Edge Estimation and Community Detection in Time-varying Networks

by

Jie Jian

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Statistics

Waterloo, Ontario, Canada, 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: **Emma Jingfei Zhang**
Associate Professor, Goizueta Business School,
Emory University

Supervisors: **Peijun Sang**
Assistant Professor, Dept. of Statistics & Actuarial Science,
University of Waterloo

Mu Zhu
Professor, Dept. of Statistics & Actuarial Science,
University of Waterloo

Internal Member: **Pengfei Li**
Professor, Dept. of Statistics & Actuarial Science,
University of Waterloo

Greg Rice
Associate Professor, Dept. of Statistics & Actuarial Science,
University of Waterloo

Internal-External Member: **Anindya Sen**
Professor, Dept. of Economics,
University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The material presented in Chapter 3 was co-authored with Prof. Peijun Sang and Prof. Mu Zhu. This work has been published in

> Jian, J., Sang, P. and Zhu, M. (2024) Two Gaussian regularization methods for time-varying networks. Journal of Agricultural, Biological, and Environmental Statistics.

The material presented in Chapter 4 was co-authored with Prof. Mu Zhu and Prof. Peijun Sang and has been submitted for review. The submitted manuscript can be found in

> Jian, J., Zhu, M. and Sang, P. (2023) Restricted Tweedie stochastic block models. arXiv preprint arXiv:2310.10952.

Abstract

In modern statistics and data science, there is a growing focus on network data that indicate interactions among a group of items in a complex system. Scientists are interested in these data as they can reveal important insights into the latent structure present among the nodes of a network. The emerging family of statistical methods effectively addresses these modeling demands in static networks. However, the evolving nature of network structures over time introduces unique challenges not present in static networks. Specifically, in dynamic networks, we want to characterize their smooth change which also controls the model complexity. To achieve this, we need to impose structural assumptions about the similarity of neighboring networks, and this usually will pose computational challenges.

This thesis studies three aspects of the statistical analysis in time-varying network problems. First, to identify the dynamic changes of associations among multivariate random variables, we propose a time-varying Gaussian graphical model with two different regularization methods imposed to characterize the smooth change of neighboring networks. These methods lead to non-trivial optimization problems that we solve by developing efficient computational methods based on the Alternating Direction Method of Multipliers algorithm. Second, given the observed time-varying financial relationships among nodes, such as their trading amounts in dollars, we propose new stochastic block models based on a restricted Tweedie distribution to accommodate non-negative continuous edge weights with a positive probability of zero counts. The model can capture dynamic nodal effects. We prove that the estimation of the dynamic covariate effects is asymptotically independent of assigned community labels, allowing for an efficient two-step algorithm. Third, when the timestamp of node interactions is accessible, we aim to enhance the modeling of the distribution of survival time of network interactions, especially in the presence of censoring. In addressing this, we employ Cox proportional hazard models to investigate the influence of community structures on the formation of networks. Overall, this thesis provides new methods for modeling and computing time-varying network problems.

Acknowledgements

I have been the truly fortunate recipient of inspiring mentorships from my supervisors, Dr. Peijun Sang and Dr. Mu Zhu. I want to thank them for allowing me to bask in their generously offered research insights, wisdom, and encouragement. They consistently guided me in focusing on the essential facets of research problems. Without their patient listening to my thoughts and ideas, including the less polished ones, and their meticulous word-by-word revision of substantial parts of my writing, this thesis would never have existed. This thesis is dedicated to them, who know I am not perfect but still believe in me.

I am indebted to Dr. Emma Jingfei Zhang, Dr. Pengfei Li, Dr. Greg Rice, and Dr. Anindya Sen, for their time and expertise in evaluating my work and providing invaluable suggestions. I am especially grateful to Dr. Pengfei Li for his instrumental support in shaping my PhD journey.

I have also been blessed with insights that have enriched my research, received encouragement at crucial moments, and benefited from various opportunities provided by fellow graduate students and professors within the department. Thank you all for being part of this supportive community. Particularly, beyond the department, I extend my gratitude to Dr. Benjamin Bagozzi, Dr. Owen Ward, Dr. Pengyu Liu, Dr. Lucy Gao, and Dr. Yu-Ru Liu.

I would like to thank the Department of Statistics and Actuarial Science and the Writing and Communication Centre for excellent resources and academic support. I also extend sincere thanks to the University of Waterloo and the Government of Ontario for their invaluable funding support.

Lastly, I am grateful to my parents for their unwavering love and the freedom they have given me. Above all, I am deeply thankful to have a husband with whom I share mutual understanding and compromise. Thank you for being a great man. I owe you everything.

Dedication

To my supervisors, who have supported me endlessly throughout this process.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the high proficiency of data acquisition, we are now living in a data-rich era where the network data reflecting the complicated interactions among a group of individuals are exploding exponentially and being collected every single day. Unfortunately, being data-rich does not necessarily imply being information-rich. For example, well-developed neuroimaging techniques such as EEG and fMRI allow researchers to directly collect an abundance of brain signals in the form of time series in different brain regions, but how can we sort out the intrinsic relationships between different brain regions? Additionally, even when the relationships among a set of items are already available, what meaningful information can we extract from those networks? Furthermore, when considering non-interactions as the censoring of an event, how can we infer meaningful information about the timestamp of interactions? Mathematics and statistics are the tools to rigorously shed light on the latent properties and patterns from fluent network data.

This thesis mainly focuses on developing new methodologies in time-varying network analysis. Network analysis – the process of recovering latent relational structures between a set of random variables – is encountered in various areas of natural and social sciences, engineering, and technology. Amidst numerous types of network problems, the time-varying network, where a network structure evolves over time in response to the exterior and interior impacts, is a large collection of enormous interest. Learning a time-varying network through the extension of existing methods is usually arduous because the dynamic change of the graphical structure makes the problem high-dimensional and heterogeneous and thus difficult to characterize. In this thesis, we present a systematic study of the theory and applications of time-varying network analysis and provide future research directions.

In Chapter 2, we state three different types of network problems that we are going to study. Recent developments, various important methods, and algorithms are summarized for subsequent Chapters 3, 4, and 5.

In Chapter 3, we study the problem of uncovering the temporal evolution of interactions and associations in a network. We model time-varying network data as realizations from multivariate Gaussian distributions with precision matrices that change over time. To facilitate parameter estimation, we require not only that each precision matrix at any given time point be sparse, but also that precision matrices at neighboring time points be similar. We accomplish this with two different algorithms, by generalizing the elastic net and the fused LASSO, respectively. While similar approaches in the literature for modeling time-varying networks have predominantly extended the graphical LASSO of Friedman et al. (2008), we extend the regression approach of Meinshausen and Bühlmann (2006) and subsequently of Peng et al. (2009). This allows us to explicitly focus on and work with the partial correlation coefficients, which are more directly meaningful and interpretable parameters for the environmental and biological sciences. We develop efficient algorithms and convenient degree-of-freedom formulae for choosing tuning parameters. The proposed methods are demonstrated through simulation studies. By applying them to an hourly temperature data set, we detect interesting time-varying connectivity among thirteen Canadian cities. Similarly, application to an fMRI dataset uncovers noteworthy differences in brain connectivity between healthy individuals and ADHD patients.

In Chapter 4, we explore the problem of clustering nodes into different communities (i.e., community detection problem) according to the edge weights on a time-varying network. The stochastic block model is a widely used framework for community detection in networks, where the network structure is typically represented by an adjacency matrix. However, conventional stochastic block models are not directly applicable to an adjacency matrix that consists of non-negative continuous edge weights with positive mass at zero. To model the international trading network, where edge weights represent trading values between countries, we propose an innovative stochastic block model based on a restricted Tweedie distribution. Additionally, we incorporate nodal information, such as the geographical distance between countries, and account for its dynamic effect on edge weights. Notably, we show that given a sufficiently large number of nodes, estimating this covariate effect becomes independent of the community labels of each node when computing the maximum likelihood estimator of parameters in our model. This result enables the development of an efficient two-step algorithm that separates the estimation of covariate effects from other parameters. We demonstrate the effectiveness of our proposed method through extensive simulation studies and an application to real-world international trading data.

In Chapter 5, we investigate the community structures on a network when modeling the event history of time-stamped interactions among nodes. This problem, of particular significance in political science, is exemplified by exploring durations of crucial international relations, such as treaties and wars among nations. Existing works that study this type of data through the use of time-to-event analysis usually treat these interactions as independent events given the covariates without considering the network structure. Ignoring the heterogeneity brought by the latent communities of these network agents can be risky. For instance, it may violate the common assumption for many proportional hazard models that given the covariates, the occurrence or timing of an event for one subject does not influence or provide information about the occurrence or timing of events for other participants. To address this issue, we propose an approach, using the Cox proportional hazard model as a case study, where we assume that the hazard ratio between two interactions depends on the community labels of the nodes involved, and it remains constant over time. Simulation studies are conducted to assess the performance of the proposed approach. Application to a network of timestamps of diplomatic relations reveals intriguing community patterns among nations.

The contributions from this thesis demonstrate novel ways to estimate the edges, cluster the nodes, and analyze the timestamps of interactions in time-varying networks.

Chapter 2

Background

In this chapter, we state the problems of graphical models and detecting communities in network modeling. We provide brief literature reviews of the related tools that have been used in the studies, which will prepare readers to appreciate the studies subsequently presented in Chapters 3, 4, and 5.

2.1 Estimation of Time-Varying Networks

In this section, we introduce the motivation for estimating time-varying networks and discuss the most recent studies. A brief review of the static Gaussian graphical model and its estimation is also provided given that we will extend it in Chapter 3 to depict the dynamic networks.

One key challenge with the static Gaussian graphical model itself, and in the extension to the dynamic case, is the regularization of the networks and their change. To tackle the issue, we will use the regularization techniques in the later chapter. Therefore, at the end of this section, we also review the Alternating Direction Method of Multipliers method, as it is crucial to numerically solving the newly proposed methods.

2.1.1 Problem Statement

Many complex systems in the scientific domain can be characterized as a network structure providing relational information among a set of entities. Modeled mathematically as a graph, the individual components can be represented as the nodes and their relationships can be represented as connecting edges. In many cases, only the information of nodes is observable, and the inference of latent associations among nodes called *edge estimation*, is of great interest.

Numerous examples of edge estimation in the fields of science include gene regulatory networks (Hecker et al., 2009), protein-protein interaction (Schwikowski et al., 2000), metabolic network (Jeong et al., 2000), and functional brain network (Friston, 1994). Interest in networks has been accompanied by developments of powerful network edge estimation tools and algorithms across the decades, the most mainstream category of which is the Gaussian Graphical Model (GGM). Many methods have been proposed to learn GGM when there is a single snapshot of data is observed, including Meinshausen and Bühlmann (2006), Yuan and Lin (2007), Friedman et al. (2008), Banerjee et al. (2008), Peng et al. (2009). More details of static GGM will appear in Section 2.1.2.

However, given the dynamic nature of real-life networks, a key challenge is to estimate dynamic networks via GGM. We highlight the scientific significance of this task through the following two specific networks of interest.

The brain network With 86 billion neurons connected by 150 trillion synapses carrying electrical or chemical impulses between neurons, the human brain shows correlated activities in different regions. From these activities, structural linkages can be inferred, which leads to the concept of functional connectivity which is defined as the macro-level correlation of activity between pairs of brain regions. Functional connectivity forms a network in the human brain, but it can only be quantified using statistical dependence measures extracted from datasets using techniques such as electroencephalography (EEG), local field potentials (LFP), magnetoencephalography (MEG), positron emission tomography (PET), or functional magnetic resonance imaging (fMRI) (Friston, 1994; Menon and Krishnamurthy, 2019). Functional connectivity can be used to find clinical biomarkers, classify patients into biologically-based groups, designate treatment targets, track disease progression over time, and even forecast future disease onset, progression, and treatment outcomes (Zhang et al., 2021; Saggar and Uddin, 2019). However, a human brain is always in active motion, whether it is engaged in a task or not, and the interactions between different regions change with time. Estimating a static network with the entire dynamic network data ignores dynamic transitions between functional brain networks and fails to adequately explain the neural activity.

The climate network Climate is another example of a complex dynamical system characterized by spatiotemporal patterns, and climate connectivity can be represented as a network composed of a set of random variables like sites in a spatial grid (nodes) interconnected by a set of interactions (edges). Such a climate network can illuminate previously unobserved but crucial spatiotemporal patterns of climatic relevance between spatially dispersed grid points, and these grids may regulate particular pertinent features of climate variability on a regional or even global scale (Ferreira et al., 2021). As the earth's climate is actively and constantly changing due to internal (climatic, e.g., from cloud formation to the global circulation) and external (non-climatic, e.g., human actions) disruptions and perturbation, modeling climate connectivity as a static network where a single snapshot of the network is observed can be misleading (Kittel et al., 2021).

These examples underscore the growing interest in the development of time-varying GGMs. In Chapter 3, our goal is to propose a method for extending static GGMs to estimate networks that: (1) exhibit sparsity at each temporal time point; (2) transition smoothly from one time point to the next; (3) can be computed efficiently. The justification for the sparsity of each individual network will be further explored in the next Section 2.1.2 as we introduce the GGM. Additionally, the smoothness assumption will be illustrated in Section 2.1.3 when we discuss the recent developments in time-varying GGMs.

2.1.2 Gaussian Graphical Model

Statistical and probabilistic methods for estimating network edges are promising, as they seek the statistical features of these networks rather than explicit answers, which reflect the uncertainty of the true world.

The most representative framework in the network estimation is the *probabilistic graphical model*, employed to investigate the conditional relationships among a set of p random variables. In a graph, represented as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, the p random variables are nodes $\mathcal{V} = \{x_1, \cdots, x_p\}$, and the edges in $\mathcal{E}$ characterize the conditional dependence structure between the random variables. A partial correlation between any two random variables x_i and x_j, denoted as ρ_{ij}, measures the degree of association between these two random variables given other variables. Consequently, partial correlations can be used to depict the conditional dependency structure. More specifically, no conditional relationships between two random variables lead to the absence of an edge between two nodes, and the thickness of an edge, which is determined by the magnitude of the corresponding partial correlation, represents the strength of the relationships between nodes. The edges are defined as

$$\begin{aligned}\mathcal{E} = \{(i, j) : x_i \text{ and } x_j \text{ are conditionally dependent given } X\backslash\{i, j\},\\ \text{or equivalently } \rho_{ij} \neq 0\},\end{aligned}$$

where $X\backslash\{i, j\} = \{x_k : k \neq i, j \text{ and } 1 \leq k \leq p\}$. The objective of network edge estimation is to identify the edges in the set $\mathcal{E}$, or equivalently, to estimate the partial correlations.

Many real-world distributions are surprisingly well approximated by Gaussian distribution, so we mainly focus on a special type of probabilistic graphical model encoded with a multivariate Gaussian distribution, namely *GGM*. The advantage of GGM is that its inverse covariance matrix is directly linked to the partial correlations, in which the entries also reveal the conditional dependencies, therefore GGM can provide a mathematically convenient way to characterize the network.

Consider a GGM with p nodes jointly following a p-dimensional Gaussian distribution: $(x_1, \cdots, x_p)^\top \sim N(\mathbf{0}, \mathbf{\Sigma}_{p\times p})$. Let $\mathbf{\Omega} := (\sigma^{ij})_{p\times p}$ denote the inverse covariance matrix $\mathbf{\Sigma}^{-1}$, which is also called *concentration matrix*. Notably, under multivariate normality, a zero-entry in the inverse covariance matrix indicates conditional independence between the two random variables given the remaining ones (Dempster, 1972). Specifically, $\sigma^{ij} = 0$ if and only if $x_i \perp x_j | X\backslash\{i, j\}$. Moreover, in GGM, partial correlations can be calculated directly from the concentration matrix (Lauritzen, 1996):

$$\rho_{ij} = -\frac{\sigma^{ij}}{\sqrt{\sigma^{ii}\sigma^{jj}}}. \tag{2.1}$$

As a result, estimating a network is equal to estimating a concentration matrix or partial correlations.

Due to sampling variation, directly taking the inverse of the sample covariance matrix leads to a dense network consisting of spurious relationships. Furthermore, a large ratio of the number of estimators over sample size, p/n, also makes the sample covariance matrix a poor estimator. Consequently, the main interest in network edge estimation is to infer a sparse concentration matrix or a sparse partial correlation matrix. Focusing on our proposed method in the subsequent Chapter 3, in the rest of this section we give a brief introduction of the penalized maximum likelihood techniques to estimate a concentration matrix that represents an interpretable network, while other methods to regularize an inverse covariance matrix include banding and thresholding the sample covariance matrix (Bickel and Levina, 2008), regularized the Cholesky decomposition of a covariance matrix (Levina et al., 2008), and the Bayesian inference for the posteriori of the concentration matrix (Wang, 2012).

The penalized maximum likelihood estimation considers two components: the first component directly models the distribution of the network data with a log-likelihood function from GGM, and the second component brings a shrinkage effect on the estimated concentration matrix. Accordingly, the penalized log-likelihood takes the form

$$\mathcal{L}(X; \mathbf{\Omega}) = L(X; \mathbf{\Omega}) - \lambda_1 P_{\text{shrink}}(\mathbf{\Omega}). \tag{2.2}$$

where $P_{\text{shrink}}(\mathbf{\Omega}) = \|\mathbf{\Omega}\|_1$ represents the l_1 regularization penalty. The l_1 norm sums up the absolute values of the elements of $\mathbf{\Omega}$, and λ_1 is the tuning parameter controlling the sparsity of the inverse covariance matrix. Maximizing the penalized likelihood function with the l_1 penalty tends to select the significant variables and shrink the rest parameters to zero.

Two approaches, covariance selection and neighborhood selection, arise from distinct emphases when modeling the multivariate Gaussian distributions, yielding two $L(X; \mathbf{\Omega})$.

Covariance selection framework aims to identify the zero elements in the inverse covariance matrix (Dempster, 1972). $L(X; \mathbf{\Omega})$ comes from the log-likelihood of a multivariate Gaussian distribution, in which

$$L(X; \mathbf{\Omega}) = \log \det \mathbf{\Omega} - \operatorname{tr}(S\mathbf{\Omega}), \tag{2.3}$$

where S is the empirical covariance matrix. Numerous methods and algorithms have been developed for maximizing (2.2) in cases where $L(X; \mathbf{\Omega})$ is specified as (2.3) (see e.g., Yuan and Lin, 2007; Friedman et al., 2008; Banerjee et al., 2008).

Neighborhood selection is another class of methods to estimate a GGM, which targets on estimating the neighborhood of any given variable individually in the regression context. A multiple regression to predict x_i using all other variables provides an alternative to estimating the partial correlation (Meinshausen and Bühlmann, 2006; Lauritzen, 1996; Peng et al., 2009):

$$x_i = \sum_{j \neq i} \beta_{ij} x_j + \epsilon_i, \tag{2.4}$$

where the prediction error variance $Var(\epsilon_i) = 1/\sigma^{ii}$. According to Lauritzen (1996), the regression coefficient $\beta_{ij} = -\sigma^{ij}/\sigma^{ii}$, and combining it with (2.1) leads to

$$\rho_{ij} = \beta_{ij}\sqrt{\frac{\sigma^{ii}}{\sigma^{jj}}} = \beta_{ji}\sqrt{\frac{\sigma^{jj}}{\sigma^{ii}}}. \tag{2.5}$$

In the context of neighborhood selection, $L(X; \mathbf{\Omega})$ adopts the following form:

$$L(X; \beta) = -\sum_{i=1}^{n} \|x_i - \sum_{j \neq i} \beta_{ij} x_j\|^2. \tag{2.6}$$

This expression arises from performing sequential multiple regressions (2.4). Therefore, the corresponding tailored (2.2) is $L(X; \mathbf{\Omega}) = L(X; \beta) - \lambda_1 \sum_{i,j} |\beta_{ij}|$, providing estimates of partial correlations.

2.1.3 Recent Developments

Whereas the static GGM is estimated under the assumption that data X are drawn identically from a generating multivariate distribution $N(\boldsymbol{\mu}, \mathbf{\Sigma}_{p \times p})$, when a multivariate time series is observed, we want to recover a time-varying network from the dynamic data. Therefore, in this section, we give an introduction to the recent developments in the estimation of time-varying networks.

A natural solution to the estimation of time-varying networks is to assume a temporal GGM at each temporal time point, i.e., $x(t) = (x_1(t), \cdots, x_p(t)) \sim N(\mu(t), \Sigma(t))$, and thus to estimate a network at each single time point and meanwhile control the change from one network to the next. To this end, a large group of methods imposes a penalty term of the difference between inverse covariance matrices at neighboring time points on the summation of the temporal penalized Gaussian log-likelihood (2.2) over all time points:

$$L = \sum_{t=1}^{T} [L(X(t); \mathbf{\Omega}(t)) - \lambda_1 P_{\text{shrink}}(\mathbf{\Omega}(t))] - \lambda_2 P_{\text{smooth}}(\{\mathbf{\Omega}(t)\}_{t=1}^{T}). \tag{2.7}$$

To achieve temporal smoothness, different norms can be utilized in P_{smooth}. Primarily, two types of models exist based on distinct forms of P_{smooth}: generalized fused LASSO and generalized group LASSO. Both models modify the LASSO penalty to take into account the ordering of the parameters. We first introduce these two different models and their original forms, and detail the related work on time-varying GGM that applies these models.

The likelihood function (2.7) of time-varying GGM that utilizes the generalized fused LASSO model takes the form

$$L = \sum_{t=1}^{T} L(X(t);\mathbf{\Omega}(t)) - \lambda_1 \sum_{t=1}^{T}\sum_{i<j} |\Omega_{ij}(t)| - \lambda_2 \sum_{t=2}^{T}\sum_{i<j} |\Omega_{ij}(t) - \Omega_{ij}(t-1)|, \tag{2.8}$$

which is a generalized version of the original fused LASSO. The original fused LASSO was first proposed by Tibshirani et al. (2005) to solve a regression problem when the coefficients need to be sparse and the features have a natural order:

$$\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p-1} |\beta_{j+1} - \beta_j|. \tag{2.9}$$

The last term enforces the similarities across consecutive regression coefficients via l_1 norm, which will make the estimated coefficients be piecewise constant in the order it takes the difference. (2.8) imitates this way to punish the difference of the coefficients. Consequently, the l_1 norms for the coefficients and their pair-wise differences in (2.8) together will encourage the sparsity of the temporal network and also the sparsity of their corresponding differences. Generalized fused LASSO is favorable for detecting abrupt changes and the corresponding change points. Many methods use generalized fused LASSO or its variants to characterize the change of temporal GGM, including Ahmed and Xing (2009), Kolar et al. (2010), Kolar and Xing (2011), Danaher et al. (2014), Monti et al. (2014), Gibberd and Nelson (2017), and Hallac et al. (2017). Danaher et al. (2014) uses l_1 norm to control the difference of inverse covariance matrix at any two time points, i.e., $P_{\text{smooth}} = \sum_{t_1 \neq t_2} \sum_{i<j} |\Omega_{ij}(t_1) - \Omega_{ij}(t_2)|$, and it adopts covariance selection likelihood (2.3) as $L(X(t);\mathbf{\Omega}(t))$ in the time-varying likelihood function (2.7). Moreover, a fast Alternating Direction Method of Multipliers algorithm is proposed in Danaher et al. (2014) to solve the corresponding convex optimization problems. Monti et al. (2014) considers the fused-type penalty in (2.8), but it uses a weighted sum of sample covariance at near time points as the local covariance matrices at the current time point in the covariance selection likelihood (2.3) as $L(X(t);\mathbf{\Omega}(t))$. Yang et al. (2015b) proposes a new second-order method, and derive the necessary and sufficient condition for the use of their method to maximize (2.8) where $L(X(t);\mathbf{\Omega}(t))$ is the covariance selection likelihood (2.3). Some methods apply fused LASSO-type penalty, but instead of using GGM to model $L(X(t);\mathbf{\Omega}(t))$ in (2.7) the distribution of a set of random variables in a network is modeled differently. For example, Ahmed and Xing (2009) models the temporal graphical model as evolving Markov random fields instead of GGM, which identifies transient relationships through logistic regression

formalism.

Generalized group LASSO is the other common form for time-varying GGM. The original sparse group LASSO proposed by Simon et al. (2013) assumes that a group structure exists among the coefficients so that all the coefficients in that group are non-zero, so the l_2 norm is applied to achieve the variable selection within a group of coefficients. Group LASSO resembles the form

$$\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{k=1}^{K} \|\beta_k\|_2, \tag{2.10}$$

where β_k represents the vector containing the coefficients in the k^{th} group. Since the l_2 norm is not differentiable at 0, some groups will be completely zeroed out. Gibberd and Nelson (2017) uses Frobenious norm to capture the change of graphical structures over time, i.e., $P_{\text{smooth}} = \sum_{t=2}^{T} \|\boldsymbol{\Omega}(t) - \boldsymbol{\Omega}(t-1)\|_F = \sum_{t=2}^{T} \left(\sum_{i<j} (\Omega_{ij}(t) - \Omega_{ij}(t-1))^2 \right)^{1/2}$, where the changes of each entry in the concentration matrix from one time point to the next form a group. Yang and Peng (2020) considers the local weighted sum of sample covariance in (2.3), and meanwhile employ a local group-LASSO penalty $\sum_{i \neq j} (\sum_t \Omega_{ij}(t)^2)^{1/2}$ where the group is based on each temporal entry across all time points.

In addition to the two types of penalty terms on the difference of the successive networks, the smoothing spline can be used to characterize the evolution of the network as well, so that the temporal smoothness can be achieved automatically. Xue et al. (2020) extends the neighborhood selection framework by proposing a semi-parametric method that models all partial correlations as smooth functions via the same set of B-spline bases functions. A spline method naturally offers smoothness to partial correlations. B-spline functions enjoy the local support property which means that each B-spline function is only positive in a small portion of the whole interval, hence the sparsity of spline coefficients leads to the sparsity of the partial correlations. Accordingly, the sum of all temporal likelihood function (2.2) is maximized where the l_1 norm of spline coefficients is penalized.

Instead of maximizing (2.7) directly, some papers reflect the dynamic interactions between nodes using the change of the sample covariance matrix. Zhou et al. (2010) estimates each temporal inverse covariance $\Sigma^{-1}(t)$ separately by maximizing (2.2), where the empirical covariance matrix S is modified as a weighted sum of all temporal sample covariance matrices according to their time difference to the current time t. Kolar and Xing (2011) further proves that such a method can be consistently estimated in the high dimensional setting under suitable conditions.

