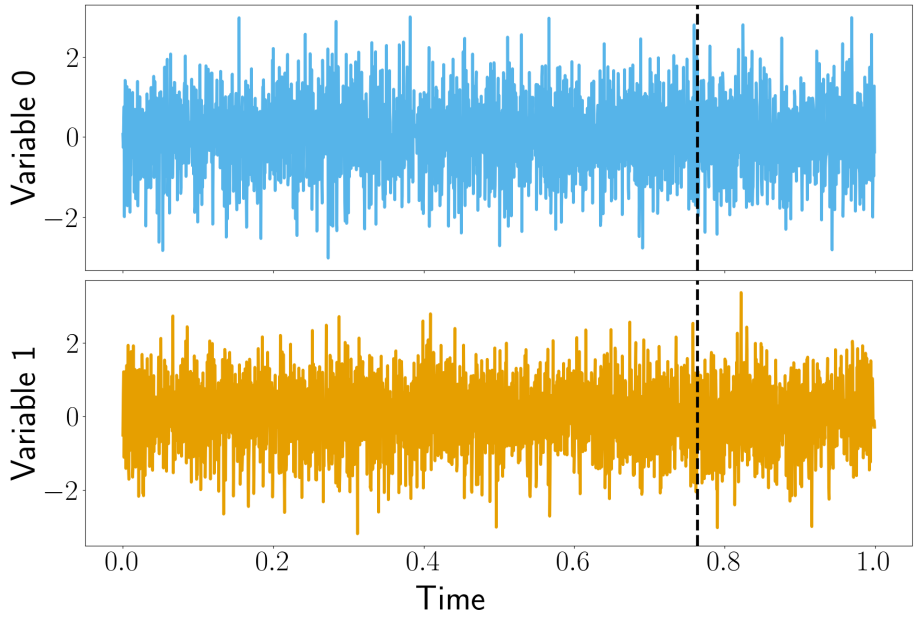


Figure C.3: Realization of $\mathrm{NLARU}_2(4)$ for one data seed. Dashed lines indicate true change-points.

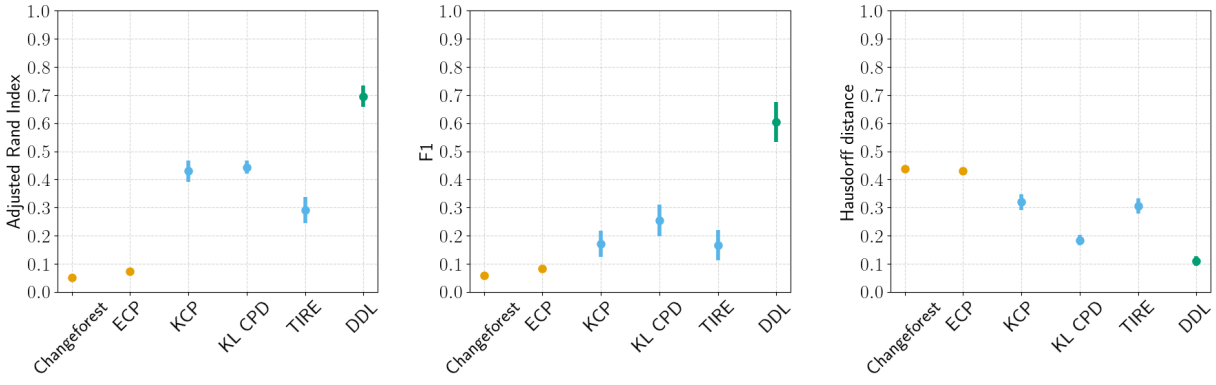


Figure C.4: Simulation results for $\mathrm{NLAR}_3(4)$ using misspecified number of lags $l = 2$. True number of lags are $l = 4$. Vertical lines indicate 95% confidence intervals.

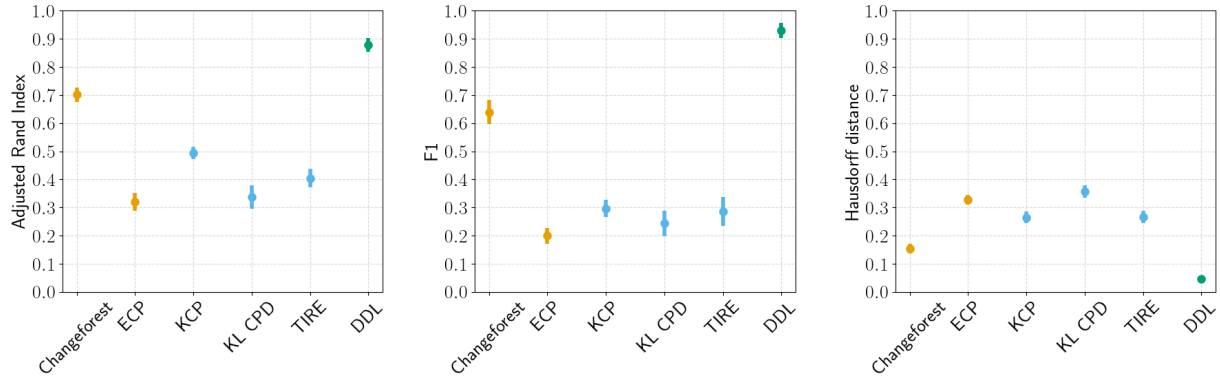


Figure C.5: Simulation results for $\mathrm{VAR}_3(2)$ using misspecified number of lags $l = 1$. True number of lags are $l = 2$. Vertical lines indicate 95% confidence intervals.

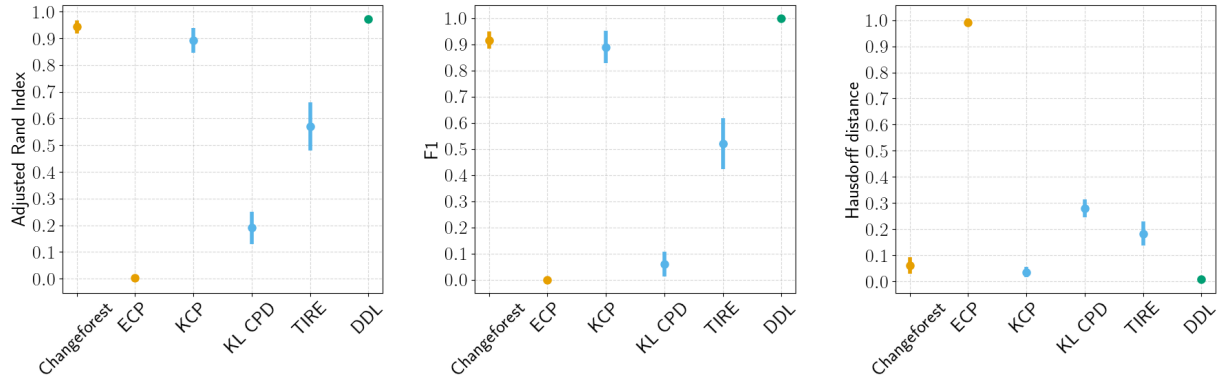


Figure C.6: Simulation results for $\mathrm{NLARU}_2(4)$ using misspecified number of lags $l = 2$. True number of lags are $l = 4$. Vertical lines indicate 95% confidence intervals.