Reference	Gaussian likelihood	Shrinkage	Smoothness
Danaher et al. (2014)	sum of temporal (2.3)	$\sum_{t=1}^{T}\sum_{i<j}\lvert\Omega_{ij}(t)\rvert$	$\sum_{t_1\neq t_2}\sum_{i<j}\lvert\Omega_{ij}(t_1) - \Omega_{ij}(t_2)\rvert$; $\sum_{i\neq j}(\sum_{t=1}^{T}\Omega_{ij}(t)^2)^{1/2}$
Monti et al. (2014)	sum of temporal (2.3); local weighted S	$\sum_{t=1}^{T}\sum_{i<j}\lvert\Omega_{ij}(t)\rvert$	$\sum_{t=2}^{T}\sum_{i<j}\lvert\Omega_{ij}(t)-\Omega_{ij}(t-1)\rvert$
Yang et al. (2015b)	sum of temporal (2.3)	$\sum_{t=1}^{T}\sum_{i<j}\lvert\Omega_{ij}(t)\rvert$	$\sum_{t=2}^{T}\sum_{i<j}\lvert\Omega_{ij}(t)-\Omega_{ij}(t-1)\rvert$
Gibberd and Nelson (2017)	sum of temporal (2.3); local weighted S	$\sum_{t=1}^{T}\sum_{i<j}\lvert\Omega_{ij}(t)\rvert$	$\sum_{t=2}^{T}\lVert\mathbf{\Omega}(t) - \mathbf{\Omega}(t-1)\rVert_F$
Yang and Peng (2020)	sum of temporal (2.3); local weighted S	$\sum_{t=1}^{T}\sum_{i<j}\lvert\Omega_{ij}(t)\rvert$	$\sum_{i\neq j}(\sum_{t}\Omega_{ij}(t)^2)^{1/2}$
Ahmed and Xing (2009)	logistic regression method; local weighted S	$\sum_{t=1}^{T}\sum_{i<j}\lvert\Omega_{ij}(t)\rvert$	$\sum_{t=2}^{T}\sum_{i<j}\lvert\Omega_{ij}(t)-\Omega_{ij}(t-1)\rvert$
Zhou et al. (2010)	temporal (2.3); local weighted S	$\sum_{i<j}\lvert\Omega_{ij}(t)\rvert$	-
Xue et al. (2020)	sum of temporal (2.6)	$\sum\lvert\beta\rvert$	-

Table 2.1: Recent developments for dynamic edge estimation

2.1.4 Alternating Direction Method of Multipliers (ADMM)

The *Alternating Direction Method of Multipliers* (ADMM) was first introduced in 1974 and has been used under the name of ALG2 in Physics and Mechanics, among others (Glowinski, 2014). As an efficient first-order method for solving convex optimization problems, it becomes increasingly popular under the name of ADMM since the recent decade due to its wide applications in hot topics including statistical learning, image processing, data mining, and so on (Boyd et al., 2011). In this section, we describe the mechanism of ADMM.

ADMM ties together two methods, *Dual Ascent method* and *Method of Multipliers*. We first introduce the Dual Ascent method. Let's consider the equality-constrained convex optimization problem, namely the *primal problem*:

$$\begin{aligned} &\min_x f(x), \\ s.t.\ &Ax = b. \end{aligned} \tag{2.11}$$

The Lagrangian for this problem is

$$L(x,y) = f(x) + y^\top (Ax - b). \tag{2.12}$$

The primal problem (2.11) is associated with a *dual problem*

$$\max_y g(y),\ \ g(y) = \inf_x L(x,y), \tag{2.13}$$

where y is the dual variable or Lagrange multiplier.

Let x^* be a solution to the primal problem (2.11) and y^* to the corresponding dual problem (2.13). The difference $f(x^*) - L(x^*, y^*)$ is called the duality gap. If the duality gap is zero, then we say that the strong duality holds in this convex optimization problem. Assuming that f is convex and the strong duality holds, the optimal values of the primal and dual problems are the optimal ones in the minimization of the Lagrangian (2.12). Hence, we can recover the solution to the primal problem, x^*, from a dual optimal point y^* as

$$x^* = \arg\min_x L(x, y^*). \tag{2.14}$$

Accordingly, we can address the optimization problem (2.11) by iteratively solving y^* from (2.13) and solving x^* from (2.14), which leads to the *Dual Ascent method*:

$$x^{k+1} = \arg\min_x L(x, y^k); \tag{2.15}$$

$$y^{k+1} = y^k + a \cdot (Ax^{k+1} - b), \tag{2.16}$$

where the update of y is done by gradient descent of the maximization problem (2.13), therefore a is a positive constant. When $f(x)$ is separable, more specifically, it can be decomposed as some independent functions with the partition of the variable x into sub-vectors, $f(x) = \sum_{i=1}^N f_i(x_i)$, the $x-$minimization (2.15) can be split into N decentralized and parallel updates. The advantage of this Dual Ascent method is the computational efficiency brought by such decomposition, while the disadvantage is that only the strong convexity of $f(x)$ ensures the convergence, otherwise the strong duality does not hold.

Augmented Lagrangian method modifies the problem by adding a quadratic term to loosen the strong condition on $f(x)$. For a penalty parameter $\rho > 0$, we consider the problem

$$\begin{aligned} \min_x f(x) + \frac{\rho}{2}\|Ax - b\|_2^2, \\ s.t.\ Ax = b, \end{aligned} \tag{2.17}$$

which can be interpreted as a tradeoff between minimizing $f(x)$ and making Ax close to b for some weight ρ. The problem (2.17) is equivalent to the primal problem (2.11), as any x that satisfies the equality $Ax = b$ makes the quadratic term zero.

The Lagrangian of (2.17) is

$$L_{ALM}(x, y) = f(x) + y^\top(Ax - b) + \frac{\rho}{2}\|Ax - b\|_2^2. \tag{2.18}$$

Applying the Dual Ascent method to solve the modified problem is the iteration between

$$\begin{aligned} x^{k+1} &= \arg\min_x L_{ALM}(x, y^k); \\ y^{k+1} &= y^k + a \cdot (Ax^{k+1} - b). \end{aligned} \tag{2.19}$$

Furthermore, we use ρ as the step size, i.e., $a = \rho$. Better convergence properties are the main advantage of this Augmented Lagrangian method. However, there is a price for including the extra quadratic term in (2.18). If $f(x)$ is separable, the $x-$update in Dual Ascent iteration (2.15), mainly the minimization of $f(x)$, can be calculated efficiently, for example, by coordinate descent or dual decomposition. Unfortunately, x-minimization in the Augmented Lagrangian method cannot be calculated separately in parallel due to the quadratic term. ADMM tries to tackle this challenge by combining the decomposability of dual ascent with the superior convergence properties of the method of multipliers.

We consider the new form of the original problem (2.11) by splitting the variable in (2.11) into two parts x and z:

$$\min_{x,z} f(x) + g(z), \tag{2.20}$$

$$s.t.\ Ax + Bz = c. \tag{2.21}$$

As in the method of multipliers, the augmented Lagrangian is

$$L_a(x, y, z) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2. \tag{2.22}$$

The Dual Ascent method iterates between

$$\begin{aligned} (x^{k+1}, z^{k+1}) &= \arg\min_{x,z} L(x, z, y^k); \qquad (2.23)\\ y^{k+1} &= y^k + \rho \cdot (Ax^{k+1} + Bz^{k+1} - c), \end{aligned}$$

where (2.23) is solved jointly with respect to the two primal variables. ADMM updates x and z in an alternating way, which explains its name *alternating direction*. That is, at the $k + 1^{th}$ iteration ADMM performs the following:

$$x^{k+1} = \arg\min_{x} L(x, z^k, y^k); \tag{2.24}$$

$$z^{k+1} = \arg\min_{z} L(x^{k+1}, z, y^k); \tag{2.25}$$

$$y^{k+1} = y^k + \rho \cdot (Ax^{k+1} + Bz^{k+1} - c). \tag{2.26}$$

In Chapter 3, we will use the scaled form of ADMM instead of the above original form. We set $u = y/\rho$ as the scaled dual variable, then the augmented Lagrangian takes the form

$$L_{scaledADMM}(x, y, z) = f(x) + g(z) + \rho u^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2. \tag{2.27}$$

The corresponding three steps (2.24)-(2.25) for the scaled ADMM are expressed as

$$\begin{aligned} x^{k+1} &= \arg\min_{x} L_{scaledADMM}(x, z^k, u^k);\\ z^{k+1} &= \arg\min_{z} L_{scaledADMM}(x^{k+1}, z, u^k);\\ u^{k+1} &= y^k + (Ax^{k+1} + Bz^{k+1} - c). \end{aligned}$$

The original formulation of ADMM and the scaled form are exactly equivalent, and a wider popularity of the scaled form simply stems from the fact that it allows for more concise expressions. In this subsection, we have demonstrated how the decomposition properties and the parallelization potentialities make ADMM an efficient and powerful algorithm. A systematic review can be found in Boyd et al. (2011).

2.2 Community Detection of Time-varying Networks

The purpose of this section is two-fold: to introduce the community detection problem and one specific resolution–the stochastic block model (SBM), and to illustrate the inference methods for the SBM. In Section 2.2.1, we discuss the challenges of applying the SBM to address the community detection of time-varying networks. Specifically, we explore two distinct community detection problems within the realm of time-varying networks, setting the stage for the proposed methods in Chapters 4 and 5. In Section 2.2.3, we showcase the effectiveness of powerful inference methods–the Variational Expectation Maximization and the Expectation Maximization–in solving the vanilla stochastic block model. This serves to provide readers with foundational knowledge as we adapt them to solve our proposed models in the later Chapters 4 and 5.

2.2.1 Problem Statement

In Section 2.1.1, we introduce the network problem of estimating the associations among nodes based on multivariate observations. Our attention now shifts to a different network challenge. Here, we have already observed or estimated the networks, that characterize pairwise relations between objects. One important structural property of many real-world networks is the community structure, which indicates the fact that nodes are gathered into several groups (called *communities*) where community members have similar connection patterns. Community detection, also called graph partition, is to cluster the nodes into different communities according to the connection information. Community detection is of significant value in marketing (Bakhthemmat and Izadi, 2021), news recommendations (Gasparetti et al., 2021), political polarization detection (Guerrero-Solé, 2017), and fraud detection (Sarma et al., 2020).

Existing community detection approaches can be roughly divided into two types: algorithm-based (see e.g. Raghavan et al., 2007; Blondel et al., 2008; Bickel and Chen, 2009; Newman and Clauset, 2016; Zhao et al., 2011) and model-based (see e.g. Holland et al., 1983). Except for some heuristic iterative procedures like the label propagation algorithm (Raghavan et al., 2007) and Louvain algorithm (Blondel et al., 2008), algorithm-based methods also achieve community detection by defining a specific objective function and then maximizing it with the most suitable algorithm. Such objective functions can be modularity which measures the strength of division of a network into modules, whose exhaustive maximization is NP-complete, therefore optimization strategies including the greedy algorithm, simulated annealing, extremal optimization, and spectral optimization have been explored in modularity-based methods. Another strategy for detecting communities is to use model-based methods, which focus on how edges are created using a probabilistic model. Popular model-based methods include the stochastic block model (Holland et al., 1983), and latent space models (Hoff et al., 2002). The reader can refer to Fortunato (2010) for a thorough survey of existing community detection approaches.

The SBM is the most representative model-based generative model, which detects communities by formalizing a generative process of a network with rigorous probability distributions. As a generative model, the SBM benefits from a well-defined likelihood function that brings consistent parameter estimates. Another advantage of the SBM is its flexibility, leading to a large number of extensions that also enjoy sound theoretical analysis.

Motivated by the dynamic nature of network structures and the wealth of available data sources, our research explores community detection within time-varying networks. Our focus is directed towards two distinct challenging scenarios, each presenting its unique set of complexities:

Dynamic Network Evolution In this scenario, we collect logged network data at regular intervals, generating a series of temporal networks observed at sequential discrete time points. This network undergoes continuous transformation, evolving dynamically from one time point to the next. Taking international trading as an illustrative example, bilateral relationships fluctuate annually due to shifts in global economic power, the emergence of trade blocs, and the impact of trade policies. Considering community structures is crucial for capturing how trading connections among countries adapt and reshape over time. Chapter 4 will further explore community detection in such time-varying networks, addressing additional challenges that involve modeling the weighted edges, as well as incorporating the covariates effect in network formation and dynamics.

Temporal Interaction Timestamps In this scenario, we not only witness interactions among items but also capture the timestamped interactions occurring within the network. For instance, in a gene regulatory network, where relationships signify the influence of one gene's expression on another, understanding biological processes relies on considering the timing of gene interactions. The dynamic regulation of gene expression plays a pivotal role in shaping critical biological events such as development and the progression of diseases. In political science, different international relations can be modeled by networks, such as the agreements, military alliances and treaties, wars, and diplomatic relations between nations. Analyzing the duration and timing of these interactions aids political scientists in contextualizing current political situations and conflicts, guiding decision-making processes. Understanding latent community structures embedded within the network becomes imperative in unraveling the temporal order and the evolution of connections over time. Therefore, Chapter 5 will explore the intricate challenge of modeling the event history of interactions among nodes, incorporating considerations for community structures and covariate effects.

Detailed literature reviews for these scenarios are included in Chapters 4 and 5, as both problems involve challenges beyond the dynamics of network data. We list them here to underscore that our research aims to contribute to advancing community detection methodologies in time-varying networks.

2.2.2 Vanilla Stochastic Block Model

In this section, we introduce the basic graph notation and formulate a canonical version of the SBM, called *the vanilla stochastic block model*.

Let $\mathcal{G} = (V, E)$ be a graph where $V = \{v_1, \cdots, v_n\}$ is a set of nodes and E is a set of edges. We assume that the network is undirected and has no loop, then it can be characterized by an $n-$by$-n$ symmetric adjacency matrix Y with zero diagonal entries. $Y_{ij} = Y_{ij} = 1$ if node i and node j are connected and 0 otherwise. Each node belongs to one of K groups called communities or blocks. A vector of latent labels $c = (c_1, \cdots, c_n)$ is generated, where c_i takes an integer value from $\{1, 2, \cdots, K\}$ following a multinomial distribution with parameters $\pi = (\pi_1, \cdots, \pi_K)$. It is worth noting that all of SBM-related approaches presented in this thesis work if K, the number of communities, is fixed and prespecified. In practice, K must be decided, but how to determine it is a different problem with a great deal of recent research (see e.g. Zhang et al., 2023; Wang and Bickel, 2017; Chen and Lei, 2018).

The key concept of the SBM is the *stochastic equivalence*, which refers to the assumption that the edge probability between node i and node j depends solely on their memberships. We assume that given the community memberships of two nodes, the connection between them is Bernoulli distributed, and the adjacency matrix Y is generated with

$$Y_{ij} | c_i = k, c_j = l \sim Ber(P_{kl}),\ i \neq j, \tag{2.28}$$

where P is a $K-$by$-K$ block matrix and the P_{kl} describes the probability of connectivity between the nodes from the k^{th} and l^{th} block. For a convenient notation, we decode the latent label c_i, as a vector of length K such that $c_i = (c_{i1}, \cdots, c_{iK})$ where c_{ik} is a $0-1$ value to indicate whether node i belongs to the k^{th} community, and use θ to denote the model parameters P and π. Given the community memberships c and the adjacency matrix Y, the complete likelihood is

$$\mathbb{P}(c, Y|\theta) = \prod_{i=1}^{n} \prod_{k=1}^{K} \pi_k^{c_{ik}} \cdot \prod_{1 \le i < j \le n} \prod_{k,l=1}^{K} \left[P_{kl}^{Y_{ij}} \cdot (1 - P_{kl})^{1-Y_{ij}} \right]^{c_{ik} \cdot c_{jl}}, \tag{2.29}$$

with the log-likelihood

$$\begin{aligned} &\log \mathbb{P}(c, Y|\theta) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} c_{ik} \log \pi_k + \sum_{1 \le i < j \le n} \sum_{k,l=1}^{K} c_{ik} c_{jl} \left[Y_{ij} \log P_{ij} + (1 - Y_{ij}) \log (1 - P_{ij}) \right]. \end{aligned} \tag{2.30}$$

However, since the latent variable c influences the distribution of network data and the patterns of communities, the dataset in this model is incomplete as we only observe the adjacency matrix Y. The marginal likelihood to describe the distribution of Y is the summation of the complete likelihood function $\mathbb{P}(c, Y|\theta)$ over all possible values of the latent variable c:

$$\mathbb{P}(Y|\theta) = \sum_{c \in [K]^n} \mathbb{P}(c, Y|\theta), \tag{2.31}$$

and the log-likelihood of the incomplete data can be expressed as

$$\log \mathbb{P}(Y|\theta) = \log \sum_{c \in [K]^n} \mathbb{P}(c, Y|\theta). \tag{2.32}$$

Our original interest is to estimate the parameters by maximizing the incomplete likelihood (2.31) or log-likelihood (2.32) equivalently. However, as (2.31) is the marginalization over the discrete latent variable c, it is not tractable especially when the network size is large.

To address the issue, a variety of strategies for estimating latent node assignments and model parameters have been applied, including the Expectation Maximisation (Dempster et al., 1977), Markov Chain Monte Carlo algorithms (Snijders and Nowicki, 1997), variational Expectation Maximization (Airoldi et al., 2008), profile likelihood (Bickel and Chen, 2009), methods of moments (Bickel et al., 2011), belief propagation (Decelle et al., 2011), pseudo-likelihood method (Amini et al., 2013; Wang et al., 2023a), and so on.

2.2.3 Variational Expectation Maximization

In this section, we present the variational Expectation Maximization (VEM), as we will modify it in Chapter 4. We first introduce the Expectation Maximisation (EM) algorithm and VEM separately, then further link the two approaches by interpolating the EM algorithm from the perspective of a variational method.

The EM algorithm (Dempster et al., 1977) is a popular approach to tackle the problem with missing data. Adopted on the SBM, the EM algorithm works on the complete log-likelihood (2.30) directly by taking the latent variable c as the missing data. As indicated by its name, the EM algorithm includes two steps iteratively, the expectation step (E-step) to solve the expected value of the complete log-likelihood given the current parameters and the maximization step (M-step) to update the parameters by maximizing the expected values found on the E-step. More specifically, it iterates the following two steps in the vanilla stochastic block model:

- **E-step**
  Based on the parameter obtained from the step t, $\theta^{(t)} = (\pi^{(t)}, P^{(t)})$, the E-step computes the conditional expected value of the complete log-likelihood (2.30):

$$\begin{aligned}\mathbb{E}[\log \mathbb{P}(c, Y|\theta)|Y, \theta^{(t)}] &= \sum_{i=1}^{n}\sum_{k=1}^{K} \log \pi_k \cdot \mathbb{E}[c_{ik}|Y, \theta^{(t)}] \\ &+ \sum_{1\leq i<j\leq n}\sum_{k,l=1}^{K} \mathbb{E}[c_{ik}c_{jl}|Y, \theta^{(t)}] \cdot [Y_{ij}\log P_{ij} + (1-Y_{ij})\log(1-P_{ij})]\,. \qquad (2.33)\end{aligned}$$

  We can see that there are two types of conditional expected values to calculate in the E-step by using the adjacency matrix Y and the current parameters of the model $\theta^{(t)}$, $\mathbb{E}[c_{ik}|Y, \theta^{(t)}]$ and $\mathbb{E}[c_{ik}z_{jl}|Y, \theta^{(t)}]$.

- **M-step**
  The M-step maximizes (2.33) in the E-step with respect to θ, and the new parameter-estimates are denoted as $\theta^{(t+1)}$,

$$\theta^{(t+1)} = \arg\max_{\theta} \mathbb{E}[\log \mathbb{P}(c, Y|\theta)|Y, \theta^{(t)}], \qquad (2.34)$$

  which will be used to find the expected values of the latent variable in the next E-step.

There are three advantages of employing the EM algorithm to infer the SBM: first, the likelihood function in the EM algorithm will increase with each iteration; second, it will always converge to a local maximum; third, the M-step solutions have a closed form. However, the calculation of $\mathbb{E}[c_{ik}|Y, \theta^{(t)}]$ and $\mathbb{E}[c_{ik}c_{jl}|Y, \theta^{(t)}]$ in the E-step is extremely challenging. For example, to evaluate $\mathbb{E}[c_{ik}|Y, \theta^{(t)}]$, we need to calculate the posterior distribution of c_i given Y and $\theta^{(t)}$, i.e., $\mathbb{P}(c_{ik} = 1|Y, \theta^{(t)})$, which requires the marginalization of the community labels of nodes associated with node i, because c_i's are not conditionally independent as nodes interact with each other through the graphical structure. As a result, calculating the exact conditional expectations is impossible, hence in the E-step, approximation approaches are utilized to solve the problem.

There are mainly two types of approximation methods to estimate the posterior distribution conditioned on the observed data in the E-step. One way is sampling through Markov Chain Monte Carlo (MCMC) algorithms, like Gibbs sampling, which generates a Markov chain of c by iteratively drawing an instance from the distribution of each latent variable c_i conditional on the current values of the other variables (Wei and Tanner, 1990; Xin et al., 2017). The stationary distribution of the Markov chain matches the desired posterior $\mathbb{P}(c_i|Y, \theta^{(t)})$, but the computational cost is high due to the slow mixing of MCMC methods, especially when dealing with a large scale network. To compensate for the shortcomings of MCMC, the other way is to approximate the complicated posterior $\mathbb{P}(c_i|Y, \theta^{(t)})$ by a simple probability distribution $q(c_i)$, which is the so-called VEM, an approximation maximization likelihood strategy based on the variational approach (Jordan et al., 1999; Jaakkola and Jordan, 2000; Daudin et al., 2008).

Now, we describe VEM independently, outside of the framework of the EM algorithm. We reconsider the original target, to estimate the parameter via the maximization of $\log \mathbb{P}(Y|\theta)$ which is intractable. In response, here we first present a lower bound of this log-likelihood of the incomplete data (2.32):

$$\begin{aligned}\log \mathbb{P}(Y|\theta) &= \log \sum_{c \in [K]^n} \frac{\mathbb{P}(c, Y|\theta)}{q(c)} q(c) \\ &\geq \sum_{c \in [K]^n} \left(\log \frac{\mathbb{P}(c, Y|\theta)}{q(c)} \right) \cdot q(c) \\ &= \mathbb{E}_q[\log \frac{\mathbb{P}(c, Y|\theta)}{q(c)}],\end{aligned} \tag{2.35}$$

where $q(c)$ is any distribution of the latent variable c. The inequality comes from the Jensen inequality, as the logarithm is a concave function. The right-hand side of the above inequality (2.35) is known as the evidence lower bound (ELBO) of $\log \mathbb{P}(Y|\theta)$. The difference between the left and right-hand sides of the inequality (2.35),

$$\begin{aligned}\log \mathbb{P}(Y|\theta) - \mathbb{E}_q[\log \frac{\mathbb{P}(c, Y|\theta)}{q(c)}] &= \sum_{c\in[K]^n} \left(\log \mathbb{P}(Y|\theta) - \log \frac{\mathbb{P}(c, Y|\theta)}{q(c)} \right) \cdot q(c) \\ &= \sum_{c\in[K]^n} \left(\log \frac{q(c)}{\mathbb{P}(c|Y, \theta)} \right) \cdot q(c), \qquad (2.36)\end{aligned}$$

happens to be the Kullback–Leibler divergence between the approximation $q(c)$ and the exact posterior $\mathbb{P}(c|Y, \theta)$, denoted as $\text{KL}\,(q(c)||\mathbb{P}(c|Y, \theta))$. The Kullback–Leibler divergence is a measure of how similar the two distributions $q(c)$ and $\mathbb{P}(c|Y, \theta)$ are, which also tells us how much information we lose by using $q(c)$ to approximate $\mathbb{P}(c|Y, \theta)$. It is non-negative, and null only when $q(c) = \mathbb{P}(c|Y, \theta)$.

Identity (2.36) is the well-known variational decomposition (Neal and Hinton, 1998), where the log-likelihood of incomplete data can be decomposed as the sum of ELBO and $\text{KL}\,(q(c)||\mathbb{P}(c|Y, \theta))$. In the implementation, $q(\cdot)$ will be specified as a known family of distribution prior to the iteration of VEM. The procedure of VEM is illustrated as Figure 2.1a, and it iterates between the following variational-expectation step and maximization step:

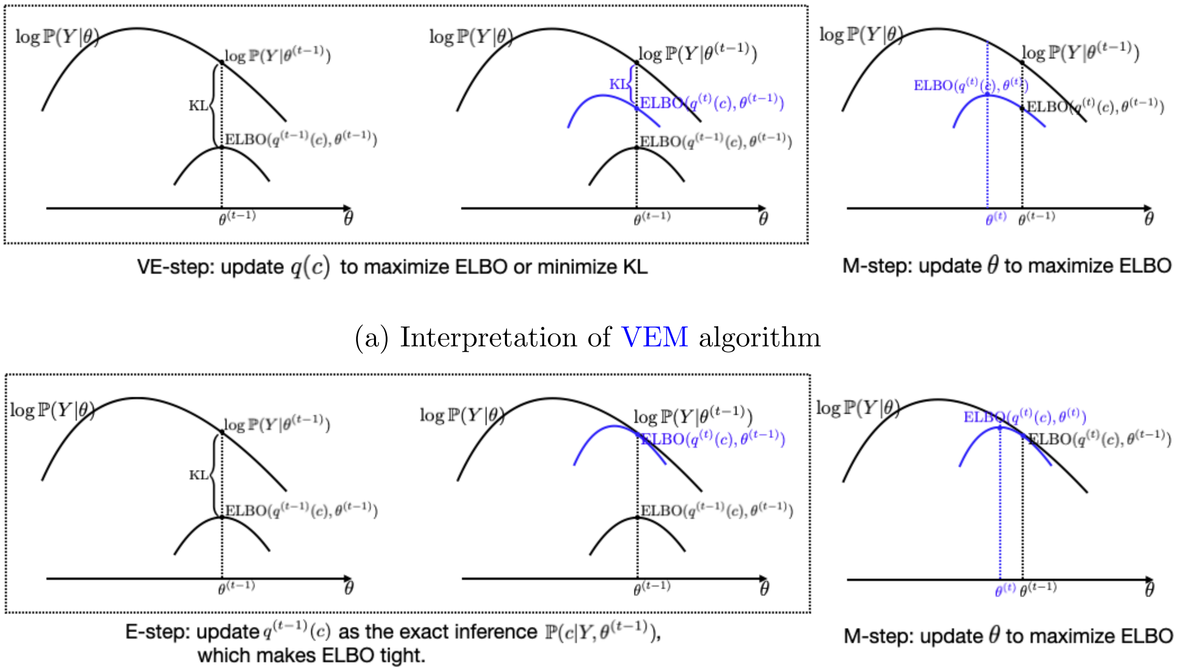
- **VE-step**: With the parameter from the previous step, $\theta^{(t-1)}$, the variational decomposition is

  $$\log \mathbb{P}(Y|\theta^{(t-1)}) = \text{ELBO}(q^{(t-1)}(c), \theta^{(t-1)}) + \text{KL}\left(q^{(t-1)}(c)||\mathbb{P}(c|Y, \theta^{(t-1)})\right). \qquad (2.37)$$

  Given an observed network Y, the left-hand side of the above identity is a constant. The target at VE-step is to find an optimal $q(c)$ within the pre-specified family which is closest to the exact posterior $\mathbb{P}(c|Y, \theta^{(t-1)})$. Since $\text{KL}\left(q(c)||\mathbb{P}(c|Y, \theta^{(t-1)})\right)$ can not be minimized directly, instead we maximize ELBO as an equivalent objective as $\log \mathbb{P}(Y|\theta^{(t-1)})$ is constant with respect to $q(c)$.

- **M-step**: After $q(c)$ is updated in the VE-step, we maximize $\text{ELBO}(q^{(t)}(c), \theta)$ with respect to θ to obtain the optimal $\theta^{(t)}$.

The selection of the variational family for $q(c)$ is significant, as the complicated variational family will bring computational difficulties while the over-simplified family will underestimate the posterior. For computational efficiency, we use the mean-field approximation on $q(c)$ to approximate the true posterior distribution $\mathbb{P}(c|Y, \theta)$ by assuming the


$\log \mathbb{P}(Y|\theta)$
$\log \mathbb{P}(Y|\theta^{(t-1)})$
KL
$\mathrm{ELBO}(q^{(t-1)}(c), \theta^{(t-1)})$
$\mathrm{ELBO}(q^{(t)}(c), \theta^{(t-1)})$
$\mathrm{ELBO}(q^{(t)}(c), \theta^{(t)})$
$\theta^{(t-1)}$
$\theta^{(t)}$
θ
VE-step: update $q(c)$ to maximize ELBO or minimize KL
M-step: update θ to maximize ELBO
E-step: update $q^{(t-1)}(c)$ as the exact inference $\mathbb{P}(c|Y, \theta^{(t-1)})$, which makes ELBO tight.
M-step: update θ to maximize ELBO

(a) Interpretation of VEM algorithm

(b) Variational interpretation of EM algorithm

Figure 2.1: Interpretations of VEM and EM algorithms

independence between latent variables c, so that $q(c)$ can be factorized over c as

$$q(c) = \prod_{i=1}^{n} q(c_i) = \prod_{i=1}^{n} \prod_{k=1}^{K} \tau_{ik}, \tag{2.38}$$

where each c_i follows a multinomial distribution with parameters $(\tau_{i1}, \cdots, \tau_{iK})$. The ELBO takes the form of

$$
\begin{aligned}
ELBO =& \mathbb{E}_q[\log \mathbb{P}(c, Y|\theta)] - \mathbb{E}_q[\log q(c)] \\
=& \sum_{i=1}^{n} \sum_{k=1}^{K} \log \pi_k \cdot \mathbb{E}_q[c_{ik}] \\
& + \sum_{1 \le i < j \le n} \sum_{k,l=1}^{K} \mathbb{E}_q[c_{ik} c_{jl}] \cdot [Y_{ij} \log P_{ij} + (1 - Y_{ij}) \log (1 - P_{ij})] \\
& - \mathbb{E}_q[\log q(c)] \\
=& \sum_{i=1}^{n} \sum_{k=1}^{K} \log \pi_k \cdot \tau_{ik} + \sum_{1 \le i < j \le n} \sum_{k,l=1}^{K} \tau_{ik} \tau_{jl} \cdot [Y_{ij} \log P_{ij} + (1 - Y_{ij}) \log (1 - P_{ij})] \\
& - \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik} \log \tau_{ik}.
\end{aligned} \tag{2.39}
$$

We can see that the above ELBO no longer requires the marginal probability. The specific VEM algorithm in SBM involves the two iterative steps to maximize ELBO with respect to the model parameters θ as well as those of the variational parameters τ:

- **VE-step**
  Variational Expectation step aims to find the optimal τ's that maximize the ELBO (2.39) under the constraint that $\sum_{k=1}^{K} \tau_{ik} = 1, \; i = 1, \cdots, n$ based on the parameter obtained from the step t, $\theta^{(t)} = (\pi^{(t)}, C^{(t)})$:

$$
\begin{aligned}
\tau^{(t+1)} =& \arg\max_{\tau} \sum_{i=1}^{n} \sum_{k=1}^{K} \log \pi_k^{(t)} \cdot \tau_{ik} - \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik} \log \tau_{ik} \\
& + \sum_{1 \le i < j \le n} \sum_{k,l=1}^{K} \tau_{ik} \tau_{jl} \cdot \left[Y_{ij} \log P_{ij}^{(t)} + (1 - Y_{ij}) \log \left(1 - P_{ij}^{(t)} \right) \right], \\
& \text{subject to } \sum_{k=1}^{K} \tau_{ik} = 1, \; i = 1, \cdots, n.
\end{aligned} \tag{2.40}
$$

- **M-step**
  The M-step is to find $\theta^{(t+1)}$ by maximizing (2.39) with the updated τ found in the

VE-step:

$$\theta^{(t+1)} = \arg\max_{\pi,P} \sum_{i=1}^{n}\sum_{k=1}^{K} \log \pi_k \cdot \tau_{ik}^{(t+1)} + \sum_{1\le i<j\le n}\sum_{k,l=1}^{K} \tau_{ik}^{(t+1)}\tau_{jl}^{(t+1)} \cdot \left[Y_{ij}\log P_{ij} + (1-Y_{ij})\log(1-P_{ij})\right], \quad \text{subject to } \sum_{k=1}^{K}\pi_k = 1, \tag{2.41}$$

which will be used to find the expected values in the next VE-step.

Both steps can be easily updated via their closed forms solved by Lagrangian method.

Conversely, the EM algorithm can also be understood as a variational method as demonstrated in Figure 2.1b. At the t^{th} step, $q^{(t)}(c)$ is the exact posterior $\mathbb{P}(c|Y, \theta^{(t)})$, which makes the Kullback–Leibler divergence null. Accordingly, the ELBO is tight, which equals $\log \mathbb{P}(Y|\theta^{(t-1)})$.

Chapter 3

Two Gaussian Regularization Methods for Time-varying Networks

3.1 Introduction

In many applications, we need to identify and estimate associations and interactions among a set of random variables to uncover their latent topological structures, such as protein-protein interaction networks (Sato et al., 2006), ecological networks in the microbial interactome (Dohlman and Shen, 2019), and gene regulatory networks (Emmert-Streib et al., 2012). Modeling an undirected static network has been studied since the last century (see e.g. Whittaker, 1990; Lauritzen, 1996). More often than not, however, network structures evolve over time in response to both endogenous and exogenous factors; therefore, the assumption of the relational structure being fixed is too restrictive. Two notable examples illustrating dynamic evolution are the brain network and the climate network. In the brain network, neural interactions continually shift in response to the brain's motion. Meanwhile, the climate network captures spatiotemporal patterns influenced by a mix of climatic and non-climatic factors. For in-depth insights into these networks and their dynamic characteristics, refer to the detailed descriptions in Chapter 2.1.1. This has catalyzed emerging interest in estimating time-varying networks where the data consist of serial snapshots of the networks that evolve over time.

3.1.1 Static Networks

There is rich literature on estimating a static network (see e.g. Meinshausen and Bühlmann, 2006; Yuan and Lin, 2007; Friedman et al., 2008; Peng et al., 2009), among which the GGM is particularly useful. For a comprehensive understanding of the static GGM, please refer to Section 2.1.2. Here, we present a brief recap of the fundamental concepts.

Consider a p-dimensional GGM $(x_1, \cdots, x_p)^\top \sim N(\mathbf{0}, \mathbf{\Sigma})$. Denote the precision matrix as $\mathbf{\Sigma}^{-1} = (\sigma^{ij})_{p\times p}$. Under multivariate normality, zero σ^{ij} indicates conditional independence between x_i and x_j given the remaining nodes. Therefore, a network can be encoded by conditional dependencies in a GGM, where nodes represent random variables and the edge connecting the ith and jth nodes is decided by whether σ^{ij} is zero. There are mainly two different ways to model a static GGM: the l_1-penalized Gaussian likelihood method, and the penalized regression method.

The former aims to identify zero elements in the precision matrix by maximizing an l_1-penalized multivariate Gaussian log-likelihood, $\log\det \mathbf{\Sigma}^{-1} - \operatorname{tr}(\boldsymbol{S}\mathbf{\Sigma}^{-1}) - \lambda_1 \|\mathbf{\Sigma}^{-1}\|_1$, where $\boldsymbol{S}$ denotes the sample covariance matrix. We will simply refer to this approach by perhaps the most widely used algorithm associated with it—namely, GLASSO, an abbreviation of "graphical LASSO" (Friedman et al., 2008).

The latter, which we refer to as the regression method, uses penalized node-wise regression to estimate partial correlations (Meinshausen and Bühlmann, 2006). The partial correlation between x_i and x_j, denoted as ρ_{ij}, measures the degree of association between the two random variables, with the effect of other random variables removed. Specifically, the partial correlation ρ_{ij} can be expressed as $-\sigma^{ij}/\sqrt{\sigma^{ii}\sigma^{jj}}$, and estimated by performing a multiple regression $x_i = \sum_{j\neq i} \beta_{ij}x_j + \epsilon_i$ sequentially for $i = 1, \cdots, p$, where the error variance $Var(\epsilon_i) = 1/\sigma^{ii}$, and the regression coefficient $\beta_{ij} = -\sigma^{ij}/\sigma^{ii} = \rho_{ij}\sqrt{\sigma^{jj}/\sigma^{ii}}$. Accordingly, estimating the partial correlations as regression coefficients can characterize the graphical structures. However, the general regression estimators are never exactly zero due to the high-dimension-low-sample-size setting of the problem and the sampling variation in the data. To obtain a sparse network and make the regression problem well-posed, regularization techniques are employed (Meinshausen and Bühlmann, 2006). Peng et al. (2009) proposed an efficient modification, referred to as the Sparse PArtial Correlation Estimation or *SPACE*, to solve this regularized regression problem. Suppose that $\boldsymbol{X}_i$ denotes the ith column of $\boldsymbol{X}_{n\times p}$ that consists of n i.i.d. observations from the GGM, $\boldsymbol{\sigma}_{p\times 1} = (\sigma^{11}, \cdots, \sigma^{pp})^\top$, and $\boldsymbol{\theta}_{p(p-1)/2\times 1} = (\rho_{12}, \cdots, \rho_{p-1,p})^\top$. *SPACE* estimates the partial correlations by minimizing a penalized likelihood function with an l_1 penalty:

$$L_{LASSO}(\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{\sigma}, n, \lambda_1) = \frac{1}{n}\sum_{i=1}^{p} \left\| \boldsymbol{X}_i - \sum_{j<i} \rho_{ji}\sqrt{\frac{\sigma^{jj}}{\sigma^{ii}}}\boldsymbol{X}_j - \sum_{j>i} \rho_{ij}\sqrt{\frac{\sigma^{jj}}{\sigma^{ii}}}\boldsymbol{X}_j \right\|^2 + \lambda_1 \left\|\boldsymbol{\theta}\right\|_1, \tag{3.1}$$

where λ_1 is a prespecified regularization parameter to control the strength of shrinkage and variable selection of $\boldsymbol{\theta}$. By stacking all columns in $\boldsymbol{X}$ into a long response vector $\boldsymbol{Y}$ and filling nonzero blocks in an $np \times p(p-1)/2$ sparse matrix $\tilde{\boldsymbol{X}}$ by $\sqrt{\sigma^{jj}/\sigma^{ii}}\boldsymbol{X}_j$, (3.1) is converted to a standard LASSO problem

$$L_{LASSO}(\boldsymbol{Y}, \tilde{\boldsymbol{X}}, \boldsymbol{\theta}, n, \lambda_1) = \frac{1}{n}\|\boldsymbol{Y} - \tilde{\boldsymbol{X}}\boldsymbol{\theta}\|^2 + \lambda_1 \left\|\boldsymbol{\theta}\right\|_1, \tag{3.2}$$

which can be solved efficiently.

The two approaches are clearly not the same, but they are related—for example, it can be shown that, under some circumstances, the regression approach is also consistent whenever the GLASSO is consistent (Meinshausen, 2008; Drton and Maathuis, 2017). The regression method can have some advantages, though. For instance, when we are only interested in the conditional dependence of p_1 random variables out of p random variables, GLASSO still needs to estimate all $p \times p$ entries in the precision matrix whereas the number of parameters to be estimated in regression methods is considerably smaller—roughly $p \times p_1$; see Meinshausen and Bühlmann (2006) and Peng et al. (2009). Moreover, the regression method is also perhaps more easily generalized to non-Gaussian data (e.g., Voorman et al., 2014; Yang et al., 2015a; Chen et al., 2015; Loh, 2017; Chen and Yi, 2021). Finally, partial correlations are more directly meaningful and interpretable parameters for the environmental sciences, as they not only characterize the graphical structure of a complex meteorological system but also measure the degree of association among random variables corrected for confounders and covariates.

Another interesting regression approach was proposed recently by Kang et al. (2020), who observed that a Cholesky decomposition of the covariance matrix would imply a different set of regression equations—x_2 onto x_1, x_3 onto (x_1, x_2), and so on—which could also be used to estimate the covariance matrix. Here, because the sequence in which these regressions are carried out matters, a clever ensemble approach was introduced to do this several times by permuting the ordering of the variables, and aggregating the results. In a follow-up study (Wang et al., 2023b), they also adopted more robust loss functions in this framework to handle data with heavy tails and outliers. Although such extensions are possible for (3.2) as well, we do not pursue it in this thesis. These different regression approaches are fundamentally similar in spirit, but the regression estimates from the Cholesky-implied equations do not have a direct scientific interpretation as partial correlations do, especially after consolidation from different permutations of the regression sequence.

3.1.2 Dynamic Networks

In contrast, the literature concerning dynamic networks have appeared only more recently. Bartlett et al. (2021), for instance, take a Bayesian approach to separate two types of sparsity—sparsity across time and sparsity across variables—when modeling time-varying networks. By and large, however, most methods simply extend the static GGM to dynamic networks; moreover, they predominantly adopt the GLASSO approach by imposing a penalty term on the difference between precision matrices at neighboring time points to achieve smoothness in estimated dynamic networks; see Zhou et al. (2010), Ahmed and Xing (2009), Kolar et al. (2010), Kolar and Xing (2011), Danaher et al. (2014), Monti et al. (2014), Gibberd and Nelson (2017), Hallac et al. (2017) and references therein.

Relatively speaking, limited work has been done to extend the regression approach to time-varying networks. Here, the approach based on Cholesky factorization (Kang et al., 2020) has the advantage that the original variables can be viewed as different combinations of a few independent factors, and one can argue that it suffices to simply model the marginal variances of these factors independently as functions of time, e.g., with GARCH-type models that are popular for time series. Such shortcuts are not easily amendable to the approach based on partial correlations (Meinshausen and Bühlmann, 2006; Peng et al., 2009), however.

In this article, we tackle precisely this problem, that is, estimating partial correlations $\rho_{ij}(t)$ that change over time. Specifically, we develop statistical methods to identify the associations and their dynamic changes in discrete time-varying networks based on *SPACE*. This goal is achieved under the assumption that the changes in the temporal network from one time point to the next are smooth, which encourages the regularization on the difference of partial correlations between adjacent time points. The regularization techniques we use in this work include both l_1 (Tibshirani, 1996; Tibshirani et al., 2005) and l_2 regularization (Zou and Hastie, 2005), leading to two different algorithms. Although Xue et al. (2020) also work with partial correlations, our work differs from theirs. They approximate $\rho_{ij}(t)$ using a linear combination of B-spline basis functions. Since B-spline basis functions have local support, sparse networks are obtained by imposing a group LASSO penalty on the coefficient vectors in the spline approximation. However, when $\rho_{ij}(t)$ is treated as a function of time, it will usually require that we observe the function at many time points in order for us to estimate it; whereas our approach does not suffer from this limitation.

3.1.3 Outline

We structure the remainder of this chapter as follows. In Section 3.2, we present our time-varying network models with two different penalties on temporal dissimilarity. In Section 3.3, we describe high-level computational details. Our main contributions are: first, we use the alternating direction method of multipliers namely ADMM to develop computationally efficient algorithms by parameterizing l_1 and l_2 penalties differently (Section 3.3.1); second, by generalizing existing results in the literature, we derive approximate degree-of-freedom formulae to characterize the effective complexity of our solutions and to facilitate the selection of tuning parameters (Section 3.3.2); third, when implementing the ADMM iterations and computing the degrees of freedom, we propose efficient methods to invert some large matrices. In Section 3.4, we illustrate the performance of our methods through a simulation study. In Section 3.5.1, our methods are applied to two datasets, including Canadian temperature data for major cities in July and fMRI data from both healthy individuals and those with attention deficit hyperactivity disorder (ADHD). Notable findings underscore dynamic characteristics in both climate and brain networks. Section 3.6 concludes the article. Some technical details and additional results of numerical studies are delegated to the supplementary materials.

3.2 Methodology

We consider a time-varying GGM defined on a set of T equidistant discrete time points indexed by $\{t_1, \cdots, t_T\}$: $(x_1(t_k), \cdots, x_p(t_k))^\top \sim N(\boldsymbol{\mu}(t_k), \boldsymbol{\Sigma}(t_k)),\ k \in \{1, \cdots, T\}$. Without loss of generality, we assume $\boldsymbol{\mu}(t_k) = \mathbf{0}$, which can be achieved in practice by centering the data set at each time point. The notation in Section 3.1 is inherited at every discrete time point. Then we have T temporal datasets $\boldsymbol{X}(t_1), \cdots, \boldsymbol{X}(t_T)$ and diagonals in temporal precision matrices $\boldsymbol{\sigma}^\top(t_1), \cdots, \boldsymbol{\sigma}^\top(t_T)$ which, by stacking, leads to a vector $\boldsymbol{\sigma}$ with length pT. We have the temporal response vector $\boldsymbol{Y}(t_k)$ and temporal predictor matrix $\tilde{\boldsymbol{X}}(t_k)$ formed as in Section 3.1. A vector $\boldsymbol{\mathcal{Y}}$ of length npT is formed by stacking all temporal reponse vector $\boldsymbol{Y}(t_k)$. Let $\boldsymbol{\mathcal{X}}$ denote a $Tnp \times Tp(p-1)/2$ block diagonal matrix, where each diagonal block is the temporal predictor matrix $\tilde{\boldsymbol{X}}(t_k),\ k \in \{1, \cdots, T\}$. Our objective is to estimate a vector $\boldsymbol{\theta}$ of length $Tp(p-1)/2$ composed of all temporal partial correlations, i.e., $\boldsymbol{\theta} = (\boldsymbol{\theta}^\top(t_1), \cdots, \boldsymbol{\theta}^\top(t_T))^\top$. Throughout the rest of the chapter, we use $\boldsymbol{\theta}$ and $\boldsymbol{\sigma}$ to denote these two long vectors containing temporal parameters over all time points.

Naïvely, one can minimize (3.2) at each time point independently to estimate the temporal partial correlations, with a loss function that can be written in matrix form as

$$\mathcal{L}_{TVN\_LASSO}(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{X}}, \boldsymbol{\theta}, n, \lambda_1) = \sum_{k=1}^{T} L_{LASSO}(\boldsymbol{Y}(t_k), \tilde{\boldsymbol{X}}(t_k), \boldsymbol{\theta}, n, \lambda_1) = \frac{1}{n}\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}}\boldsymbol{\theta}\|^2 + \lambda_1\|\boldsymbol{\theta}\|_1. \tag{3.3}$$

But it is natural to assume that the covariance matrix $\boldsymbol{\Sigma}(t)$ are element-wise smooth over t. Then, by Cramer's rule, the entries in the precision matrix and thus the corresponding partial correlations should also be smooth over t. Therefore, we propose a regularization method with an extra penalty term $\lambda_2 \cdot P(\boldsymbol{\theta})$ to encourage the partial correlations to be similar for adjacent time points. The objective function $\mathcal{L}_{TVN}$ of our time-varying network problem is

$$\mathcal{L}_{TVN}(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{X}}, \boldsymbol{\theta}, n, \lambda_1, \lambda_2) = \mathcal{L}_{TVN\_LASSO}(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{X}}, \boldsymbol{\theta}, n, \lambda_1) + \lambda_2 \cdot P(\boldsymbol{\theta}). \tag{3.4}$$

Here $P(\boldsymbol{\theta})$ denotes a penalty function measuring the total distance between neighbouring coefficients and λ_2 is another tuning parameter. We consider two different penalty functions for $P(\boldsymbol{\theta})$.

Generalized elastic net (GEN) Our first penalty generalizes the work of Zou and Hastie (2005). To achieve smoothness of partial correlations over time, the GEN applies l_2 penalties to the differences of partial correlations along the time sequences in $P(\boldsymbol{\theta})$, taking the form

$$P(\boldsymbol{\theta}) = \sum_{k=2}^{T} \sum_{1 \le i < j \le p} \{\rho_{ij}(t_k) - \rho_{ij}(t_{k-1})\}^2. \tag{3.5}$$

Generalized fused LASSO (GFL) Our second penalty generalizes the work of Tibshirani et al. (2005) by penalizing the absolute difference of the partial correlations at adjacent time points. In particular, the penalty function takes the following form:

$$P(\boldsymbol{\theta}) = \sum_{k=2}^{T} \sum_{1 \le i < j \le p} |\rho_{ij}(t_k) - \rho_{ij}(t_{k-1})|. \tag{3.6}$$

A large tuning parameter λ_2 in the GFL not only yields smoothness in the changes between neighboring coefficients, but also shrinks some of those changes to be exactly zero.

When $\lambda_2 = 0$, both the GEN and the GFL are reduced to the naïve LASSO problem (3.3), where there is no regularization on the changes in coefficients at adjacent time points. The key difference between the two penalties is that the GFL is able to force the partial correlations at adjacent time points to be identical if their difference is sufficiently small, while the GEN cannot.

3.3 Algorithm for the Time-varying Network Estimation

In this section, we describe high-level computational details, while detailed technicalities are described in the supplementary materials.

The loss function (3.4) with GEN (3.5) and GFL (3.6) penalties can be respectively written as:

$$\mathcal{L}_{GEN}(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{X}}, \boldsymbol{\theta}, n, \lambda_1, \lambda_2) = \frac{1}{n}\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}}\boldsymbol{\theta}\|^2 + \lambda_1\|\boldsymbol{\theta}\|_1 + \lambda_2\|\boldsymbol{D}\boldsymbol{\theta}\|^2 \quad \text{and} \tag{3.7}$$

$$\mathcal{L}_{GFL}(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{X}}, \boldsymbol{\theta}, n, \lambda_1, \lambda_2) = \frac{1}{n}\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}}\boldsymbol{\theta}\|^2 + \lambda_1\|\boldsymbol{\theta}\|_1 + \lambda_2\|\boldsymbol{D}\boldsymbol{\theta}\|_1, \tag{3.8}$$

where $\boldsymbol{D}$ is a block difference matrix composed of many $p(p-1)/2$−by−$p(p-1)/2$ square matrices:

$$\boldsymbol{D}_{(T-1)p(p-1)/2 \times Tp(p-1)/2} = \begin{bmatrix} I & -I & 0 & 0 & 0 \\ 0 & I & -I & 0 & 0 \\ & & \ddots & \ddots & \\ 0 & 0 & 0 & I & -I \end{bmatrix}. \tag{3.9}$$

If $\boldsymbol{\mathcal{X}}$ is completely known, the minimizations of (3.7) and (3.8) over $\boldsymbol{\theta}$ are standard GEN and GFL optimization problems. However, the predictor matrix $\boldsymbol{\mathcal{X}}$ involves an unknown parameter $\boldsymbol{\sigma}$, which requires us to leverage coordinate descent techniques and update $\boldsymbol{\sigma}$ and $\boldsymbol{\theta}$ iteratively. We extend the two-step iterative procedure developed by Peng et al. (2009). The detailed algorithm is summarized in Algorithm 1, where Steps 2 and 3 are iterated to update $\boldsymbol{\sigma}$ and $\boldsymbol{\theta}$ until convergence. Given $\boldsymbol{\sigma}$, both minimization problems—of (3.7) and of (3.8)—are convex; details are given in Section 3.3.1.

Algorithm 1 Two-step iterative procedure

```
Input: The centered original data $\{\boldsymbol{X}\}$.
Output: Estimated $\boldsymbol{\sigma}$ and $\boldsymbol{\theta}$
Initialization:
Start with the initial estimate $(\sigma^{ii})^{(0)}(t_k) = 1/\hat{\sigma}_{ii}(t_k)$, where $\hat{\sigma}_{ii}(t_k) = (n-1)^{-1}\sum_{j=1}^{n}[x_i^j(t_k) - \bar{x}_i(t_k)]^2$, $x_i^j(t_k)$ denotes the observation of the $i$th node made at time $t_j$ for the $j$th subject, and $\bar{x}_i(t_k)$ denote the average of these observations across $n$ subjects. Form the initial predictor matrix $\boldsymbol{\mathcal{X}}^{(0)}$ with the data and $\boldsymbol{\sigma}^{(0)}$.
while $\|\boldsymbol{\sigma}^{(l)} - \boldsymbol{\sigma}^{(l-1)}\|_2 > \tau_1$ and $\|\boldsymbol{\theta}^{(l)} - \boldsymbol{\theta}^{(l-1)}\|_2 > \tau_2$ do
    Estimate $\boldsymbol{\theta}^{(l+1)}$ by solving (3.7) or (3.8) with the given $\boldsymbol{\sigma}^{(l)}$.
    Update $\boldsymbol{\sigma}^{(l+1)}$, where $1/\hat{\sigma}^{ii}(t_k) = n^{-1}\|X_i(t_k) - \sum_{j\neq i}\hat{\beta}_{ij}(t_k)X_j(t_k)\|^2$ and $\hat{\beta}_{ij}(t_k) = (\rho_{ij})^{(l+1)}(t_k)\sqrt{(\sigma^{jj})^{(l)}(t_k)/(\sigma^{ii})^{(l)}(t_k)}$.
    Update the predictor matrix $\boldsymbol{\mathcal{X}}^{(l+1)}$ with the data and $\boldsymbol{\sigma}^{(l+1)}$.
end while
```

3.3.1 Fast ADMM Algorithms for GEN and GFL

Given $\boldsymbol{\sigma}$, we minimize (3.7) and (3.8) using the ADMM. The ADMM algorithm and its convergence properties are comprehensively illustrated in Boyd et al. (2011). For an in-depth exploration of ADMM, please refer to Section 2.1.4. In this section, our attention is directed towards customizing ADMM within the specific context of our study. The key trick to use the technique in our context is that, by adding a new constraint $\boldsymbol{\theta} - \boldsymbol{z} = \boldsymbol{0}$, we can freely re-express our objective functions (3.7) and (3.8) in either of the following ways,

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z}) = \begin{cases} \frac{1}{n}\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}}\boldsymbol{\theta}\|^2 + \lambda_1\|\boldsymbol{z}\|_1 + \lambda_2 P(\boldsymbol{\theta}), \\ \frac{1}{n}\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}}\boldsymbol{\theta}\|^2 + \lambda_1\|\boldsymbol{z}\|_1 + \lambda_2 P(\boldsymbol{z}), \end{cases}$$

depending on the specific form of the penalty function $P(\cdot)$.

Define $\boldsymbol{u} \in \mathbb{R}^{Tp(p-1)/2\times 1}$ as the dual variable, and let $a \in \mathbb{R}^+$ be a penalty parameter. The augmented Lagrangian L_a for minimizing $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z})$, subject to $\boldsymbol{\theta} - \boldsymbol{z} = \boldsymbol{0}$, is

$$L_a(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{u}) = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z}) + a \cdot \boldsymbol{u}^\top(\boldsymbol{\theta} - \boldsymbol{z}) + \frac{a}{2}\|\boldsymbol{\theta} - \boldsymbol{z}\|^2, \tag{3.10}$$

where we have scaled the dual variable $\boldsymbol{u}$ by a itself, so that the ADMM algorithm iterates over the following three steps:

$$\text{(i) } \boldsymbol{\theta}_{(l)} = \arg\min_{\boldsymbol{\theta}} L_a(\boldsymbol{\theta}, \boldsymbol{z}_{(l-1)}, \boldsymbol{u}_{(l-1)}) \tag{3.11}$$

$$\text{(ii) } \boldsymbol{z}_{(l)} = \arg\min_{\boldsymbol{z}} L_a(\boldsymbol{\theta}_{(l)}, \boldsymbol{z}, \boldsymbol{u}_{(l-1)}) \tag{3.12}$$

$$\text{(iii) } \boldsymbol{u}_{(l)} = \boldsymbol{u}_{(l-1)} + \boldsymbol{\theta}_{(l)} - \boldsymbol{z}_{(l)} \tag{3.13}$$

over $l = 0, 1, 2, \ldots$ until convergence, with typical initialization $\boldsymbol{\theta}_{(0)} = \mathbf{0}$, $\boldsymbol{z}_{(0)} = \mathbf{0}$ and $\boldsymbol{u}_{(0)} = \mathbf{0}$.

The separation of the underlying optimization problem into two subproblems—namely, (3.11) and (3.12)—allows us to obtain the key insight that, for the GEN penality $P(\boldsymbol{\theta}) = \|\boldsymbol{D\theta}\|^2$, it is more advantageous to parameterize the penalty as $P(\boldsymbol{\theta})$; whereas, for the GFL penalty $P(\boldsymbol{\theta}) = \|\boldsymbol{D\theta}\|_1$, it is more advantageous to parameterize it as $P(\boldsymbol{z})$. More details are presented below, where, for clarity, we shall suppress the step index l in all formulae.

ADMM for GEN

As stated above, for the GEN penalty we parameterize it as $P(\boldsymbol{\theta})$ in the ADMM iterations, so that (3.11) merely minimizes over a quadratic function of $\boldsymbol{\theta}$,

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \frac{1}{n}\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}\theta}\|^2 + \lambda_2\|\boldsymbol{D\theta}\|^2 + a \cdot \boldsymbol{u}^\top(\boldsymbol{\theta} - \boldsymbol{z}) + \frac{a}{2}\|\boldsymbol{\theta} - \boldsymbol{z}\|^2, \tag{3.14}$$

which has closed form solution,

$$\boldsymbol{\theta} = \left(\frac{2}{n}\boldsymbol{\mathcal{X}}^\top\boldsymbol{\mathcal{X}} + 2\lambda_2\boldsymbol{D}^\top\boldsymbol{D} + aI\right)^{-1}\left\{\frac{2}{n}\boldsymbol{\mathcal{Y}}^\top\boldsymbol{\mathcal{X}} + a(\boldsymbol{z} - \boldsymbol{u})\right\}. \tag{3.15}$$

While this may appear easy, it is worth emphasizing that, for us, the matrix that must be inverted in (3.15) can be very large. Fortunately, the inversion can be pre-calculated outside the ADMM iterations, for it remains constant from one iteration to another. Furthermore, (3.15) can be computed efficiently by exploiting the fact that $\left(2\boldsymbol{X}^\top\boldsymbol{X}/n + 2\lambda_2\boldsymbol{D}^\top\boldsymbol{D} + aI\right)$ is a symmetric block tri-diagonal matrix whose off-diagonal blocks are $-2\lambda_2 I$. Technical details for efficiently inverting such a matrix are given in A.1 in the supplementary materials.

The minimization (3.12) over $\boldsymbol{z}$,

$$\boldsymbol{z} = \arg\min_{\boldsymbol{z}} \lambda_1 \|\boldsymbol{z}\|_1 + a \cdot \boldsymbol{u}^\top(\boldsymbol{\theta} - \boldsymbol{z}) + \frac{a}{2}\|\boldsymbol{\theta} - \boldsymbol{z}\|^2,$$

is simply a LASSO-type problem. As $\|\boldsymbol{z}\|_1$ is not differentiable everywhere, we leverage its sub-differential and obtain the solution as

$$\boldsymbol{z} = \begin{cases} \boldsymbol{u} + \boldsymbol{\theta} - \dfrac{\lambda_1}{a}, & \text{if } \boldsymbol{u} + \boldsymbol{\theta} > \dfrac{\lambda_1}{a}, \\ \boldsymbol{u} + \boldsymbol{\theta} + \dfrac{\lambda_1}{a}, & \text{if } \boldsymbol{u} + \boldsymbol{\theta} < -\dfrac{\lambda_1}{a}, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{3.16}$$

ADMM for GFL

The GFL problem is in itself important for a wide range of scientific procedures including signal processing and machine learning, especially when the matrix $\boldsymbol{D}$ in (3.8) takes on more general forms. Even though (3.8) is convex and there exists a global optimal solution, minimizing it is still computationally challenging. A large body of literature exists on solving the GFL problem (e.g., Tibshirani and Taylor, 2011; Ye and Xie, 2011; Xin et al., 2016), but many methods still suffer from high computational cost or have difficulties in achieving sparsity in both $\boldsymbol{\theta}$ and $\boldsymbol{D\theta}$ simultaneously. To get around these bottlenecks, we design a specific ADMM algorithm by exploiting the special block structure in our problem.

Again, as stated earlier, for the GFL penalty we parameterize it as $P(\boldsymbol{z})$ in the ADMM iterations, so (3.11) still merely minimizes over a quadratic function of $\boldsymbol{\theta}$,

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \frac{1}{n}\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}}\boldsymbol{\theta}\|^2 + a \cdot \boldsymbol{u}^\top(\boldsymbol{\theta} - \boldsymbol{z}) + \frac{a}{2}\|\boldsymbol{\theta} - \boldsymbol{z}\|^2,$$

with the closed-form solution,

$$\boldsymbol{\theta} = \left(\frac{2}{n}\boldsymbol{\mathcal{X}}^\top\boldsymbol{\mathcal{X}} + aI\right)^{-1}\left\{\frac{2}{n}\boldsymbol{\mathcal{Y}}^\top\boldsymbol{\mathcal{X}} + a(\boldsymbol{z} - \boldsymbol{u})\right\}. \tag{3.17}$$

The minimization (3.12) over $\boldsymbol{z}$ now becomes

$$\begin{aligned}\boldsymbol{z} &= \underset{\boldsymbol{z}}{arg\,min}\ \lambda_1\|\boldsymbol{z}\|_1+\lambda_2\|\boldsymbol{D}\boldsymbol{z}\|_1+a\cdot\boldsymbol{u}^\top(\boldsymbol{\theta}-\boldsymbol{z})+\frac{a}{2}\|\boldsymbol{\theta}-\boldsymbol{z}\|^2. \\ &= \underset{\boldsymbol{z}}{arg\,min}\ \frac{1}{2}\|\boldsymbol{\theta}+\boldsymbol{u}-\boldsymbol{z}\|^2+\frac{\lambda_1}{a}\|\boldsymbol{z}\|_1+\frac{\lambda_2}{a}\|\boldsymbol{D}\boldsymbol{z}\|_1. \end{aligned} \tag{3.18}$$

Let $\boldsymbol{z}(t_k) = (z_{12}(t_k), \cdots, z_{p-1,p}(t_k))^\top$, and $\boldsymbol{u}$ be defined in a similar way. Then (3.18) can be decomposed into $p(p-1)/2$ independent optimization problems,

$$\begin{aligned}\{z_{ij}(t_k)\}_{k=1}^T = \underset{\{z_{ij}(t_k)\}_{k=1}^T}{arg\,min}\ &\sum_{k=1}^T [z_{ij}(t_k)-\rho_{ij}(t_k)-u_{ij}(t_k)]^2 + \\ &\frac{\lambda_1}{a}\sum_{k=1}^T|z_{ij}(t_k)|+\frac{\lambda_2}{a}\sum_{k=2}^T|z_{ij}(t_k)-z_{ij}(t_{k-1})|, \quad 1\le i<j\le p,\end{aligned} \tag{3.19}$$

a collection of fused LASSO signal approximator (FLSA) problems, and we use the algorithm in Hoefling (2010) to solve them.

3.3.2 Tuning Parameter Selection

Readers may notice that the updating equations for $\boldsymbol{\theta}$—specifically, (3.15) and (3.17)—do not produce sparse solutions, but the key ADMM constraint $\boldsymbol{\theta}-\boldsymbol{z}=0$ means that we can simply take $\boldsymbol{z}$ to be the final solution, and the updating equations for $\boldsymbol{z}$—in particular, (3.16) and (3.19)—do indeed produce sparse solutions. We now discuss how to choose the tuning parameters (λ_1, λ_2).

We mainly adopt the Bayesian Information Criterion (BIC),

$$BIC(\lambda_1,\lambda_2) = n\times\sum_{k=1}^T\left[-\log|\hat{\boldsymbol{\Sigma}}^{-1}(t_k)|+\mathrm{tr}\left(\hat{\boldsymbol{\Sigma}}^{-1}(t_k)\cdot\boldsymbol{S}(t_k)\right)\right]+\log(n)\times\hat{df}(\lambda_1,\lambda_2), \tag{3.20}$$

where $\hat{\boldsymbol{\Sigma}}^{-1}$ is the estimated precision matrix based on the estimated partial correlations. For the degree of freedom defined $\hat{df}(\lambda_1, \lambda_2)$ in (3.20), we use existing results in the literature to derive specific formulae for both (3.7) and (3.8). We also consider the extended BIC (EBIC) as an alternative criterion for tuning parameter selection, since it is generally believed to work better for models with many parameters (Foygel and Drton, 2010), but in our simulation studies (Section 3.4) both criteria give very similar results.

Zou et al. (2007) derived an explicit degree-of-freedom formula for the LASSO. Their approach can be easily adapted to derive the degree of freedom for (3.7)—by combing the l_2 penalty $\|\boldsymbol{D\theta}\|^2$ with the least-squares objective $\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}\theta}\|^2$; see A.2 in the supplementary materials. We obtain

$$\hat{df}_{GEN} = \mathrm{tr}\left\{ \left(\boldsymbol{\mathcal{X}}_{\mathcal{A}}^\top \boldsymbol{\mathcal{X}}_{\mathcal{A}} + n\lambda_2 \boldsymbol{D}_{\mathcal{A}}^\top \boldsymbol{D}_{\mathcal{A}}\right)^{-1} \boldsymbol{\mathcal{X}}_{\mathcal{A}}^\top \boldsymbol{\mathcal{X}}_{\mathcal{A}} \right\} \tag{3.21}$$

$$\approx \mathrm{tr}\left\{ \left(I + n\lambda_2 \left(\boldsymbol{\mathcal{X}}_{\mathcal{A}}^\top \boldsymbol{\mathcal{X}}_{\mathcal{A}} + \eta I\right)^{-1} \boldsymbol{D}_{\mathcal{A}}^\top \boldsymbol{D}_{\mathcal{A}} \right)^{-1} \right\}, \tag{3.22}$$

where $\mathcal{A} = \{i : \hat{\boldsymbol{\theta}}_i \neq 0\}$ denotes the active set, and $\boldsymbol{\mathcal{X}}_{\mathcal{A}}$ (or $\boldsymbol{D}_{\mathcal{A}}$) denotes the corresponding submatrix containing only the columns indexed by $\mathcal{A}$. Note that the matrix $\boldsymbol{\mathcal{D}}$, having fewer rows than columns, is not full-rank, and that, for a large p, the matrix $\boldsymbol{\mathcal{X}}$ is often not full-rank, either. Therefore, to compute (3.21) for any $\mathcal{A}$, it is necessary to first add a small perturbation matrix $\eta \boldsymbol{I}$ to $\boldsymbol{\mathcal{X}}_{\mathcal{A}}^\top \boldsymbol{\mathcal{X}}_{\mathcal{A}}$—we set $\eta = 10^{-5}$. The relation (3.22) holds because of the identity $(\boldsymbol{A} + \boldsymbol{B})^{-1} = (I + \boldsymbol{A}^{-1}\boldsymbol{B})^{-1}\boldsymbol{A}^{-1}$; it has an additional advantage over (3.21) that the trace of such a matrix inverse can be approximated by Chebyshev interpolation (Han et al., 2017).

Tibshirani and Taylor (2012) worked out how to compute the degree of freedom for a generalized LASSO problem, into which (3.8) can be transformed—see A.3 in the supplementary materials. Applying their result, we conclude that the degree of freedom for the GFL problem (3.8) is equal to dimension of the null space of $[\boldsymbol{D}^\top, \boldsymbol{I}]_{-\mathcal{B}}^\top$, where $\mathcal{B} = \{i : [\boldsymbol{D}^\top, \boldsymbol{I}]^\top \hat{\boldsymbol{\theta}}_i \neq 0\}$. It turns out this somewhat abstract conclusion can be further characterized (again, see A.3) by something more interpretable—namely,

$$\hat{df}_{GFL} = \sum_{1 \le i < j \le p} \left[\mathbb{1}\{\hat{\rho}_{ij}(1) \neq 0\} + \sum_{k=2}^{T} \mathbb{1}\{\hat{\rho}_{ij}(k) \neq \hat{\rho}_{ij}(k-1),\ \hat{\rho}_{ij}(k) \neq 0\} \right], \tag{3.23}$$

where $\mathbb{1}(\cdot)$ denotes a binary indicator function. That is, the degree of freedom here is simply the total number of nonzero fused groups over all $\hat{\rho}_{ij}$.

Remark Strictly speaking, the degree-of-freedom formulae derived by Zou et al. (2007) and Tibshirani and Taylor (2012) both require the regression of $\boldsymbol{\mathcal{Y}}$ onto $\boldsymbol{\mathcal{X}}$ to be homoscedastic, which is not the case for us, but we apply their results nonetheless because deriving similar results without the homescedastic assumption is currently an unsolved problem on its own. Hence, our degree-of-freedom formulae (3.22) and (3.23) are necessarily ad-hoc approximations, but they are still useful in facilitating the choice of tuning parameters through the BIC, as our empirical results below will demonstrate.

3.4 Simulation

In this section, we perform simulation studies to assess the performance of the two proposed approaches, GEN and GFL (with tuning parameters selected by both the BIC and the EBIC), and compare them with the a number of other methods. First, as basic benchmarks we compare with direct sample estimates and the naïve LASSO. Next, we apply the LASSO by allowing (3.3) to use a different tuning parameter λ_{1k} at each time point t_k—we refer to this as TS-LASSO for "time-specific LASSO". Finally, we compare with two methods extending the GLASSO by Danaher et al. (2014), to be abbreviated respectively as GLASSO-GGL and GLASSO-FGL. Among the methods that extend the GLASSO, these two leverage independent copies of the same multivariate Gaussian distribution at each time point; while most other GLASSO extensions are based on only one observation at each node per time point and calculate the sample covariance at each time point by down-weighting observations from other time points using a kernel function. In fact, GLASSO-GGL and GLASSO-FGL are originally proposed for K multivariate network datasets that are assumed to follow K individual GGMs with similar structures. Therefore, in addition to formulating the GGM likelihood differently, GLASSO-GGL and GLASSO-FGL apply the group LASSO and the generalized fused LASSO respectively to penalize the difference between any two precision matrices, while our methods are more specific to the temporal context and only penalize the differences in partial correlations at neighboring time points. We will see below in Section 3.4.2 that their GGL and FGL penalties, which operate on the differences between *any* two precision matrices, will usually result in over penalization in the temporal context.

The first objective of the simulation studies is to compare the performance of these methods in uncovering the underlying networks of the synthetic data, and the second objective is to investigate the accuracy of the estimated partial correlations.

3.4.1 Data Generation

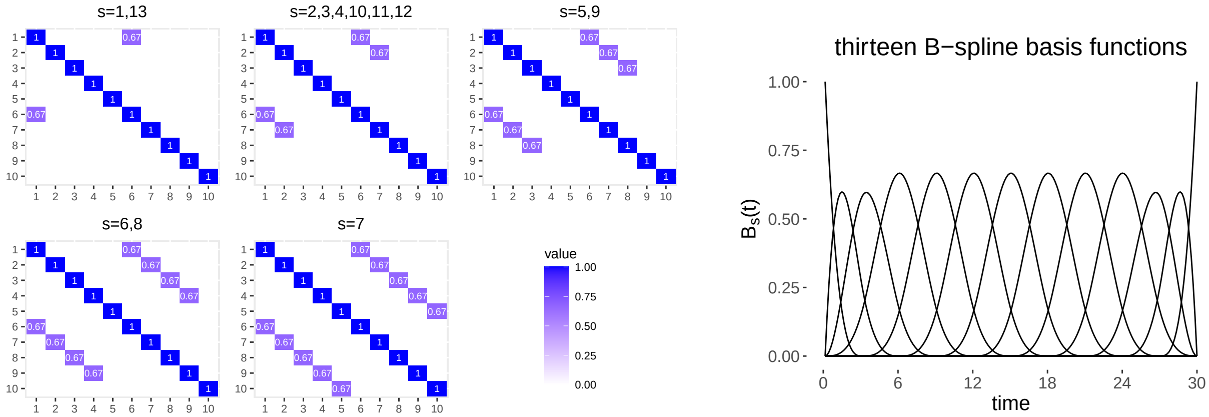
To generate synthetic time-varying network data, most existing work uses covariance matrices $\boldsymbol{\Sigma}(t_1), \boldsymbol{\Sigma}(t_2), \ldots$ that change smoothly over time, and then generates $\boldsymbol{X}(t_k)$ from $\boldsymbol{\Sigma}(t_k)$ independently at each t_k (e.g., Danaher et al., 2014; Gibberd and Nelson, 2017; Yang and Peng, 2020; Xue et al., 2020). There are two problems with this approach. First, since they are generated independently at each t_k, the actual data $\boldsymbol{X}(t_1), \boldsymbol{X}(t_2), \ldots$ do not usually change smoothly over time even though the covariance matrices $\boldsymbol{\Sigma}(t_1), \boldsymbol{\Sigma}(t_2), \ldots$ do. Second, for any $k \neq k'$, we will always have $\operatorname{cov}(\boldsymbol{X}(t_k), \boldsymbol{X}(t_{k'})) = 0$ under such a generating mechanism, which does not mimic reality very well.

We make another contribution in this work by proposing a different generating mechanism. Specifically, we generate each $x_i(t)$ as $x_i(t) = \mu(t) + e_i(t)$, where $\mu(t) = t + \sin(t)$, and the random vector $(e_1(t), \cdots, e_p(t))^\top$ as a linear combination from a set of S independent p-dimensional Gaussian random vectors whose coefficients are smooth functions, i.e., $(e_1(t), \cdots, e_p(t))^\top = \sum_{s=1}^{S} B_s(t)\,(\xi_{1,s}, \cdots, \xi_{p,s})^\top$, where $\{B_s(t), 1 \leq s \leq S\}$ denote S cubic B-spline basis functions defined on $[0, 1]$. For each s, $(\xi_{1,s}, \cdots, \xi_{p,s})^\top$ follows a centered multivariate Gaussian distribution with covariance matrix Σ_s. Thus, both the random components $e_i(t) = \sum_{s=1}^{S} B_s(t)\xi_{i,s}$ and the true precision matrix of $(x_1(t), \cdots, x_p(t))^\top$, given by $\{\sum_{s=1}^{S} B_s^2(t)\Sigma_s\}^{-1}$, are smooth over time.

The compact local support of B-splines (see Figure 3.1b) then makes it easy for us to generate sparse graphs over time. Since the graphic structure at one time point only involves a small number of B-spline basis functions and their Gaussian coefficients, we can easily achieve a sparse graphical structure at each time point by careful choices of each Σ_s. In particular, we assume that p is an even number, and rewrite the covariance matrix as a 2×2 block matrix, $\Sigma_s = (\Sigma_s^{11}, \Sigma_s^{12}; \Sigma_s^{21}, \Sigma_s^{22})$, where the four block submatrices are all diagonal. Under this design, it is easy to show (see Supplementary Section S.4) that the non-zero entries in the true precision matrix at time t are at the same positions as in $\sum_{s=1}^{S} B_s^2(t)\Sigma_s$. Therefore, as the non-zero entries in $\sum_{s=1}^{S} B_s^2(t)\Sigma_s$ change over time, the resulting precision matrix is still guaranteed to be sparse.

We take $S = 13$. Figure 3.1a shows what each Σ_s looks like, for all $s = 1, ..., 13$. Under this setting, there are only five true connections: 1–6, 2–7, 3–8, 4–9, and 5–10, and the profiles of the corresponding partial correlations are depicted as the red lines in Figure 3.2. Altogether, $p = 10$-dimensional normal random variables are simulated at each of 30 equally spaced time points on $[0, 1]$. We generate $x_i(t)$, $i = 1, \cdots, p$ on $[0, 1]$ independently for n subjects. To investigate the effects of sample size on the performance, we consider $n = 50$ and $n = 200$, while repeating each simulation for 100 times.

We think the aforementioned generating mechanism is interesting—not only does it ensure that both $x_i(t)$ and $\Sigma(t)$ are changing smoothly over t, it also guarantees $\Sigma^{-1}(t)$ is still sparse. But there is no "free lunch". One drawback of this approach is that the resulting network will necessarily remain relatively simple, due to the special block structure of each Σ_s needed to ensure the sparsity of the final precision matrix $\Sigma^{-1}(t)$. That's why additional simulation studies using the more "conventional" data generation mechanism (described at the beginning of this section) are still included in Supplementary Section A.6, with more “interesting” structures in the network—such as clusters, and differential connection tendencies within and between different clusters.



(a) (b)

Figure 3.1: (a) Heat maps of the pre-specified covariance matrices $\{\mathbf{\Sigma}_s\}_{s=1}^{13}$ corresponding to 13 cubic basis functions.(b) Thirteen B-spline basis functions $\{B_s(t)\}_{s=1}^{13}$ defined on $[0, 1]$, which is divided into 30 subintervals of equal length.

3.4.2 Results

We evaluate different methods with two metrics: (i) the estimation error $\sum_{t=1}^{T}[\sum_{1\leq i,j\leq p}\{\hat{\rho}_{ij}(t) - \rho_{ij}(t)\}^2]^{1/2}$, and (ii) the area under the ROC curve (AUC) which, in our context, is equal to the frequency that $|\hat{\rho}_{ij}(t)| > |\hat{\rho}_{i'j'}(t')|$ over all (i, j, t)-(i', j', t') pairings such that $\rho_{ij}(t) \neq 0$ and $\rho_{i'j'}(t') = 0$. While the first metric measures estimation quality, the second is simply an empirical estimate of the conditional probability that, given a truly-existing edge and a non-existing one, the estimated parameters would rank the true edge ahead of the non-existing one. Hence it measures the ability of different methods to detect the underlying network structure.

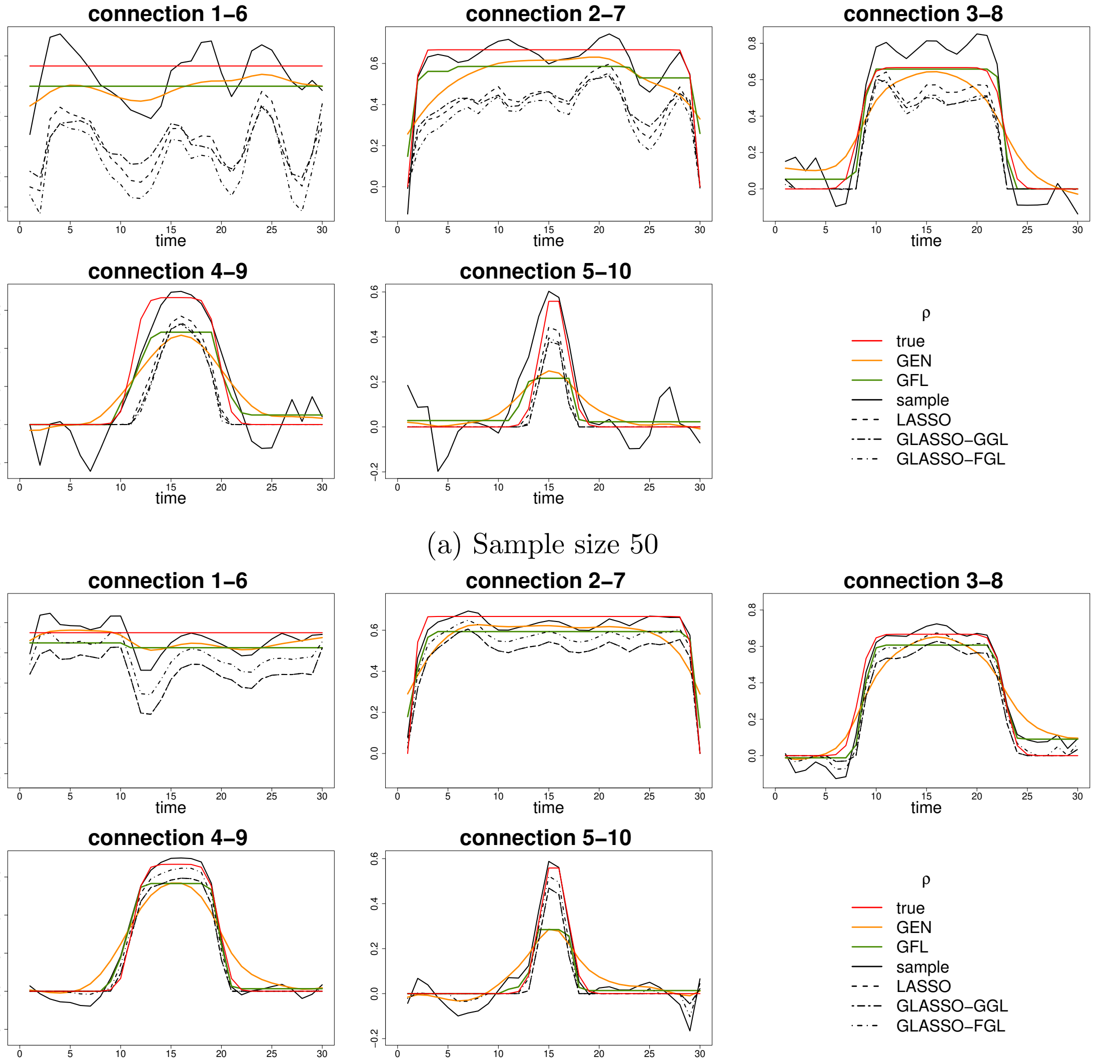
Tables 3.1 summarizes the estimation errors and the estimated AUCs of different methods over 100 simulation replicates. We can see that the performances of all methods improve when sample size is increased from 50 to 200, as expected, and that our proposed methods (GEN and GFL) offer noticeable improvements over the other methods considered. The results also indicate that, for both GEN and GFL, selecting tuning parameters by either the BIC or the EBIC does not make a substantial difference.

Method	Estimation error		AUC	
	$n = 50$	$n = 200$	$n = 50$	$n = 200$
Sample	42.77 (0.17)	20.01 (0.08)	0.882 (0.002)	0.895 (0.001)
LASSO	21.73 (0.17)	9.22 (0.06)	0.876 (0.001)	0.904 (0.001)
TS-LASSO	22.00 (0.17)	9.56 (0.06)	0.876 (0.001)	0.905 (0.002)
GLASSO-GGL	23.23 (0.17)	9.32 (0.07)	0.889 (0.002)	0.895 (0.001)
GLASSO-FGL	18.93 (0.16)	9.32 (0.07)	0.881 (0.001)	0.895 (0.001)
GEN	18.41 (0.09)	12.87 (0.06)	0.961 (0.002)	0.974 (0.001)
GEN-EBIC	18.30 (0.08)	13.31 (0.05)	0.973 (0.001)	0.990 (0.001)
GFL	13.54 (0.17)	6.14 (0.07)	0.943 (0.003)	0.949 (0.002)
GFL-EBIC	11.93 (0.11)	6.58 (0.07)	0.942 (0.003)	0.934 (0.002)

Table 3.1: Mean estimation error and AUCs across 100 replicates with standard error in parentheses.

To gain more insights, we also selectively showcase some specific results, all of which are based on one simulation rather than over 100 repetitions. First, for the set $\{(i, j) :$ there exists t such that $\rho_{ij}(t) \neq 0\}$, Figure 3.2 shows the estimated profiles $\hat{\rho}_{ij}(t)$ as a function of time from one simulation instance. We observe more clearly the improvement from $n = 50$ to $n = 200$. Not surprisingly, we find that GEN produces smooth functions while GFL produces staircase-shaped functions. Next, Figures A.1 and A.2 in the supplementary materials display the GFL- and GEN-estimated networks (with $n = 200$) over every time point, with false-positive and false-negative edges clearly indicated at these time points as well. We can see that our estimated network structure does not show rapid bursts of changes from time to time.

In Figure 3.3 and similar ones included in the supplementary materials (Figures A.3, A.4 and A.5), we compare the effectiveness of our tuning parameter selection strategy as well as that of Danaher et al. (2014). The top panels of Figure 3.3 show the contours of the BIC, of the estimation error, and of the AUC over a grid of (λ_1, λ_2) for our GEN penalty. They demonstrate that the tuning parameters selected by our BIC indeed lead to good solutions in terms of both performance metrics. The naïve LASSO solution, with its tuning parameter also selected by the BIC, is indicated as well, whereas the sample estimate is, of course, at the origin $(0, 0)$. We can see that, while the LASSO solution is clearly better than the sample estimate, incorporating the additional GEN penalty provides substantial further improvements. The bottom panels of Figure 3.3, on the other hand, reveal that the tuning parameters selected by Danaher et al. (2014) using their AIC criterion do not really lead to the best possible GLASSO-GGL estimate. This could be attributed to the



(a) Sample size 50

(b) Sample size 200

Figure 3.2: Estimated partial correlations for true non-vanishing edges.

oversimplified degrees of freedom used by Danaher et al. (2014) in their AIC formula—rather than counting the degrees of freedom carefully for each different penalty function, as we do in Section 3.3.2, they simply count the number of non-zero elements in the estimated precision matrices. Similar contour plots for our GFL penalty and for their GLASSO-FGL are shown in Section A.5 of the supplementary materials.

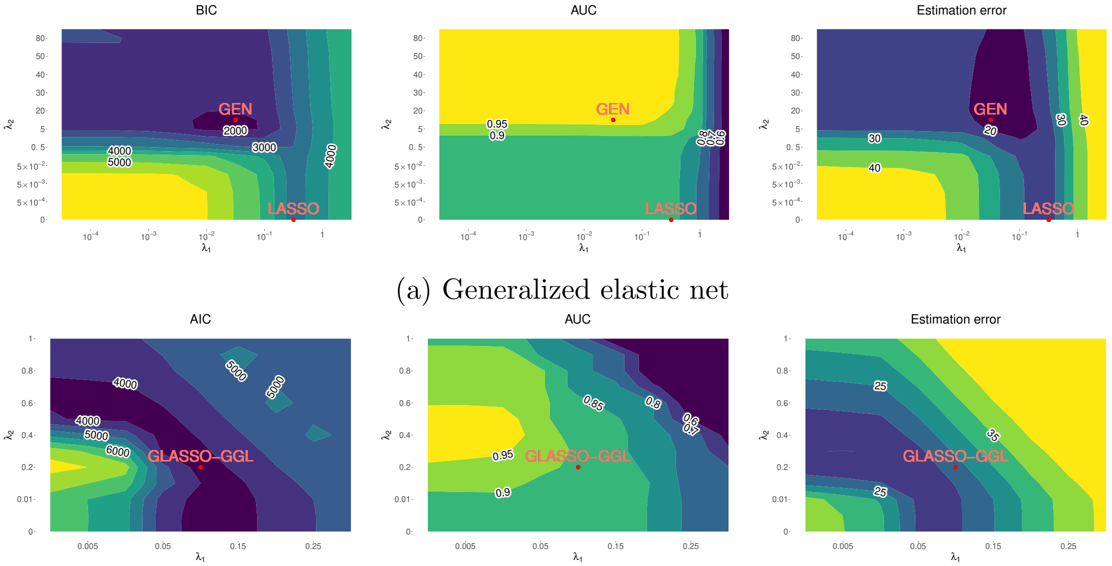
We also note that, when the sample size is 200, the optimal λ_2's selected by the two GLASSO methods are both zero, due to the over-penalization we mentioned earlier at the beginning of Section 3.4. To a certain extent, this confirms that, when network structures are smoothly changing over time, it is more appropriate to penalize *only* the difference of two adjacent networks than to penalize the difference of *any* two networks. With $\lambda_2 = 0$, both GLASSO-GGL and GLASSO-FGL are simply solving a vanilla LASSO problem—except here they are using the GLASSO algorithm, which gives slightly different numeric solutions from the SPACE algorithm; that's why the performance of these two GLASSO extensions are not identical to vanilla LASSO in Table 3.1, but one should not read too much into these small numeric differences.

3.5 Applications

3.5.1 Application: Analysis of Canadian Temperature Data

In this section, we analyze a real data set of hourly temperature measurements in July for thirteen Canadian cities, extracted from a government website (Environment and Climate Change Canada, 2022). The data set includes hourly air temperature measurements in degrees Celsius (°C) at thirteen stations across Canada, taken repeatedly during the month of July between 2019 and 2022, i.e., $X_{ij}(t)$, for $j = 1, 2, ..., 13$ stations, $t = 1, 2, ..., 24$ time points, and $i = 1, 2, ..., 31 \times 4 = 124$ measurements. There were 36 missing values at 11 different stations, a relatively small proportion, so we simply imputed them using $\bar{X}_{\cdot j}(t)$ for the corresponding station j and time point t. For maximal compatibility, the 13 stations we selected are mostly near an airport, so they have at least similar terrain conditions (e.g., not in the middle of a forest or on a lake, etc). We also centered the data so that $\bar{X}_{\cdot j}(t) = 0$ at every j and t.

Strictly speaking, for each station j and time point t, the repeated measurements $X_{ij}(t)$ here are not independent across $i = 1, 2, ..., 124$, since they are mostly observed on consecutive days, but most of the serial correlations are fairly weak—see residual plots of $X_{ij}(t) - \hat{X}_{ij}(t)$ against i together with Durbin-Watson statistics, as well as autocorrelation



(a) Generalized elastic net

(b) GLASSO-GGL

Figure 3.3: Simulation performance measurements from generalized elastic net and GLASSO-GGL both with sample size 50. Darker areas represent lower values. For reference, in the simulation from generalized elastic net, the sample AUC is 0.8757, comparing to the GEN AUC 0.9690 in the figure; the sample estimation error is 41.81, comparing to the estimation error of GEN as 18.20 in the figure. In the simulation from GLASSO-GGL, the GLASSO-GGL AUC is 0.8907 and the GLASSO-GGL estimation error is 22.92 in the figure.

function (ACF) plots for these residuals, in Supplementary Section A.8.3. Therefore, we feel it is not unreasonable to still regard them as “almost independent” replications.

To characterize the time-varying associations of hourly temperatures in July across these thirteen cities, we apply both GEN and GFL to fit time-varying networks separately. We estimate partial correlation networks using a series of λ_1 and λ_2 values. The BIC surfaces—i.e., (3.20) for both methods—turn out to be low and quite flat over moderate λ_1 values, suggesting that there are no substantial differences among different solutions in this overall valley region. For fixed λ_1 values in this region, the BIC values generally decrease in the λ_2 direction, quickly at first but slowing down afterwards. For each method, we therefore examine two specific solutions: one, which we refer to as “Result 1”, is given by the largest λ_1 that still puts us in this valley region and the “first” strictly positive λ_2; and another, which we refer to as “Result 2”, is given by a pair of (λ_1, λ_2) also in this valley region, with a slightly smaller λ_1 but larger λ_2 so that the resulting solution still has a degree of freedom not too different from that of "Result 1". Table A.4 in the supplementary materials summarizes some key features of the four solutions, including the degree of freedom, and the total number of edges/connections over all time points.

The main conclusions we can draw from all four sets of results—i.e., (Result 1, Result 2)×(GEN, GFL)—turn out to be identical, which give us confidence in their scientific validity, to the extent justified by the quality of the data set itself. To reduce redundancy, therefore, we present only Result 1 of both GEN and GFL in the main text; the other set of results are provided in A.8 in the supplementary materials.

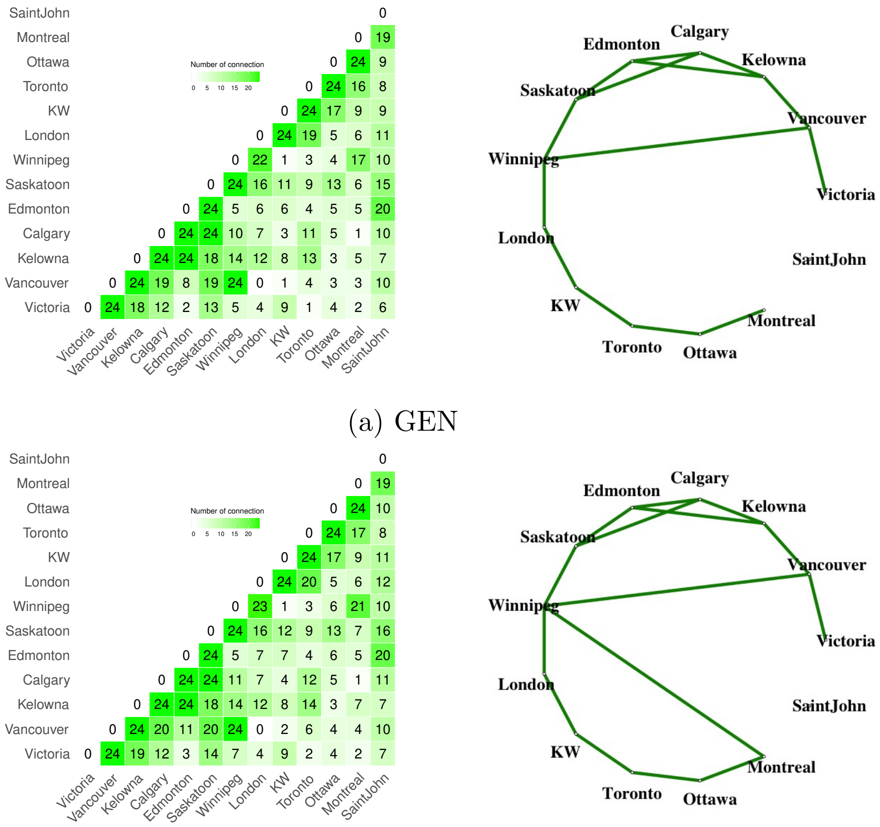
First, Figure 3.4—and similarly Figures A.8 in Section A.8—show the frequency of connections between any two cities over all 24 hours. The left panel contains matrices where each entry represents the total number of occurrences for the corresponding connection, and, after removing the connections that occur ≤ 20 times, these matrices are visualized as sparse networks of 13 cities in the right panel.

Next, graphically displayed in Figure 3.5 here—and likewise in Figures A.9 of Section A.8—are the estimated partial correlations between any two cities at each time point. They provide further information as to when different interactions occur and how they change over time. The most prominent observations here—from all sets of results—are: (i) there are more connections over the 24-hour period between cities that are geographically close to each other (bottom rows) and fewer between those that are far apart (top rows); (ii) estimated partial correlations between cities that are geographically close to each other (again, bottom rows) tend to be positive; (iii) connections, either positive or negative, occur more frequently in the afternoon and less frequently in the morning.

The first two observations are quite intuitive; temperatures at locations that are geographically close tend to move together more often. The last observation reveals an interesting phenomenon akin to the heat island effect of individual cities, whereby the temperature differences between a metropolitan area and its surrounding rural areas are usually larger at night than they are during the day due to human activities. Here, we can detect a similar pattern for the number of connections in a temperature network between different cities that spread over a very wide area: fewer connections early in the morning, and more connections in the afternoon.

Another noticeable observation concerns the city of Saint John. Many of its estimated partial correlations with cities very far away (such as Edmonton and Saskatoon) turn out to be positive, which is at first counter-intuitive. We think such a special temperature pattern is probably due to the special geographic location of Saint John in a sheltered harbor.

Finally, as a baseline benchmark to compare with, results from naïve LASSO and TS-LASSO are provided in Supplementary Section A.8.



	Victoria	Vancouver	Kelowna	Calgary	Edmonton	Saskatoon	Winnipeg	London	KW	Toronto	Ottawa	Montreal	SaintJohn
SaintJohn													0
Montreal												0	19
Ottawa											0	24	9
Toronto										0	24	16	8
KW									0	24	17	9	9
London								0	24	19	5	6	11
Winnipeg							0	22	1	3	4	17	10
Saskatoon						0	24	16	11	9	13	6	15
Edmonton					0	24	5	6	6	4	5	5	20
Calgary				0	24	24	10	7	3	11	5	1	10
Kelowna			0	24	24	18	14	12	8	13	3	5	7
Vancouver		0	24	19	8	19	24	0	1	4	3	3	10
Victoria	0	24	18	12	2	13	5	4	9	1	4	2	6

(a) GEN

	Victoria	Vancouver	Kelowna	Calgary	Edmonton	Saskatoon	Winnipeg	London	KW	Toronto	Ottawa	Montreal	SaintJohn
SaintJohn													0
Montreal												0	19
Ottawa											0	24	10
Toronto										0	24	17	8
KW									0	24	17	9	11
London								0	24	20	5	6	12
Winnipeg							0	23	1	3	6	21	10
Saskatoon						0	24	16	12	9	13	7	16
Edmonton					0	24	5	7	7	4	6	5	20
Calgary				0	24	24	11	7	4	12	5	1	11
Kelowna			0	24	24	18	14	12	8	14	3	7	7
Vancouver		0	24	20	11	20	24	0	2	6	4	4	10
Victoria	0	24	19	12	3	14	7	4	9	2	4	2	7

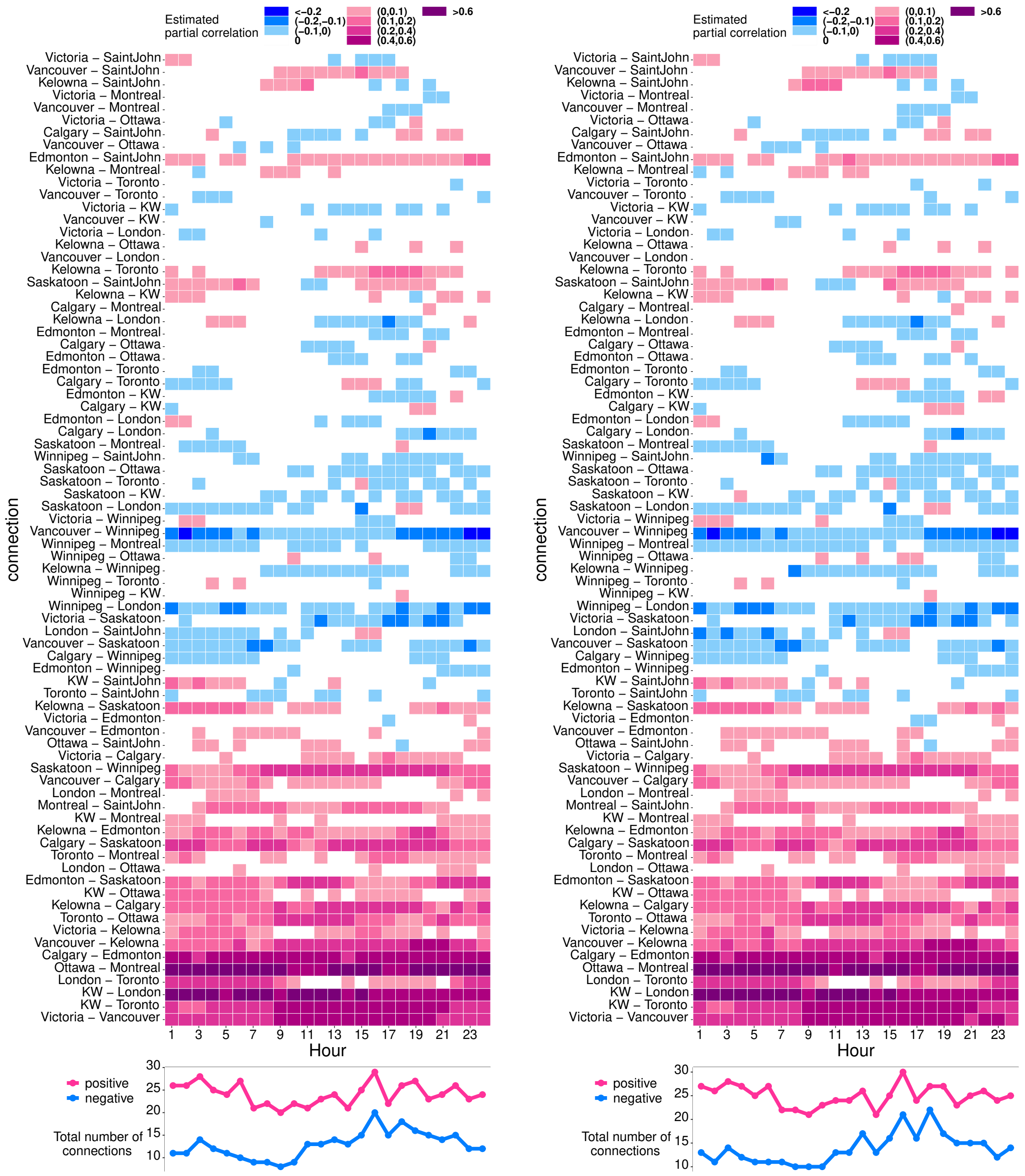
(b) GFL

Figure 3.4: [Result 1] Aggregated connections between different cities over 24 time points.

3.5.2 Application: Analysis of ADHD Data

In this section, we apply the proposed methods (GEN and GFL) to estimate time-varying brain connectivity with a real data set about attention deficit hyperactive disorder (ADHD).

ADHD is a mental health disorder characterized by impulsivity, motoric hyperactivity, and especially, attention deficits. With a global community prevalence of 5% - 10%, however, its causes are unknown, although gene damage may be a contributing factor. Diagnosis is mainly supported by clinical assessment based on long-term observations and on



(a) GEN

(b) GFL

Figure 3.5: [Result 1] Estimated partial correlations between different cities over 24 hours. Connections are listed in descending order of the distances between cities from the top to the bottom. Each cell corresponds to a connection at a given time point, and the color characterizes the value of the estimated partial correlation. The lower line chart marginalizes the total connections at each hour.

identifying a range of symptoms. Recently, the cerebellum that contains more than 50% of the neurons in the brain has been thought of as having vital involvement in ADHD. MRI studies (Berquin et al., 1998) suggest that the cerebellar hemispheric volumes in ADHD sufferers are up to 6% smaller than healthy subjects, and ADHD children have less vermal volume than healthy ones. Subsequent researches consistently find significant differences between ADHD brains and healthy ones in posterior inferior lobe of the cerebellum (lobules VIII–X) and the posterior–inferior cerebellar vermis (Mostofsky et al., 1998). Therefore, exploring the topological structures in the cerebellum region and their dynamic changes under ADHD is critical to a better understanding of the mechanisms underlying the disorder. Moreover, knowing the differences in brain connectivity between healthy and ADHD groups can contribute to the development of diagnosis methods.

We use the resting-state fMRI data set collected at New York University Medical Center (NYU), one of the eight imaging sites contributing to the ADHD-200 Global Competition that was held to gather neuroimaging data for the classification of ADHD subjects. The data set consists of filtered and preprocessed resting-state data for 116 brain regions of interest (ROIs) segmented by the Automated Anatomical Labeling (AAL) atlas. We extracted data only from the 91st to the 108th ROIs, corresponding to the cerebellum region. For each particular region, the mean blood-oxygen-level dependent (BOLD) signal was recorded at 172 equally spaced time points. There are 98 healthy subjects and 118 ADHD patients.

To characterize how time-varying associations in cerebellum regions differ for the ADHD and healthy groups, we apply both GEN and GFL to fit time-varying networks for the two groups separately. First, we center the mean BOLD signal at each ROI to zero at each time point. Then we estimate partial correlation networks using a series of λ_1 and λ_2 values. The BIC surfaces—i.e., (3.20) for both methods—turn out to be quite flat, suggesting that there are no substantial differences among different solutions. For each method, we therefore examine two specific solutions: one, which we refer to as “Result 1”, is given by the “first” strictly positive $(\lambda_1, \lambda_2) > (0, 0)$ in our grid; and another, which we refer to as “Result 2”, is given by a pair of (λ_1, λ_2) in our grid such that the corresponding degree of freedom $\hat{df}(\lambda_1, \lambda_2)$ is closest to half of that from “Result 1”.

Table 3.2 summarizes some key features of the four solutions, including the degree of freedom, the total number of edges/connections over all time points, and the respective number of edges/connections during the first $(1 \leq t \leq 86)$ and second $(87 \leq t \leq 172)$ halves of the scanning period. Note that, for each set of solutions, the degrees of freedom and total number of edges/connections are similar between the healthy and ADHD groups, so it is reasonable and meaningful to compare them.

Generalized elastic net	Result 1		Result 2	
	Healthy	ADHD	Healthy	ADHD
degrees of freedom	685.59	748.21	292.53	312.04
# of connections	990	1073	1098	1171
# of connections during $(1^{st}, 2^{nd})$ half period	(490,500)	(269,804)	(541,557)	(292,879)

Generalized fused LASSO	Result 1		Result 2	
	Healthy	ADHD	Healthy	ADHD
degrees of freedom	937	853	563	504
# of connections	1606	1766	855	941
# of connections during $(1^{st}, 2^{nd})$ half period	(780,826)	(509,1257)	(408,447)	(172,769)

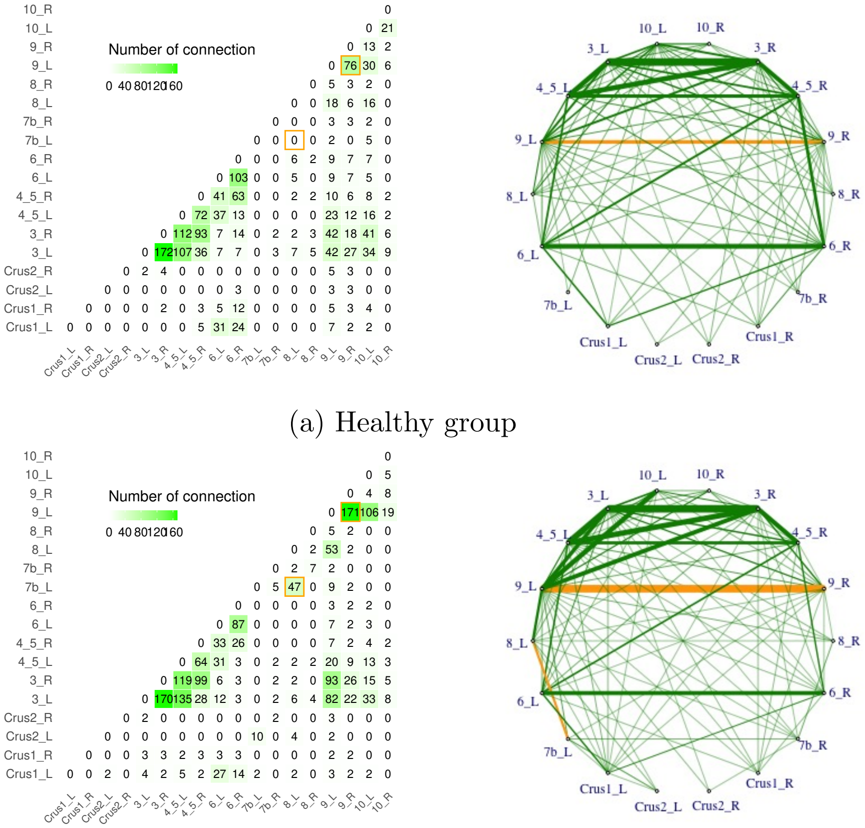
Table 3.2: Summary of degrees of freedom and number of detected edges based on the GEN and the GFL with two pairs of selected tuning parameters.

The main conclusions we can draw from all four sets of results—i.e., (Result 1, Result 2)×(GEN, GFL)—turn out to be identical, which give us confidence in their scientific validity, to the extent justified by the quality of the data set itself. To reduce redundancy, therefore, we present only Result 1 from GFL in the main text; the other three sets of results are provided in A.9 in the supplementary materials.

First, Figure 3.6—and similarly Figures A.16, A.18, A.20 in Section A.9—show the frequency of connections between any two regions over all 172 time points of the entire scanning period. The left panel contains matrices where each entry represents the total number of occurrences for the corresponding connection, and these matrices are visualized as networks of eighteen cerebellar ROIs in the right panel, where the thickness of each edge is proportional to the number of occurrences for that connection. The most prominent observations here—from all four sets of results—are that (i) the connection between 7b_L and 8_L occurred only in the ADHD group but never appeared in the healthy group, and that (ii) the connection between 9_L and 9_R occurred a lot more often in the ADHD group than it did in the healthy group.

Next, graphically displayed in Figure 3.7 here—and likewise in Figures A.17, A.19, A.21 of Section A.8—are the estimated partial correlations at each time point. They provide further information as to when different interactions occur and how they change over time. Some connections are never observed; some are only active at certain time points; and others persistently appear throughout the entire scanning period—e.g., the top five most frequently detected connections are listed in table 3.3. A consistent observation here—

again, from all four sets of results (also see Table A.4)—is that, other than connections that persistently show up during the entire scanning period, the time points at which specific connections occur between ROIs are markedly different for the two groups. In particular, for the healthy group, there were more or less equal number of connections during the first and second halves of the scanning period; whereas, for the ADHD group, many more connections occurred during the second half than the first.



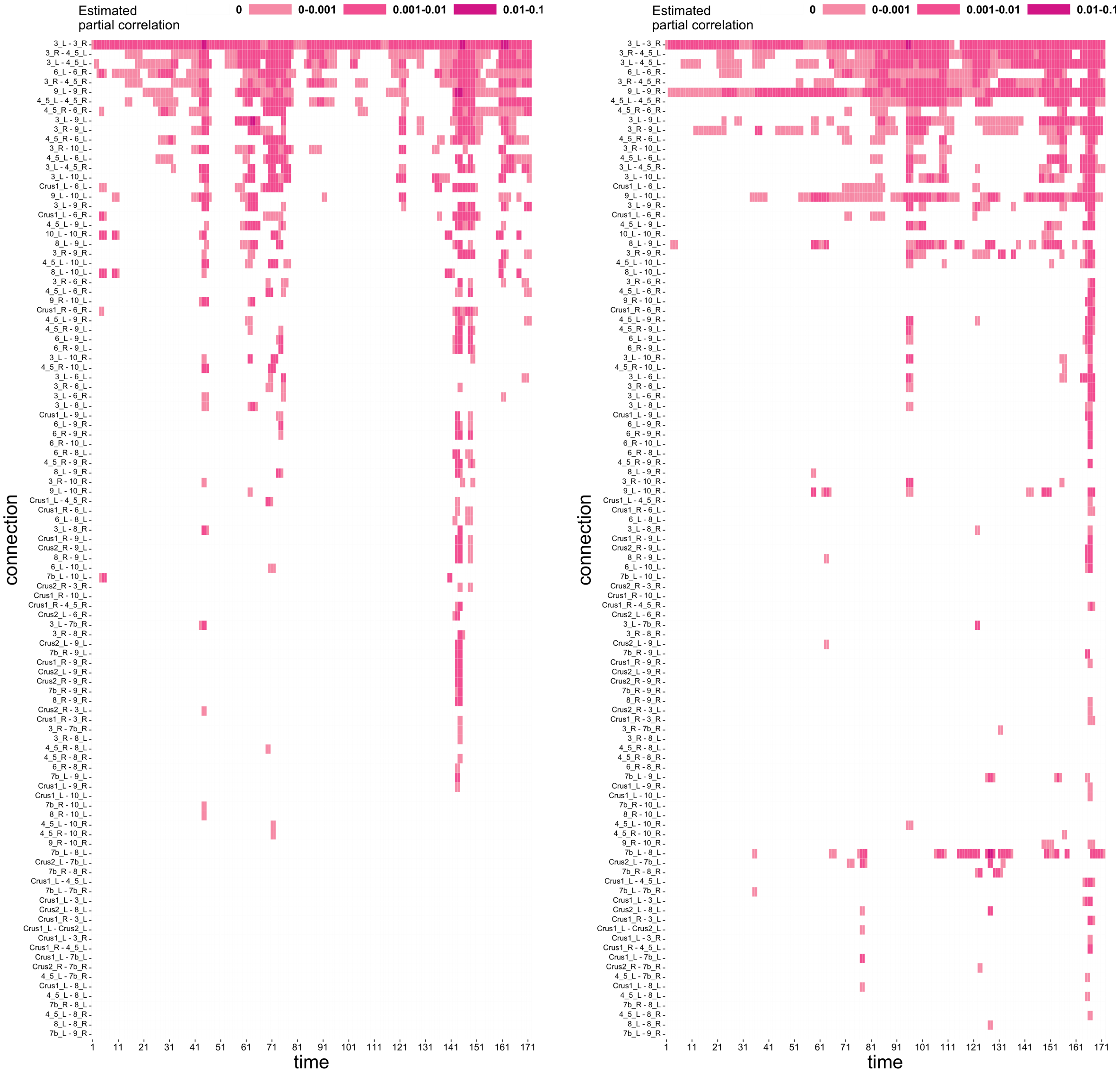
	Crus1_L	Crus1_R	Crus2_L	Crus2_R	3_L	3_R	4_5_L	4_5_R	6_L	6_R	7b_L	7b_R	8_L	8_R	9_L	9_R	10_L	10_R
10_R																		0
10_L																	0	21
9_R																0	13	2
9_L															0	76	30	6
8_R														0	5	3	2	0
8_L													0	0	18	6	16	0
7b_R												0	0	0	3	3	2	0
7b_L											0	0	0	0	2	0	5	0
6_R										0	0	0	6	2	9	7	7	0
6_L									0	103	0	0	5	0	9	7	5	0
4_5_R								0	41	63	0	0	2	2	10	6	8	2
4_5_L							0	72	37	13	0	0	0	0	23	12	16	2
3_R						0	112	93	7	14	0	2	2	3	42	18	41	6
3_L					0	172	107	36	7	7	0	3	7	5	42	27	34	9
Crus2_R				0	2	4	0	0	0	0	0	0	0	0	5	3	0	0
Crus2_L			0	0	0	0	0	0	0	3	0	0	0	0	3	3	0	0
Crus1_R		0	0	0	0	2	0	3	5	12	0	0	0	0	5	3	4	0
Crus1_L	0	0	0	0	0	0	0	5	31	24	0	0	0	0	7	2	2	0

(a) Healthy group

	Crus1_L	Crus1_R	Crus2_L	Crus2_R	3_L	3_R	4_5_L	4_5_R	6_L	6_R	7b_L	7b_R	8_L	8_R	9_L	9_R	10_L	10_R
10_R																		0
10_L																	0	5
9_R																0	4	8
9_L															0	171	106	19
8_R														0	5	2	0	0
8_L													0	2	53	2	0	0
7b_R												0	2	7	2	0	0	0
7b_L											0	5	47	0	9	2	0	0
6_R										0	0	0	0	0	3	2	2	0
6_L									0	87	0	0	0	0	7	2	3	0
4_5_R								0	33	26	0	0	0	0	7	2	4	2
4_5_L							0	64	31	3	0	2	2	2	20	9	13	3
3_R						0	119	99	6	3	0	2	2	0	93	26	15	5
3_L					0	170	135	28	12	3	0	2	6	4	82	22	33	8
Crus2_R				0	2	0	0	0	0	0	0	2	0	0	3	0	0	0
Crus2_L			0	0	0	0	0	0	0	0	10	0	4	0	2	0	0	0
Crus1_R		0	0	0	3	3	2	3	3	3	0	0	0	0	2	2	0	0
Crus1_L	0	0	2	0	4	2	5	2	27	14	2	0	2	0	3	2	2	0

(b) ADHD group

Figure 3.6: [Result 1,GFL] Aggregated connections between different cerebellum regions over 172 time points based on the GFL. The yellow squares on the left highlight the number of the edge 7b_L - 8_L and the edge 9_L - 9_R.



(a) Healthy group

(b) ADHD group

Figure 3.7: [Result 1, GFL] Estimated partial correlations between different cerebellum regions over 172 time points based on the GFL. Each cell corresponds to a connection at a given time point, and the color represents the magnitude of the estimated partial correlation.

Generalized elastic net			
Result 1		Result 2	
Healthy	ADHD	Healthy	ADHD
3_L - 3_R (147)	3_L - 3_R (144)	3_L - 3_R (157)	3_L - 3_R (148)
3_L - 4_5_L (67)	9_L - 9_R (115)	3_L - 4_5_L (74)	9_L - 9_R (125)
3_R - 4_5_L (61)	3_L - 4_5_L (79)	3_R - 4_5_L (71)	3_L - 4_5_L (85)
6_L - 6_R (51)	3_R - 4_5_L (68)	6_L - 6_R (56)	3_R - 4_5_L (70)
3_R - 4_5_R (50)	9_L - 10_L (51)	3_R - 4_5_R (55)	3_R - 4_5_R (55)

Generalized fused lasso			
Result 1		Result 2	
Healthy	ADHD	Healthy	ADHD
3_L - 3_R (172)	9_L - 9_R (171)	3_L - 3_R (160)	3_L - 3_R (153)
3_R - 4_5_L (112)	3_L - 3_R (170)	3_L - 4_5_L (60)	9_L - 9_R (123)
3_L - 4_5_L (107)	3_L - 4_5_L (135)	3_R - 4_5_L (49)	3_R - 4_5_L (74)
6_L - 6_R (103)	3_R - 4_5_L (119)	4_5_L - 4_5_R (42)	3_L - 4_5_L (73)
3_R - 4_5_R (93)	9_L - 10_L (106)	3_R - 4_5_R (40)	3_R - 4_5_R (54)

Table 3.3: Top 5 connections and the number of their occurrences over 172 time points of each set of result.

3.6 Conclusion

Although generalizing the elastic net and the fused LASSO as we have done in this section are not the only ways to model time-varying network data, they are useful additions to the existing toolbox, especially since we extend the regression approach whereas most existing literature in this area has focused on extending the GLASSO approach. While the idea of imposing l_1 and l_2 penalties on the difference in partial correlations at adjacent time points may be straightforward, the resulting optimization problems are not exactly trivial to solve. Some valuable lessons from our work are: first, the ADMM provides a unifying framework for solving both the GEN and the GLF problems, but it is critical to parameterize the penalty functions differently for the two approaches in order to gain full computational advantage; second, tricks that exploit special structures in otherwise large matrices are always useful.

The approximate degree-of-freedom formulae we derived are useful—and perhaps even sufficient as we have demonstrated—for practical purposes, but the correct degrees of freedom remain elusive and an open problem.

We also proposed a new generating mechanism (Section 3.4.1) to simulate time-varying network data. In order to control exactly where nonzero entries can occur in the precision matrix, the "usual" simulation strategy has faced some long-standing difficulties—for example, only the precision matrix can be smooth functions of time but not the simulated data themselves; and the simulated data at different time points are always independent of each other. Our method overcomes both of these difficulties but, as a trade-off, it is currently incapable of simulating very complex network structures.

Finally, our analysis of the Canadian temperature data set also leads to some interesting discoveries. Other than common-sense observations such as daily temperatures for cities close to each other tend to move together, we are also able to discover that temperatures across different cities are less correlated early in the morning than they are in the afternoon, and that the city of Saint John is a very different node in this network of 13 Canadian cities. Moreover, our analysis of the fMRI data set also leads to some interesting speculations. On the one hand, that the BIC-surfaces are quite flat over a wide range of (λ_1, λ_2) values may be an indication that there is limited information in this data set. Indeed, scientists have questioned the usefulness of resting-state fMRI scans for studying ADHD (Lurie et al., 2020). On the other hand, that the same conclusions can be drawn from four different solutions, with remarkably different degrees of freedom, is in itself a strong testament that these conclusions are probably not false discoveries. This is in line with the basic philosophy behind stability selection (Meinshausen and Bühlmann, 2010). If no scientific explanation is immediately available, we think they are at least genuine artefacts of this particular data set.

Chapter 4

Restricted Tweedie Stochastic Block Models

4.1 Introduction

4.1.1 Background

A community can be conceptualized as a collection of nodes that exhibit similar connection patterns in a network. Community detection is a fundamental problem in network analysis, with wide applications in social network (Bedi and Sharma, 2016), marketing (Bakhthemmat and Izadi, 2021), recommendation systems (Gasparetti et al., 2021), and political polarization detection (Guerrero-Solé, 2017). Identifying communities in a network not only enables nodes to be clustered according to their connections with each other, but also reveals the hierarchical structure that many real-world networks exhibit. Furthermore, it can facilitate network data processing, analysis, and storage (Lu et al., 2018).

Among the various methods for detecting communities in a network, the Stochastic Block Model (SBM) stands out as a probabilistic graph model. It is founded based on the *stochastic equivalence* assumption, positing that the connecting probability between node i and node j depends solely on their community memberships (Holland et al., 1983). If we assume that given the community memberships of two nodes i and j, denoted by c_i and c_j, the edge weight between them is Bernoulli distributed. In particular, letting Y_{ij} denote this weight, the adjacency matrix $Y = (Y_{ij})$ is generated as

$$Y_{ij} \mid c_i = k, c_j = l \sim \text{Bernoulli}(B_{kl}), \tag{4.1}$$

where B_{kl} denotes the probability of connectivity between the nodes from the kth and lth communities.

As indicated in (4.1), an SBM provides an interpretable representation of the network's community structure. Moreover, an SBM can be efficiently fitted with various algorithms, such as maximum likelihood estimation and Bayesian inference (Lee and Wilkinson, 2019). In recent few years, there has been extensive research on theoretical properties of the estimators obtained from these algorithms (Lee and Wilkinson, 2019).

In this chapter, we are motivated to leverage the remarkable capability of the SBM in detecting latent community structures to tackle an interesting problem—clustering countries into different groups based on their international trading patterns. However, in this application, we encounter three fundamental challenges that can not be addressed by existing SBM models.

4.1.2 Three Main Challenges

Edge Weights

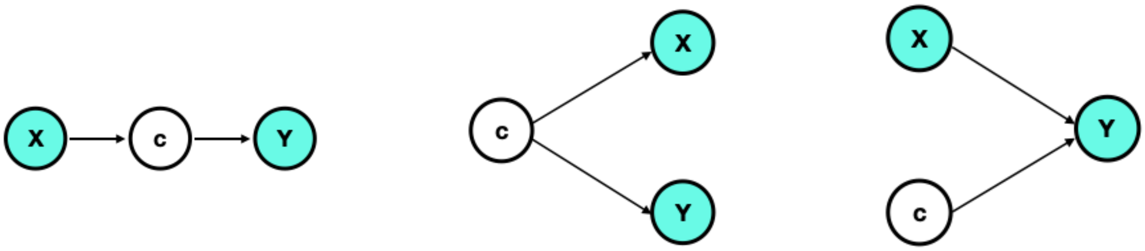
The classical SBM, as originally proposed by Holland et al. (1983), is primarily designed for binary networks, as indicated in (4.1). However, in the context of the international trading network, we are presented with richer data, encompassing not only the presence or absence of trading relations between countries but also the specific trading volumes in dollars. These trading volumes serve as the intensity and strength of the trading relationships between countries. In such cases, thresholding the data to form a binary network would inevitably result in a loss of valuable information.

In the literature, several methods have been developed to extend the modeling of edge weights beyond the binary range. Some methods leverages distributions capable of handling edge weights. For instance, Aicher et al. (2013, 2015) adopt a Bayesian approach to model edge weights using distributions from the exponential family. Ludkin (2020) allows for arbitrary distributions in modeling edge weights and sample the posterior distribution using a reversible jump MCMC method. Ng and Murphy (2021) and Motalebi et al. (2021) use a compound Bernoulli-Gamma distribution and a Hurdle model to represent edge weights respectively. Haj et al. (2022) apply the binomial distribution to networks with integer-valued edge weights that are bounded from above. In contrast, there is a growing interest in multilayer networks, where edge weights are aggregated across network layers. Notable examples of research in this area include the work by MacDonald et al. (2022) and Chen and Mo (2022). Notably, Zhang and Chen (2020) introduce a novel modularity-based community detection framework specifically tailored for heterogeneous networks comprising multiple types of nodes and diverse types of edges linking them.

However, the above approaches cannot properly deal with financial data that involve non-negative continuous random variables with a large number of zeros and a right-skewed distribution.

Incorporating Nodal Information

Many SBMs assume that nodes within the same community exhibit stochastic equivalence. However, this assumption can be restrictive and unrealistic, as real-world networks are influenced by environmental factors, individual node characteristics, and edge properties, leading to heterogeneity among community members that affects network formation. Depending on the relationship between communities and covariates, there are generally three classes of models, as shown in Figure 4.1. Models (b) and (c) have been previously discussed by Huang et al. (2023). We are also particularly interested in model (c), where latent community labels and covariates jointly shape the network structure. In our study on international trading networks, factors such as the geographical distance between countries, along with community labels, play critical roles in shaping trading relations. Neglecting these influential factors can significantly compromise the accuracy of SBM estimations.



(a) Covariates-driven (b) Covariates-confounding (c) Covariates-adjusted

Figure 4.1: Three network models with covariates. The symbols X, Y and c represent covariates, network connection and community memberships, respectively. A shaded/unshaded cell means the corresponding quantity is observable/latent.

Various works in the past have considered the incorporation of nodal information. For instance, Roy et al. (2019) and Choi et al. (2012) considered a pairwise covariate effect in the logistic link function when modeling the edge between two nodes. In contrast, Ma and

Ma (2017) and Hoff et al. (2002) incorporated the pairwise covariate effect but with a latent space model. Other research considering covariates in an SBM includes Tallberg (2004), Vu et al. (2013) and Peixoto (2018). Moreover, Mariadassou et al. (2010) and Huang et al. (2023) addressed the dual challenge of incorporating the covariates and modeling the edge weights by assuming that each integer-valued edge weight follows a Poisson distribution and accounting for the pairwise covariates into the mean.

While the aforementioned literature has made significant progress in incorporating covariate information into network modeling, the complexity escalates when we confront the third challenge — the observed network is changing over time. This challenge necessitates a deeper exploration of how covariates influence network formation dynamically — a facet that remains unaddressed in the existing literature.

Dynamic Network

Recent advances in capturing temporal network data demand the extension of classic SBMs to dynamic settings, as previous research predominantly focused on static networks.

Researchers have attempted to adapt SBMs to dynamic settings, employing various strategies such as state-space models, hidden Markov chains, and change point detection. Fu et al. (2009) and Xing et al. (2010) extended a mixed membership SBM for static networks to dynamic networks by characterizing the evolving community memberships and block connection probabilities with a state space model. Both Yang et al. (2011) and Xu and Hero (2014) studied a sequence of SBMs, where the parameters were dynamically linked by a hidden Markov chain. Matias and Miele (2017) applied Markov chains to the evolution of the node community labels over time. Bhattacharjee et al. (2020) proposed a method to detect a single change point such that the community connection probabilities are different constants within the two intervals separated by it. Zhang and Cao (2017) investigate functional modules in gene regulation networks, assuming fixed community assignments over time while acknowledging dynamic interactions within and between communities. Xin et al. (2017) characterized the occurrence of a connection between any two nodes in an SBM using an inhomogeneous Poisson process. Zhang et al. (2020) proposed a regularization method for estimating the network parameters at adjacent time points to achieve smoothness.

4.1.3 Our Contributions

The main contribution of this chapter is to extend the classical SBM to address the three challenges mentioned above. Given the community membership of each node, we generalize the assumption that edges in the network follow Bernoulli distributions to that they follow compound Poisson-Gamma distributions instead (Section 4.2). This allows us to model edges that can take on any non-negative real value, including exactly zero itself. Later in Section 4.6, we apply the proposed model to an international trading network, where each edge between two countries represents the dollar amount of their trading values, for which our model is more appropriate than the classical one. Moreover, not only do we incorporate nodal information in the form of covariates, we also allow the effects of these covariates to be time-varying (Section 4.2).

We use a variational approach (Section 4.4) to conduct statistical inference for such a time-varying network. We also prove an interesting result (Section 4.3) that, asymptotically, the covariate effects in our model can be estimated irrespective of how community labels are assigned to each node. This result also allows us to use an efficient two-step algorithm (Section 4.4), separating the estimation of the covariate effects and that of the other parameters—including the unknown community labels. A similar two-step procedure is also used by Huang et al. (2023).

4.2 Methodology

In this section, we first give a brief review of a rarely-used distribution, the Tweedie distribution, which can be used to model network edges with zero or positive continuous weights. Next, we propose a general SBM using the Tweedie distribution in three successive steps, each addressing a challenge mentioned in Section 4.1.2. More specifically, we start with a vanilla model, a variation of the classic SBM where each edge value between two nodes now follows the Tweedie distribution rather than the Bernoulli distribution. We then incorporate covariate terms into the model, before we finally arrive at a time-varying version of the model by allowing the covariates to have dynamic effects that change over time.

4.2.1 Tweedie Distribution

Let N be a random variable following the Poisson distribution with mean λ. Conditional on $N = n$, $Z_1, \ldots, Z_n \overset{iid}{\sim} \text{Gamma}(\alpha, \gamma)$. Define

$$Y = \begin{cases} 0, & \text{if} \quad \text{N} = 0, \\ Z_1 + Z_2 + \cdots + Z_N, & \text{if} \quad \text{N} = 1, 2, 3, \cdots. \end{cases}$$

Then, Y has a compound Poisson-gamma distribution, with a nonzero probability mass at 0. As $Y = 0$ if and only if $N = 0$, $\mathbb{P}(Y = 0) = \mathbb{P}(N = 0) = \exp(-\lambda)$. Conditional on $N = n > 0$, Y follows a gamma distribution with mean $n\alpha\gamma$ and variance $n\alpha\gamma^2$. In the context of international trading (also see Section 4.6 below), N may be the number of trades in a given year; $Z_1, \ldots, Z_N$ may be the dollar amount of each trade; then, Y is the simply total trading amount from that year.

The compound Poisson-gamma distribution, known as a special case of the Tweedie distribution (Tweedie, 1984), is related to an exponential dispersion (ED) family. If Y follows an ED family distribution with mean μ and variance function V, then Y satisfies $\text{var}(Y) = \phi V(\mu)$ for some dispersion parameter ϕ. The Tweedie distribution belongs to the ED family with $V(\mu) = \mu^\rho$ for some constant ρ. Specified by different values of ρ, the Tweedie distribution includes the normal ($\rho = 0$), the gamma ($\rho = 2$) and the inverse Gaussian distribution ($\rho = 3$), and the scaled Poisson distribution ($\rho = 1$). Tweedie distributions exist for all values of ρ outside the interval $(0, 1)$. Of special interest to us here is the restricted Tweedie distribution with $1 < \rho < 2$, which is the aforementioned compound Poisson–gamma distribution with a positive mass at zero but a continuous distribution of positive values elsewhere. We add the word "restricted" to describe the Tweedie distribution when ρ is constrained to lie on the interval $(1, 2)$; it will become clearer later in Section 4.4 that this particular restriction also simplifies the overall estimation procedure somewhat.

Specifically, the aforementioned compound Poisson–gamma distribution with parameters $(\lambda, \alpha, \gamma)$ can be reparameterized as a restricted Tweedie distribution, with parameters (μ, ϕ, ρ) satisfying $1 < \rho < 2$ and the following relationships:

$$\lambda = \frac{\mu^{2-\rho}}{\phi(2-\rho)}, \quad \alpha = \frac{2-\rho}{\rho-1}, \quad \gamma = \phi(\rho-1)\mu^{\rho-1}.$$

That is, the marginal distribution of Y, defined above, can be expressed as

$$f(y|\mu, \phi, \rho) = a(y, \phi, \rho) \cdot \exp\left\{\frac{1}{\phi}\left(\frac{y\mu^{1-\rho}}{1-\rho} - \frac{\mu^{2-\rho}}{2-\rho}\right)\right\}, \quad 1 < \rho < 2, \tag{4.2}$$

where

$$a(y, \phi, \rho) = \begin{cases} \dfrac{1}{y}\displaystyle\sum_{j=1}^{\infty} \frac{y^{j\alpha}}{(\rho-1)^{j\alpha}\phi^{j(1+\alpha)}(2-\rho)^{j} j!\,\Gamma(j\alpha)}, & \text{for} \quad \mathrm{y} > 0, \\ 1, & \text{for} \quad \mathrm{y} = 0. \end{cases}$$

4.2.2 Vanilla Model

Let $\mathcal{G} = (V, E)$ denote a weighted graph, where V denotes a set of nodes with cardinality $|V| = n$ and E denotes the set of edges between two nodes. For SBMs, each node in the network can belong to one of K groups. Let $c_i \in \{1, \cdots, K\}$ denote the unobserved community membership of node i and c_i follows a multinomial distribution with the probability $\pi = (\pi_1, \cdots, \pi_K)$.

Usually, the set E is represented by an $n \times n$ matrix $Y = [y_{ij}] \in \mathbb{R}^{n\times n}$. In classical SBMs, each y_{ij} is modelled either as a Bernoulli random variable taking on binary values of 0 or 1, or as a Poisson random variable taking on non-negative *integer* values. We first relax this restriction by allowing y_{ij} to take on non-negative *real* values. Since we focus on an undirected weighted network without self-loops, Y is a (for us, non-negative) real-valued symmetric matrix with zero diagonal entries.

Given the observed data set $D = \{y_{ij}\}_{1\le i<j\le n}$, we assume that each edge value y_{ij} follows a restricted Tweedie distribution with power $\rho \in (1, 2)$ and dispersion ϕ:

$$y_{ij} \sim \mathrm{Tw}(\mu_{ij}, \phi, \rho), \quad 1 < \rho < 2, \tag{4.3}$$

where the mean μ_{ij} is modelled as a positive constant determined by the latent community label of nodes i and j through a log-link function, i.e.,

$$\log(\mu_{ij}) = \beta_0^{kl}, \quad \text{if} \quad c_i = k \quad \text{and} \quad c_j = l, \tag{4.4}$$

where $\beta_0 = [\beta_0^{kl}] \in \mathbb{R}^{K\times K}$ is a symmetric matrix. For a constant model, the log-link may not appear to be necessary, but it will become more useful later on as we incorporate covariates into this baseline model.

4.2.3 Model with Covariates

In many real-life situations, we observe additional information about the network. For example, in addition to the relative existence or importance of each edge, a collection of p symmetric covariate matrices $X^{(1)}, ..., X^{(p)} \in \mathbb{R}^{n \times n}$ may also be available, where the (i, j)-th entry $x_{ij}^{(u)}$ of each $X^{(u)}$ represents a pair-wise covariate containing some information about the connection between node i and node j, and $x_{ii}^{(u)} = 0$ for all $1 \leq i \leq n$ and $u = 1, ..., p$. Given a data set $D = \{Y, X^{(1)}, ..., X^{(p)}\}$, the vanilla model from Section 4.2.2 above can be easily extended by replacing (4.4) with

$$\log(\mu_{ij}) = \beta_0^{kl} + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}, \quad \text{if} \quad c_i = k \quad \text{and} \quad c_j = l, \tag{4.5}$$

so that μ_{ij} is affected not only by the community labels c_i, c_j but also by the covariates contained in $\boldsymbol{x}_{ij}$. Here, both $\boldsymbol{x}_{ij} \equiv (x_{ij}^{(1)}, ..., x_{ij}^{(p)})^\top$ and $\boldsymbol{\beta}$ are p-dimensional vectors.

4.2.4 Time-varying Model

Now suppose we observe an evolving network at a series of T discrete time points $\{t_1, \cdots, t_T\}$, with a common set of n nodes. Specifically, our data set is of the form $D = \{Y(t_1), \ldots, Y(t_T); X^{(1)}, \ldots, X^{(p)}\}$. Without loss of generality, we may assume each $t_\nu \in [0, 1]$.

To model such data, we assume in this chapter that the latent community labels $c_1, \ldots, c_n$ are fixed over time but allow the covariate effects to change over time by incorporating a varying-coefficient model. In reality, the community labels may also change over time, but a fundamentally different set of tools will be required to model these changes and we will study them separately—not in this chapter. Here, we simply assume that model (4.3) holds pointwise at every time point t, i.e.,

$$y_{ij}(t) \sim \text{Tw}(\mu_{ij}(t), \phi, \rho), \quad 1 < \rho < 2, \tag{4.6}$$

and

$$\log\{\mu_{ij}(t)\} = \beta_0^{kl} + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t), \quad \text{if} \quad c_i = k \quad \text{and} \quad c_j = l, \tag{4.7}$$

where $\boldsymbol{\beta}(t) \equiv (\beta_1(t), \ldots, \beta_p(t))^\top$ and each $\beta_u(t)$ is a smooth function of time. The full likelihood function corresponding to our time-varying model (4.6)–(4.7) is given by

$$L(\beta_0, \boldsymbol{\beta}(t), \pi, \phi, \rho; D, c) = \prod_{i=1}^{n} \prod_{k=1}^{K} \pi_k^{\mathbb{1}(c_i=k)} \prod_{\nu=1}^{T} \prod_{1 \le i < j \le n} \prod_{k,l=1}^{K} \Bigg[a(y_{ij}(t_\nu), \phi, \rho) \times \exp\left\{ \frac{1}{\phi} \left(\frac{y_{ij}(t_\nu) \exp[(1-\rho)\{\beta_0^{kl} + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}]}{1-\rho} - \frac{\exp[(2-\rho)\{\beta_0^{kl} + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}]}{2-\rho} \right) \right\} \Bigg]^{\mathbb{1}(c_i=k, c_j=l)} . \quad (4.8)$$

The likelihood functions for the earlier, simpler models—namely, the vanilla model in Section 4.2.2 and the static model with covariates in Section 4.2.3—are simply special cases of (4.8).

4.3 Theory

The resulting log-likelihood based on (4.8) contains three additive terms: the first involves only π; the second involves only (ϕ, ρ); and the third is the only one that involves both β_0 and $\boldsymbol{\beta}(t)$. Define

$$\ell_n(\boldsymbol{\beta}(t), \phi_0, \rho_0; D, z) \equiv \frac{1}{\binom{n}{2}} \sum_{\nu=1}^{T} \sum_{1 \le i < j \le n} \sum_{k,l=1}^{K} \frac{\mathbb{1}(z_i = k, z_j = l)}{\phi_0} \times \left(\frac{y_{ij}(t_\nu) \exp[(1-\rho_0)\{\hat{\beta}_0^{kl}(\boldsymbol{\beta}(t_\nu)) + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}]}{1-\rho_0} - \frac{\exp[(2-\rho_0)\{\hat{\beta}_0^{kl}(\boldsymbol{\beta}(t_\nu)) + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}]}{2-\rho_0} \right) \quad (4.9)$$

to be the aforementioned third term *after having*

- replaced the unknown labels $c = (c_1, \ldots, c_n)$ with an arbitrary set of labels $z = (z_1, \ldots, z_n)$, where each z_i is independently multinomial$(p_1, \ldots, p_K)$;
- profiled out the parameter β_0 by replacing it with $\hat{\beta}_0(\boldsymbol{\beta}(t))$, while presuming $\phi = \phi_0$ and $\rho = \rho_0$ to be known and fixed; and

- re-scaled it by the total number of pairs, $\binom{n}{2}$.

This quantity turns out to be very interesting. Not only does $\hat{\beta}_0(\boldsymbol{\beta}(t))$ have an explicit expression, but (4.9) can also be shown to converge to a quantity *not* dependent on z as n tends to infinity.

In other words, it does *not* matter that z is a set of *arbitrarily* assigned labels! This has immediate computational implications (see Section 4.4). Some high-level details of this theory are spelled out below in Section 4.3.1, while actual proofs are given in the Appendix.

4.3.1 Details

To simplify the notation, we first define two population parameters,

$$\theta = \sum_{\nu=1}^{T} \mathbb{E}[y_{ij}(t_\nu)\exp\{(1-\rho_0)\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}] \quad \text{and} \quad \gamma = \sum_{\nu=1}^{T} \mathbb{E}[\exp\{(2-\rho_0)\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}].$$

For these to be properly defined, we require the following two conditions, which are fairly standard and not fundamentally restrictive.

Condition 4.3.1. *The covariates $\{\boldsymbol{x}_{ij}, 1 \le i < j \le n\}$ are i.i.d., and there exists some $\alpha > 0$ such that $\mathbb{P}(\exp\{\boldsymbol{x}_{ij}^\top \boldsymbol{u}\} \ge \delta) \le 2\exp(-\delta^2/\alpha)$ for any $\delta > 0$, $i \ne j$ and $\boldsymbol{u} \in \mathbb{R}^p$ satisfying $\|\boldsymbol{u}\|_2 = \sqrt{u_1^2 + \cdots + u_p^2} = 1$.*

Condition 4.3.2. *The function $\beta_u(t)$ is continuous on $[0,1]$, for all $u = 1, \ldots, p$.*

The corresponding empirical versions of θ and γ between any two groups, k and l, according to an arbitrary community label assignment, z, are given by

$$\hat{\theta}_{kl} = \frac{1}{\binom{n}{2}} \sum_{\nu=1}^{T} \sum_{1 \le i < j \le n} y_{ij}(t_\nu) \exp[(1-\rho_0)\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)] \mathbb{1}(z_i = k, z_j = l),$$

$$\hat{\gamma}_{kl} = \frac{1}{\binom{n}{2}} \sum_{\nu=1}^{T} \sum_{1 \le i < j \le n} \exp[(2-\rho_0)\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)] \mathbb{1}(z_i = k, z_j = l).$$

We can then establish the following main theorem.

Theorem S1. *Theorem 1. As $n \to \infty$ while K remains constant,*

$$\ell_n(\boldsymbol{\beta}(t), \phi_0, \rho_0; D, z) = \frac{1}{\phi_0} \frac{1}{(1-\rho_0)(2-\rho_0)} \sum_{k,l=1}^{K} \hat{\theta}_{kl}^{2-\rho_0} \cdot \hat{\gamma}_{kl}^{\rho_0 - 1}$$
$$= \frac{1}{\phi_0} \frac{1}{(1-\rho_0)(2-\rho_0)} \theta^{2-\rho_0} \cdot \gamma^{\rho_0 - 1} + o_p(1). \quad (4.10)$$

Remark 1. *So far, we have simply written $\hat{\theta}_{kl}$, $\hat{\gamma}_{kl}$, θ and γ in order to keep the notation short. To better appreciate the conclusion of the theorem, however, it is perhaps important for us to emphasize here that these quantities are more properly written as $\hat{\theta}_{kl}(\boldsymbol{\beta}(t), \rho_0; D, z)$, $\hat{\gamma}_{kl}(\boldsymbol{\beta}(t), \rho_0; D, z)$, $\theta(\boldsymbol{\beta}(t), \rho_0; D)$, and $\gamma(\boldsymbol{\beta}(t), \rho_0; D)$.*

The implication here is that, asymptotically, our inference about $\boldsymbol{\beta}(t)$ is not affected by the community labels—nor is it affected by the total number of communities, K, since z can follow *any* multinomial$(p_1, \ldots, p_K)$ distribution, including those with some $p_k = 0$. Thus, even if we got K wrong, our inference about $\boldsymbol{\beta}(t)$ would still be correct.

4.4 Estimation Method

4.4.1 Two-step Estimation

In this section, we outline an algorithm to fit the restricted Tweedie SBM. Since, for us, the parameter ρ is restricted to the interval $(1, 2)$, we find it sufficient to simply perform a grid search (e.g., Dunn and Smyth, 2005, 2008; Lian et al., 2023) over an equally-spaced sequence, say, $1 < \rho_1 < \cdots < \rho_m < 2$, to determine its "optimal" value. However, our empirical experiences also indicate that a sufficiently accurate estimate of ρ is important for making correct inferences on other quantities of interest, including the latent community labels c.

For any given ρ_0 in a pre-specified sequence/grid, we propose an efficient two-step algorithm to estimate the other parameters. In Step 1 (Section 4.4.1), we obtain an estimate $\hat{\boldsymbol{\beta}}_{\rho_0}(t)$ of $\boldsymbol{\beta}(t)$ using an arbitrary set of community labels. This is made possible by the theoretical result earlier in Section 4.3. In Step 2 (Section 4.4.1), we obtain estimates of the remaining parameters parameters—$\hat{\beta}_0(\rho_0), \hat{\pi}(\rho_o), \hat{\phi}(\rho_0)$—while keeping $\hat{\boldsymbol{\beta}}_{\rho_0}(t)$ fixed.

The optimal ρ is then chosen to be

$$\hat{\rho} = \underset{\rho_0 \in \{\rho_1, \cdots, \rho_m\}}{\arg\max}\ L(\hat{\beta}_0(\rho_0), \hat{\boldsymbol{\beta}}_{\rho_0}(t), \hat{\pi}(\rho_0), \hat{\phi}(\rho_0), \rho_0; D, c).$$

Step 1: Estimation of Covariates Coefficients

It is clear from our earlier theoretical result in Section 4.3 that, when $\rho = \rho_0$ is given and fixed, the quantity (4.9) can be used directly as a criterion to estimate $\boldsymbol{\beta}(t)$. To begin, here one can fix the parameter ϕ at $\phi_0 = 1$, since it only appears as a scaling constant in (4.9) and does not affect the optimum. The main computational saving afforded by Theorem 1 is that we can use an *arbitrary* set of labels z to carry out this step, estimating $\boldsymbol{\beta}(t)$ separately without simultaneously concerning ourselves with β_0 or having to make inference on c. Both of those tasks can be temporarily delayed until after $\boldsymbol{\beta}(t)$ is estimated.

For our static model (Section 4.2.3), we use the `optim` function in R to maximize (4.9) directly over $\boldsymbol{\beta}$, with $T = 1$. For our time-varying model (Section 4.2.4), we add (component-wise) smoothness penalties to (4.9) and estimate $\boldsymbol{\beta}(t)$ as

$$\hat{\boldsymbol{\beta}}_{\rho_0}(t) = \underset{\boldsymbol{\beta}(t)}{\arg\max}\ \ell_n(\boldsymbol{\beta}(t), 1, \rho_0; D, z) - \frac{1}{2}\sum_{u=1}^{p} \lambda_u \cdot \int \{\boldsymbol{\beta}_u''(t)\}^2 dt. \tag{4.11}$$

The penalty parameters $\lambda_1, \ldots, \lambda_p$ are chosen by cross-validation (see Section 4.4.2 below). With given penalty parameters, technical details for calculating (4.11) are provided in the Appendix.

Step 2: Variational Inference

In Step 2, with the estimate $\hat{\boldsymbol{\beta}}_{\rho_0}(t)$ from Step 1 (and, again, a pre-fixed $\rho = \rho_0$), we estimate the remaining parameters β_0, π, and ϕ, as well as make inferences about the latent label c.

If we directly optimized the likelihood function (4.8) using the EM algorithm, the E-step would require us to compute $\mathbb{E}_{c|D}(\cdot)$ but, here, the conditional distribution of the latent variable c given D is complicated because c_i and c_j are not conditionally independent in general. We will use a variational approach instead.

To proceed, it will be more natural for us to emphasize the fact that (4.8) is really just the joint distribution of (D, c). Thus, instead of writing it as $L(\beta_0, \boldsymbol{\beta}(t), \pi, \phi, \rho; D, c)$, in this section we will write it simply as $\mathbb{P}(D, c; \beta_0, \pi, \phi)$, where we have also dropped $\boldsymbol{\beta}(t)$

and ρ to keep the notation short because, within this step, $\rho = \rho_0$ and $\boldsymbol{\beta}(t) = \hat{\boldsymbol{\beta}}_{\rho_0}(t)$ are both fixed and not being estimated.

Ideally, since the latent variable c is not observable, one may want to work with the marginal distribution of D and estimate (β_0, π, ϕ) as:

$$\begin{aligned}\left(\hat{\beta}_0, \hat{\pi}, \hat{\phi}\right) &= \underset{\beta_0,\pi,\phi}{\arg\max} \log \mathbb{P}(D; \beta_0, \pi, \phi) \\ &= \underset{\beta_0,\pi,\phi}{\arg\max}\ \log \sum_{c \in [K]^n} \mathbb{P}(D, c; \beta_0, \pi, \phi),\end{aligned} \tag{4.12}$$

but this is difficult due to the summation over K^n terms. The key idea of variational inference is to approximate $\mathbb{P}(c|D; \beta_0, \pi, \phi)$ with a distribution $q(c)$ from a more tractable family—also referred to as the "variational distribution" in this context—and to decompose the objective function in (4.12) into two terms:

$$\begin{aligned}&\log \mathbb{P}(D; \beta_0, \pi, \phi) \\ =& \sum_{c \in [K]^n} (\log \mathbb{P}(D; \beta_0, \pi, \phi)) \cdot q(c) \\ =& \sum_{c \in [K]^n} \left(\log \frac{\mathbb{P}(D; \beta_0, \pi, \phi) \cdot q(c)}{\mathbb{P}(D, c; \beta_0, \pi, \phi)} + \log \frac{\mathbb{P}(D, c; \beta_0, \pi, \phi)}{q(c)} \right) \cdot q(c) \\ =& \underbrace{\mathbb{E}_q \left[\log \frac{q(c)}{\mathbb{P}(c|D; \beta_0, \pi, \phi)} \right]}_{\text{KL}} + \underbrace{\mathbb{E}_q \left[\log \frac{\mathbb{P}(D, c; \beta_0, \pi, \phi)}{q(c)} \right]}_{\text{ELBO}}.\end{aligned} \tag{4.13}$$

The first term in (4.13) can be recognized as the Kullback–Leibler (KL) divergence between $q(c)$ and $\mathbb{P}(c|D; \cdot)$, which is non-negative. This makes the second term in (4.13) a lower bound of objective function. It is referred to in the literature as the "evidence lower bound" (ELBO), and is equal to the objective function itself when the first term is zero, i.e., when $q(c) = \mathbb{P}(c|D; \cdot)$.

So, instead of maximizing (4.12) directly, one maximizes the ELBO term—not only over (β_0, π, ϕ), but also over q. Since the original objective function—that is, the left-hand side of (4.13)—does not depend on q, maximizing the ELBO term over q is also equivalent to minimizing the KL term. And when the KL term is small, not only is the variational distribution $q(c)$ close to $\mathbb{P}(c|D; \cdot)$, but the ELBO term is also automatically close to the original objective, which justifies why this approach often gives a good approximate solution to the otherwise intractable problem (4.12) and why the variational distribution

$q(c) \approx \mathbb{P}(c|D;\cdot)$ can be used to make approximate inferences about c.

Since the decomposition (4.13) holds for any q, in practice one usually chooses it from a "convenient" family of distributions so that $\mathbb{E}_q(\cdot)$ is easy to compute. In particular, we can choose

$$q(c) = \prod_{i=1}^{n} q_i(c_i)$$

to be a completely factorizable distribution; here, each q_i is simply a standalone multinomial distribution with probability vector $(\tau_{i1}, \cdots, \tau_{iK})$. Under this choice, $\mathbb{E}_q[\mathbb{1}(c_i = k)] = \tau_{ik}$, $\mathbb{E}_q[\mathbb{1}(c_i = k, c_j = l)] = \tau_{ik}\tau_{jl}$, and the ELBO term in (4.13) is simply

$$\begin{aligned} \text{ELBO}(\tau, \beta_0, \pi, \phi; D) = &\sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik} \cdot \log(\pi_k) + \sum_{\nu=1}^{T} \sum_{1 \le i < j \le n} \log a(y_{ij}(t_\nu), \phi, \rho_0) \\ &+ \sum_{\nu=1}^{T} \sum_{1 \le i < j \le n} \sum_{k,l=1}^{K} \frac{\tau_{ik}\tau_{jl}}{\phi} \Bigg(y_{ij}(t_\nu) \frac{\exp\left[(1-\rho_0)\{\beta_0^{kl} + x_{ij}^\top \hat{\boldsymbol{\beta}}_{\rho_0}(t_\nu)\}\right]}{1-\rho_0} \\ &- \frac{\exp\left[(2-\rho_0)\{\beta_0^{kl} + x_{ij}^\top \hat{\boldsymbol{\beta}}_{\rho_0}(t_\nu)\}\right]}{2-\rho_0} \Bigg) - \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik} \cdot \log(\tau_{ik}), \end{aligned} \tag{4.14}$$

which is easy to maximize in a coordinate-wise fashion, i.e., successively over τ, β_0, π and ϕ.

The maxima of (4.14) with respect to τ and π is found by the method of Lagrange multipliers respectively, as the according optimization problem is subject to equality constraints $\sum_{k=1}^{K} \tau_{ik} = 1$ and $\sum_{k=1}^{K} \pi_k = 1$ for any i respectively. Specifically, at iteration step h

$$\tau_{ik}^{(h)} = \frac{f_{ik}}{\sum\limits_{k=1}^{K} f_{ik}}, \quad k = 1, \cdots, K \text{ and } i = 1, \cdots, n,$$

where

$$\begin{aligned} f_{ik} = &\pi_k^{(h-1)} \cdot \exp\Bigg[\sum_{\nu=1}^{T} \sum_{j \ne i} \sum_{l=1}^{K} \frac{\tau_{jl}^{(h-1)}}{\phi^{(h-1)}} \Bigg\{ y_{ij}(t) \frac{\exp[(1-\rho_0)\{(\beta_0^{kl})^{(h-1)} + x_{ij}^\top \hat{\boldsymbol{\beta}}_{\rho_0}(t_\nu)\}]}{1-\rho_0} \\ &- \frac{\exp[(2-\rho_0)\{(\beta_0^{kl})^{(h-1)} + x_{ij}^\top \hat{\boldsymbol{\beta}}_{\rho_0}(t_\nu)\}]}{2-\rho_0} \Bigg\}\Bigg], \end{aligned}$$

and

$$\pi_k^{(h)} = \frac{\sum\limits_{i=1}^{n} \tau_{ik}^{(h)}}{n}, \; k = 1, \cdots, K.$$

The objective function (4.14) is concave down in β_0^{kl} for each community label pair $k-l$, which allows the zeros of the first derivative of (4.14) to be its maxima. We update $\beta_0^{(h-1)}$ to $\beta_0^{(h)}$ by solving the equation $\frac{\partial}{\partial \beta_0}\mathrm{ELBO}(\tau^{(h)}, \beta_0, \pi^{(h)}, \phi; D) = 0$ analytically for each pair k-l, which can be implemented in one step:

$$(\beta_0^{kl})^{(h)} = \log \frac{\sum\limits_{\nu=1}^{T} \sum\limits_{1\leq i<j\leq n} y_{ij}(t)\exp[(1-\rho_0)x_{ij}^{\top}\hat{\boldsymbol{\beta}}_{\rho_0}(t_\nu)] \cdot \tau_{ik}^{(h)}\tau_{jl}^{(h)}}{\sum\limits_{\nu=1}^{T} \sum\limits_{1\leq i<j\leq n} \exp[(2-\rho_0)x_{ij}^{\top}\hat{\boldsymbol{\beta}}_{\rho_0}(t_\nu)] \cdot \tau_{ik}^{(h)}\tau_{jl}^{(h)}}.$$

With $\pi_k^{(h)}$, $\tau_{ik}^{(h)}$ and $\beta_0^{(h)}$ fixed, we can now directly maximize the ELBO term (4.14) over ϕ to update it in principle. However, the function $a(y_{ij}(t_\nu), \phi, \rho_0)$ is "a bit of a headache" to compute, so we use the R package `tweedie` by Dunn and Smyth (2005, 2008) that computes (4.2) for us, and update ϕ by letting $c_i^{(h)} = \arg\max_k \tau_{ik}^{(h)}$ and maximizing over the original log-likelihood function instead, i.e.,

$$\phi^{(h)} = \arg\max_{\phi} \; \log \mathbb{P}(D, c^{(h)}; \beta_0^{(h)}, \pi^{(h)}, \phi). \tag{4.15}$$

We do this directly using the R function `optim`.

4.4.2 Tuning Parameter Selection

We adapt the leave-one-out cross validation to choose the tuning parameter λ when fitting our model. In particular, each time we utilize observations made at $T-1$ time points to train the model and then test the trained model on the observations made at the remaining time points. To avoid boundary effects, our leave-one-out procedure is repeated for only $T-2$ times (as opposed to the usual T times), because we always retain the observations at times t_1 and t_T in the training set—only those at times $t_2, \ldots, t_{T-1}$ are used (one at a time) as test points. In our implementations, the loss is defined as the negative log-likelihood of the fitted model, and the overall loss is taken as the average across the $T-2$ repeats. We select the “optimal” λ that gives rise to the smallest loss.

4.5 Simulation

In this section, we present simulation results to validate the performance of our restricted Tweedie SBM. We do so in successive steps—from the vanilla model (Section 4.5.1), to the static model with covariates (Section 4.5.2), and finally, the most general, time-varying version of the model (Section 4.5.3).

We mainly focus on two aspects of the results, the clustering quality and the accuracy of the estimated covariate effects. We measure the latter by the mean squared error, and the former by a metric called “normalized mutual information” (NMI) (Danon et al., 2005), which ranges in $[0, 1]$, with values closer to 1 indicating better agreements between the estimated community labels and the true ones.

For all simulations, we fix the true number of communities to be $K = 3$, with prior probabilities $\pi = (0.2, 0.3, 0.5)$. For the true matrix β_0, we set all diagonal entries β_0^{kk} to be equal, and all off-diagonal entries β_0^{kl} to be equal as well—so the entire matrix is completely specified by just two numbers.

To avoid getting stuck at poor local optima, we use multiple initial values in each run.

4.5.1 Simulation of Vanilla Model

First, we assess the performance of our vanilla model (Section 4.2.2), and compare it with the Poisson SBM and spectral clustering. The Poisson SBM assumes the edges follow Poisson distributions; we simply round each y_{ij} into an integer and use the function `estimateSimpleSBM` in R package `sbm` to fit it. To run spectral clustering, we use the function `reg.SSP` from the R package `randnet`. The function `estimateSimpleSBM` uses results from a bipartite SBM as its initial values. To make a more informative comparison, we use two different initialization strategies to fit our model: (i) starting from 30 sets of randomly drawn community labels and picking the best solution afterwards, and (ii) starting from the Poisson SBM result itself.

We generate Y using nine different combinations of (ϕ, ρ) with $\phi = 0.5, 1, 2$ and $\rho =$

$1.2, 1.5, 1.8$, and three different β_0 matrices:

$$\begin{aligned}
&\text{scenario 1, } (\beta_0^{kk}, \beta_0^{kl}) = (1.0, \quad 0.0) \Rightarrow \exp(\beta_0^{kk}) - \exp(\beta_0^{kl}) \approx 1.72;\\
&\text{scenario 2, } (\beta_0^{kk}, \beta_0^{kl}) = (0.5, -0.5) \Rightarrow \exp(\beta_0^{kk}) - \exp(\beta_0^{kl}) \approx 1.04;\\
&\text{scenario 3, } (\beta_0^{kk}, \beta_0^{kl}) = (0.0, -1.0) \Rightarrow \exp(\beta_0^{kk}) - \exp(\beta_0^{kl}) \approx 0.63.
\end{aligned}$$

According to the discrepancy in μ_{ij} between (i, j)-pairs belonging to the same group and those belonging to different groups, the clustering difficulty of the three designs can be roughly ordered as: scenario 1 < scenario 2 < scenario 3.

Table 4.1, 4.2, and 4.3 summarize the averages and the standard errors of the NMI metric for different methods over 50 simulation runs, respectively for scenarios 1, 2 and 3. As expected, all methods perform the best in scenario 1 and the worst in scenario 3. Their performances also improve when the sample size n increases, and as the parameter ϕ decreases—as the dispersion parameter, a smaller ϕ means a reduced variance and an easier problem.

Overall, our restricted Tweedie SBM and the Poisson SBM tend to outperform spectral clustering. Among all 54 sets of simulation results, our model with random initialization compares favorably with other methods in 50 of them. In the remaining four sets (marked by a superscript "†" in the tables), the Poisson SBM is slightly better, but we could still outperform it in three of them and match it in the other if we initialized our algorithm with the Poisson SBM result itself. It is evident in all cases that our restricted Tweedie SBM can further improve the clustering result of the Poisson SBM.

4.5.2 Simulation of Model with Covariates

Next, we study our static model with covariates (Section 4.2.3). We use exactly the same combination of ϕ, ρ and n as we did previously in Section 4.5.1, but only scenario 2 for the matrix β_0—the one with medium difficulty—for conciseness.

For the covariates, we take $p = 1$ so there is just one scalar covariate x_{ij}, which we generate independently for each (i, j)-pair from the uniform distributions on $(-1, 1)$. The true covariate effect β is simulated to be either weak ($\beta = 1$) or strong ($\beta = 2$).

Table 4.4 summarizes the results. Clearly, if there is a covariate x_{ij} affecting the outcome y_{ij}, not taking it into account (and simply fitting a vanilla model) will significantly affect the clustering result, as measured by the NMI metric. On the other hand, the mean and standard error of the estimate $\hat{\beta}$ over repeated simulation runs clearly validate the correctness of Theorem S1 and the effectiveness of our two-step algorithm—the covariate effects can indeed be estimated quite well with arbitrarily assigned community labels.

4.5.3 Simulation of Time-varying Model

We now study the most general, time-varying version of our model (Section 4.2.4), having already established empirical evidence for the usefulness of the restricted Tweedie model in its vanilla form (Section 4.5.1) and the importance of taking covariates into account in a static setting (Section 4.5.2).

Instead of different combinations of (ϕ, ρ, n), these are now fixed at $\phi = 1$, $\rho = 1.5$, and $n = 50$. But we introduce three more scenarios for the true matrix β_0:

$$\begin{aligned}
&\text{scenario 4, } (\beta_0^{kk}, \beta_0^{kl}) = (0.50, \quad 0.00) \Rightarrow \exp(\beta_0^{kk}) - \exp(\beta_0^{kl}) \approx 0.65;\\
&\text{scenario 5, } (\beta_0^{kk}, \beta_0^{kl}) = (0.25, -0.25) \Rightarrow \exp(\beta_0^{kk}) - \exp(\beta_0^{kl}) \approx 0.51;\\
&\text{scenario 6, } (\beta_0^{kk}, \beta_0^{kl}) = (0.00, -0.50) \Rightarrow \exp(\beta_0^{kk}) - \exp(\beta_0^{kl}) \approx 0.39.
\end{aligned}$$

These are similar to the earlier scenarios 1, 2 and 3, but respectively more difficult to cluster.

We generate one scalar covariate x_{ij} in exactly the same way as we did in Section 4.5.2, except that its effect is now time-varying, with coefficient $\beta(t)$ generated in six different ways: (i) $\beta(t) = 2t - 1$, (ii) $\beta(t) = \sin(2\pi t)$, (iii) $\beta(t) = 2t$, (iiii) $\beta(t) = \sin(2\pi t) + 1$, (v) $\beta(t) = 0.5(2t - 1)$, and (vi) $\beta(t) = 0.5 \sin(2\pi t)$. Finally, the data sets are simulated in such a way that the network is observed at $T = 20$ equally spaced time points on $[0, 1]$.

We use 10 different sets of initial values for each simulation run. To evaluate the performance of the estimated $\hat{\beta}(t)$, we calculate the estimation error as

$$\mathrm{Err}(\hat{\beta}(t)) = \frac{1}{20} \sum_{\nu=1}^{20} [\hat{\beta}(t_\nu) - \beta(t_\nu)]^2.$$

In general, the tuning parameter λ is to be selected by cross-validation (see Section 4.4.2). To reduce computational cost, we simply fix it at $\lambda = 0.5$ for the current simulation study. Appendix B.4.2 contains a small sensitivity analysis using $\lambda = 0.1 < 0.5$ and $\lambda = 1.0 > 0.5$, from which one can see that it makes little difference whether $\lambda = 0.1$, 0.5 or 1.0 is used in this study.

For all simulated cases with different combinations of β_0 and $\beta(t)$, Table 4.5 summarizes the two metrics, NMI and $\text{Err}(\hat{\beta}(t))$, while Figure 4.2 displays the true function $\beta(t)$ together with the pointwise mean and standard deviation of $\hat{\beta}(t)$, over repeated simulation runs. The standard deviation is hard to visualize because it is very small at all t.

Theorem S1 again explains why the varying-coefficient $\beta(t)$ can be estimated so well. Once $\beta(t)$ has been estimated, the community structure is actually easier to detect with time-varying data than it is with static data because, for each pair (i, j), observations at all time points, $\{y_{ij}(t_\nu)\}_{\nu=1}^{T}$, contain this information, not just a single observation y_{ij}.

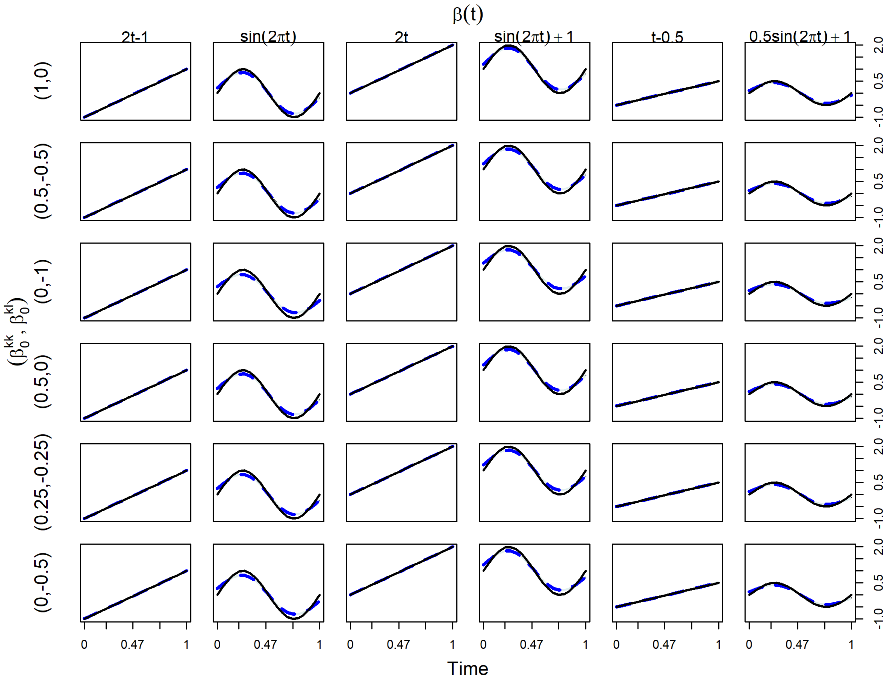


Figure 4.2: Estimations of $\beta(t)$ using a tuning parameter of $\lambda = 0.5$ for all simulated cases with different combinations of β_0 and $\beta(t)$. In each panel, the black solid line is the true function $\beta(t)$; the blue dashed line is the pointwise mean of $\hat{\beta}(t)$; and the light blue shadow (hardly visible) marks the corresponding pointwise confidence band.

4.6 Application: International Trading

In this section, we apply the restricted Tweedie SBM to study international trading relationships among different countries and how these relationships are influenced by geographical distances. As an example, we focus on the trading of apples—not only are these data readily available from the World Bank (World Integrated Trade Solution, 2023), but one can also surmise *a priori* that geographical distances will likely have a substantial impact on the trading due to the heavyweight and perishable nature of this product.

From the international trading data sets provided by the World Bank (World Integrated Trade Solution, 2023), we have collected annual import and export values of edible and fresh apples among $n = 66$ countries from $t_1 = 2002$ to $t_{20} = 2021$. In each given year t_ν, we observe a 66-by-66 matrix $Y(t_\nu)$ where each cell $y_{ij}(t_\nu)$ represents the trading value from country i to country j in thousands of US dollars during that year. We then average $Y(t_\nu)$ with its transpose to ensure symmetry. Finally, a small number of entries with values ranging from 0 to 1 (i.e., total trading values less than \$1,000) are thresholded to 0, and the remaining entries are logarithmically transformed. For the covariate x_{ij}, we use the shortest geographical distance between the two trading countries based on their borders, which we calculate using the R packages `maps` and `geosphere`.

We employ the cross-validation procedure outlined in Section 4.4.2 to choose the tuning parameter λ. Figure 4.3 displays the CV error, showing the optimal tuning parameter to be $\lambda^* = 0.1$.

Table 4.6 shows how the 66 countries are clustered into three communities by our method. Figure 4.4 displays the aggregated matrix, $Y(2002) + Y(2003) + \cdots + Y(2021)$, with rows and columns having been permuted according to the inferred community labels. Clearly, countries in the first community trade intensively with each other and with countries in the third community. While both the second and third communities consist of countries that mainly trade with countries in the first community (rather than among themselves or between each other), the trading intensity with the first community is lot higher for the third community than it is for the second.

Figure 4.5 displays $\hat{\beta}(t)$, the estimated effect of geographical distances on apple trading over time. We can make three prominent observations. First, the function $\hat{\beta}(t)$ is negative over the entire time period being studied—not surprising since longer distances can only increase the cost and time of transportation, and negatively impact fresh apple trading. Next, generally speaking the magnitude of $\hat{\beta}(t)$ is decreasing over the twenty-year period, implying that the negative effect of geographical distances is diminishing. This may be attributed to more efficient method and reduced cost of shipment overtime. Finally, two relatively big "dips" in $\hat{\beta}(t)$ are clearly visible—one after the financial crisis in 2008, and another after the onset of the Covid-19 pandemic in 2020.

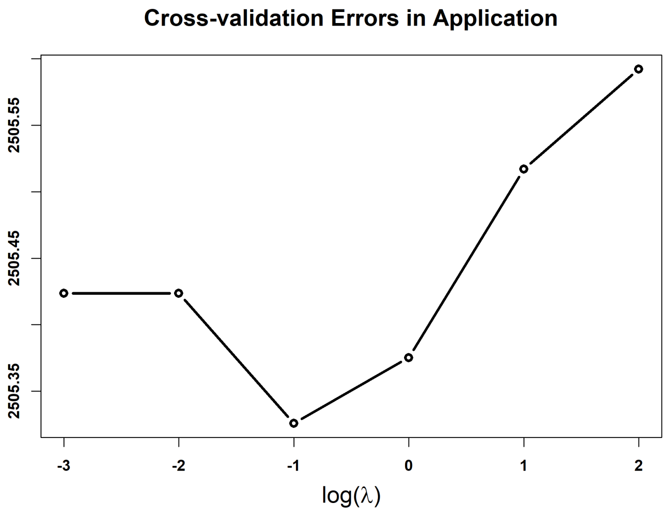


Figure 4.3: Cross validation errors change across a range of plausible values for the tuning parameter λ.

4.7 Conclusion

This chapter generalizes the vanilla SBM by replacing the Bernoulli distribution with the restricted Tweedie distribution to accommodate non-negative continuous edge weights with potential for zero values. Moreover, our model accounts for dynamic effects of nodal information. We show that as the number of nodes diverges to infinity, estimating the covariates coefficients is asymptotically irrelevant to the community labels when we maximize the likelihood function. This startling finding leads to the efficient two-step algorithm. Applying our framework to the international apple trading data provides insight into the dynamic effect of the geographic distance between countries in the trading network.

Moreover, simulation studies in Section 4.5 demonstrates the appealing performance of the proposed framework in clustering. This can be attributed to time independent community labels for each node, as the temporal data provide sufficient information for inferring the community labels. However, in many real world dynamic networks, the community label of each node is time dependent; it renders our current framework inapplicable. Xu and Hero (2014) and Matias and Miele (2017) proposed to use a Markov chain to address this problem, but there exist idenfitiability issues for parameters to be resolved in future work.

			Restricted Tweedie SBM		Poisson	Spectral
ϕ	ρ	n	Random Init.	Poisson Init.	SBM	Clustering
2	1.2	50	0.9097 (0.016)	0.8275 (0.023)	0.8099 (0.022)	0.5547 (0.012)
		100	0.9958 (0.002)	0.9958 (0.002)	0.9950 (0.002)	0.9185 (0.019)
	1.5	50	0.8647 (0.019)	0.7780 (0.02)	0.7275 (0.02)	0.5152 (0.012)
		100	0.9878 (0.003)	0.9878 (0.003)	0.9865 (0.003)	0.769 (0.025)
	1.8	50	0.7644 (0.017)	0.7180 (0.020)	0.6539 (0.02)	0.4857 (0.015)
		100	0.9828 (0.004)	0.9828 (0.004)	0.9826 (0.004)	0.6597 (0.015)
1	1.2	50	0.9918 (0.005)	0.9946 (0.004)	0.9880 (0.004)	0.7529 (0.027)
		100	1 (0)	1 (0)	1 (0)	1 (0)
	1.5	50	0.9778 (0.008)	0.9859 (0.006)	0.9745 (0.008)	0.7034 (0.023)
		100	1 (0)	1 (0)	1 (0)	0.9991 (0.001)
	1.8	50	0.9653 (0.01)	0.9644 (0.01)	0.9512 (0.012)	0.6702 (0.019)
		100	0.9992 (0.001)	0.9992 (0.001)	0.9992 (0.001)	0.9656 (0.013)
0.5	1.2	50	1 (0)	1 (0)	1 (0)	0.9934 (0.007)
		100	1 (0)	1 (0)	1 (0)	1 (0)
	1.5	50	1 (0)	1 (0)	1 (0)	0.9297 (0.019)
		100	1 (0)	1 (0)	1 (0)	1 (0)
	1.8	50	1 (0)	1 (0)	0.9985 (0.001)	0.8307 (0.025)
		100	1 (0)	1 (0)	1 (0)	1 (0)

Table 4.1: Summary of the NMI in scenario 1, $(\beta_0^{kk}, \beta_0^{kl}) = (1, 0)$, over 50 simulation runs.

			Restricted Tweedie SBM		Poisson	Spectral
ϕ	ρ	n	Random Init.	Poisson Init.	SBM	Clustering
2	1.2	50	0.7490 (0.023)	0.6713 (0.024)	0.640 (0.021)	0.4515 (0.014)
		100	0.9698 (0.007)	0.9592 (0.011)	0.9603 (0.011)	0.6936 (0.023)
	1.5	50	0.6921 (0.023)	0.6327 (0.021)	0.6031 (0.021)	0.4596 (0.018)
		100	0.9568 (0.009)	0.9650 (0.007)	0.9430 (0.011)	0.6133 (0.014)
	1.8	50	0.7052 (0.022)	0.6315 (0.023)	0.5727 (0.020)	0.4174 (0.02)
		100	0.9803 (0.004)	0.9539 (0.013)	0.9362 (0.013)	0.6433 (0.012)
1	1.2	50	0.9490 (0.013)	0.9284 (0.014)	0.9037 (0.013)	0.6489 (0.021)
		100	0.9992 (0.001)	0.9992 (0.001)	0.9984 (0.001)	0.9918 (0.003)
	1.5	50	0.9330 (0.014)	0.9193 (0.014)	0.9127 (0.014)	0.6304 (0.018)
		100	1 (0)	1 (0)	0.9976 (0.001)	0.9926 (0.003)
	1.8	50	0.9288 (0.013)	0.9235 (0.014)	0.9103 (0.014)	0.6437 (0.015)
		100	0.9992 (0.001)	0.9992 (0.001)	0.9967 (0.002)	0.9375 (0.017)
0.5	1.2	$50^{\dagger}$	0.9961 (0.004)	1 (0)	1 (0)	0.8504 (0.027)
		100	1 (0)	1 (0)	1 (0)	0.9991 (0.001)
	1.5	$50^{\dagger}$	0.9847 (0.009)	1 (0)	1 (0)	0.8193 (0.0260)
		100	1 (0)	1 (0)	1 (0)	1 (0)
	1.8	$50^{\dagger}$	0.9879 (0.007)	1 (0)	0.9973 (0.002)	0.7947 (0.026)
		100	1 (0)	1 (0)	1 (0)	1 (0)

Table 4.2: Summary of NMI in scenario 2, $(\beta_0^{kk}, \beta_0^{kl}) = (0.5, -0.5)$, over 50 runs. A superscript "$\dagger$" denotes a case in which (restricted Tweedie SBM with random initialization) $<$ (Poisson SBM) $\leq$ (restricted Tweedie SBM with Poisson initialization) in their respective clustering performances.

			Restricted Tweedie SBM		Poisson	Spectral
ϕ	ρ	n	Random Init.	Poisson Init.	SBM	Clustering
2	1.2	50	0.4385 (0.032)	0.4340 (0.027)	0.4243 (0.025)	0.2889 (0.022)
		100	0.8497 (0.013)	0.8025 (0.019)	0.774 (0.020)	0.5134 (0.016)
	1.5	50	0.5611 (0.023)	0.5226 (0.023)	0.5071 (0.022)	0.3462 (0.018)
		100	0.9097 (0.012)	0.8606 (0.016)	0.8146 (0.017)	0.5737 (0.012)
	1.8	50	0.6179 (0.022)	0.5771 (0.024)	0.522 (0.021)	0.4102 (0.018)
		100	0.9567 (0.009)	0.8736 (0.02)	0.8377 (0.020)	0.5985 (0.013)
1	1.2	50	0.8710 (0.016)	0.7404 (0.017)	0.7325 (0.016)	0.5379 (0.011)
		100	0.9893 (0.006)	0.9967 (0.002)	0.9842 (0.003)	0.862 (0.022)
	1.5	50	0.8709 (0.016)	0.7763 (0.017)	0.7684 (0.016)	0.5601 (0.012)
		100	0.9950 (0.004)	0.9992 (0.001)	0.9876 (0.003)	0.8311 (0.022)
	1.8	50	0.8806 (0.017)	0.8039 (0.019)	0.7901 (0.018)	0.6092 (0.013)
		100	0.9992 (0.001)	0.9992 (0.001)	0.9934 (0.002)	0.8876 (0.022)
0.5	1.2	50	0.9414 (0.014)	0.8998 (0.017)	0.8817 (0.016)	0.7379 (0.028)
		$100^{\dagger}$	0.9956 (0.004)	1 (0)	1 (0)	0.9983 (0.001)
	1.5	50	0.9591 (0.012)	0.9112 (0.015)	0.8999 (0.015)	0.7354 (0.026)
		100	1 (0)	1 (0)	1 (0)	1 (0)
	1.8	50	1 (0)	0.9727 (0.01)	0.9550 (0.01)	0.7549 (0.022)
		100	1 (0)	1 (0)	1 (0)	1 (0)

Table 4.3: Summary of NMI in scenario 3, $(\beta_0^{kk}, \beta_0^{kl}) = (0, -1)$, over 50 simulation runs. A superscript "$\dagger$" denotes a case in which (restricted Tweedie SBM with random initialization) $<$ (Poisson SBM) $\leq$ (restricted Tweedie SBM with Poisson initialization) in their respective clustering performances.

			Weak Effect ($\beta = 1$)			Strong Effect ($\beta = 2$)		
ϕ	ρ	n	NMI (excl. x_{ij})	NMI (incl. x_{ij})	$\hat{\beta}$	NMI (excl. x_{ij})	NMI (incl. x_{ij})	$\hat{\beta}$
0.5	1.2	50	0.9804 (0.009)	0.9794 (0.01)	1.0015 (0.006)	0.9289 (0.015)	1 (0)	2.0067 (0.006)
		100	1 (0)	1 (0)	0.9979 (0.002)	0.9976 (0.001)	1 (0)	1.9986 (0.002)
	1.5	50	0.9626 (0.013)	0.9908 (0.006)	1.0128 (0.005)	0.9017 (0.017)	0.9986 (0.001)	1.9969 (0.005)
		100	1 (0)	1 (0)	0.9994 (0.003)	0.9742 (0.009)	1 (0)	1.9995 (0.002)
	1.8	50	0.9667 (0.013)	0.9793 (0.009)	1.0083 (0.005)	0.8300 (0.023)	0.9883 (0.007)	2.0013 (0.006)
		100	1 (0)	1 (0)	0.9943 (0.003)	0.9731 (0.007)	1 (0)	1.9940 (0.003)
1	1.2	50	0.9234 (0.015)	0.9344 (0.016)	0.9948 (0.008)	0.8335 (0.022)	0.9846 (0.008)	1.9889 (0.007)
		100	0.9984 (0.001)	1 (0)	1.0026 (0.004)	0.9597 (0.009)	1 (0)	1.9974 (0.003)
	1.5	50	0.8811 (0.019)	0.9304 (0.015)	1.0039 (0.006)	0.7092 (0.022)	0.9687 (0.011)	1.9936 (0.007)
		100	0.9930 (0.003)	0.9992 (0.001)	0.9984 (0.004)	0.9225 (0.011)	1 (0)	1.9948 (0.004)
	1.8	50	0.8861 (0.016)	0.9404 (0.012)	1.0176 (0.007)	0.5655 (0.027)	0.9234 (0.015)	2.0155 (0.008)
		100	0.9877 (0.007)	0.9922 (0.005)	0.9946 (0.004)	0.8724 (0.013)	0.9945 (0.004)	1.9955 (0.003)
2	1.2	50	0.7058 (0.018)	0.7699 (0.018)	0.9994 (0.011)	0.6262 (0.022)	0.8621 (0.018)	2.0066 (0.009)
		100	0.9542 (0.009)	0.976 (0.008)	0.9960 (0.005)	0.9022 (0.011)	0.9887 (0.004)	1.9902 (0.004)
	1.5	50	0.6203 (0.023)	0.7028 (0.021)	1.0070 (0.012)	0.4602 (0.022)	0.7610 (0.019)	2.0025 (0.012)
		100	0.9015 (0.012)	0.9609 (0.009)	0.9868 (0.005)	0.7827 (0.015)	0.9735 (0.008)	1.9885 (0.006)
	1.8	50	0.5353 (0.025)	0.7114 (0.024)	1.0068 (0.012)	0.2581 (0.028)	0.7250 (0.022)	2.0236 (0.011)
		100	0.8477 (0.016)	0.9562 (0.01)	0.9892 (0.005)	0.6376 (0.018)	0.9403 (0.013)	1.9910 (0.006)

Table 4.4: Summary of clustering and estimation performance from the static model with covariates over 50 simulation runs, with $(\beta_0^{kk}, \beta_0^{kl}) = (0.5, -0.5)$.

$(\beta_0^{kk}, \beta_0^{kl})$		$\beta(t)$					
		$2t-1$	$\sin(2\pi t)$	$2t$	$\sin(2\pi t)$ $+1$	$0.5(2t-1)$	$0.5\sin(2\pi t)$
Scenario 1	NMI	1 (0)	1 (0)	1 (0)	1 (0)	0.996 (0.004)	1 (0)
$(1, 0)$	$\mathrm{Err}(\hat{\beta}(t))$	0.004 (0)	0.026 (0)	0.004 (0)	0.025 (0)	0.004 (0)	0.013 (0)
Scenario 2	NMI	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
$(0.5, -0.5)$	$\mathrm{Err}(\hat{\beta}(t))$	0.005 (0)	0.031 (0)	0.005 (0)	0.029 (0)	0.005 (0)	0.016 (0)
Scenario 3	NMI	1 (0)	1 (0)	1 (0)	0.996 (0.004)	1 (0)	1 (0)
$(0, -1)$	$\mathrm{Err}(\hat{\beta}(t))$	0.005 (0)	0.037 (0)	0.005 (0)	0.035 (0)	0.006 (0)	0.019 (0)
Scenario 4	NMI	0.996 (0.004)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
$(0.5, 0)$	$\mathrm{Err}(\hat{\beta}(t))$	0.004 (0)	0.029 (0)	0.005 (0)	0.027 (0)	0.005 (0)	0.015 (0)
Scenario 5	NMI	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
$(0.25, -0.25)$	$\mathrm{Err}(\hat{\beta}(t))$	0.005 (0)	0.031 (0)	0.005 (0)	0.03 (0)	0.005 (0)	0.016 (0)
Scenario 6	NMI	1 (0)	0.996 (0.004)	1 (0)	0.996 (0.004)	1 (0)	0.996 (0.004)
$(0, -0.5)$	$\mathrm{Err}(\hat{\beta}(t))$	0.005 (0)	0.034 (0)	0.005 (0)	0.033 (0)	0.005 (0)	0.017 (0)

Table 4.5: Summary of clustering and estimation performance (using $\lambda = 0.5$) from the time-varying model over 50 simulation runs, with $\phi = 1$, $\rho = 1.5$ and $n = 50$.

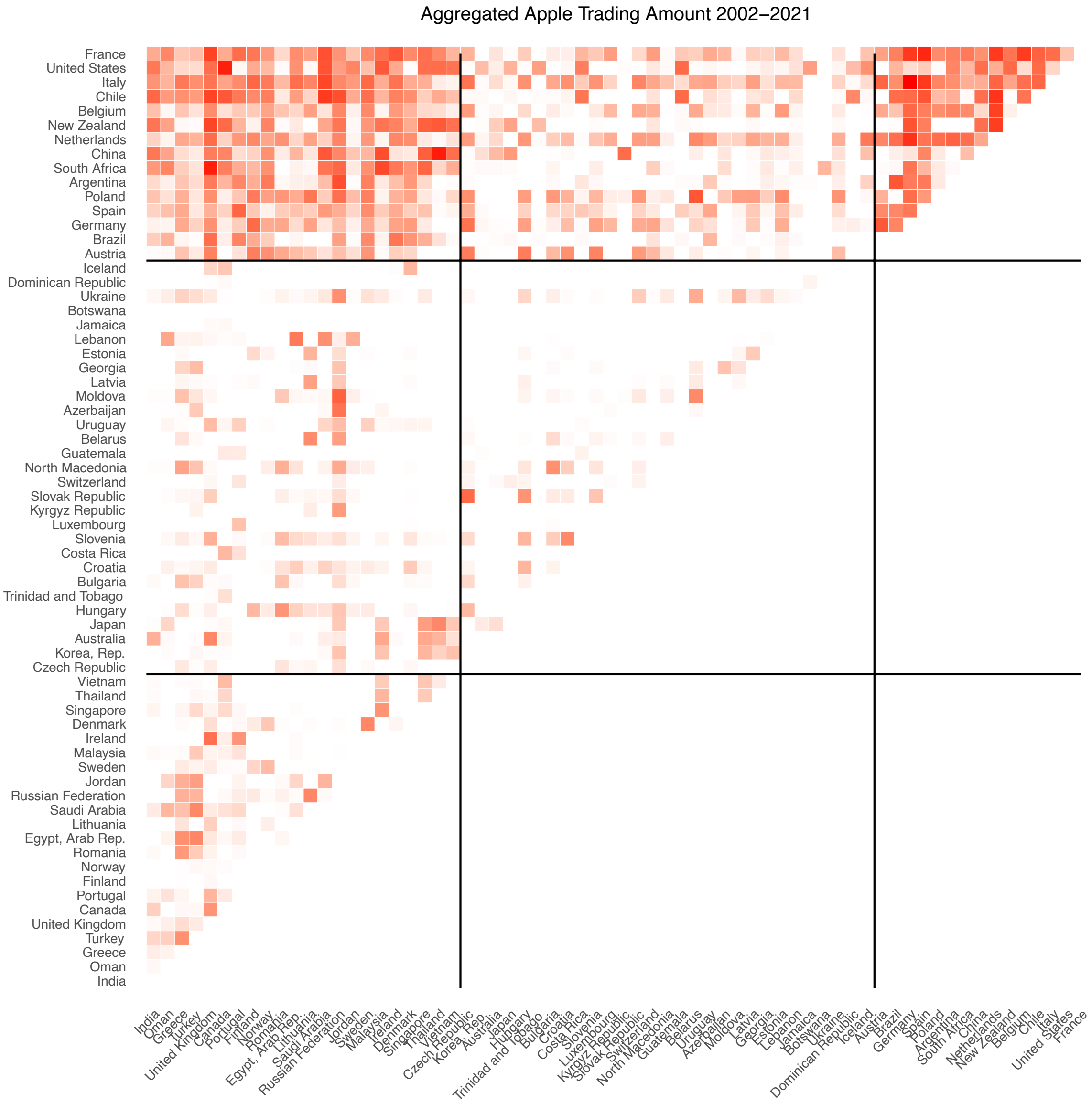


Figure 4.4: The (aggregated) matrix, $Y(2002) + \cdots + Y(2021)$, with rows and columns having been permuted according to the inferred community labels. Due to symmetry, only the upper half of the matrix is shown, with color shadings being proportional to each entry's respective magnitude.

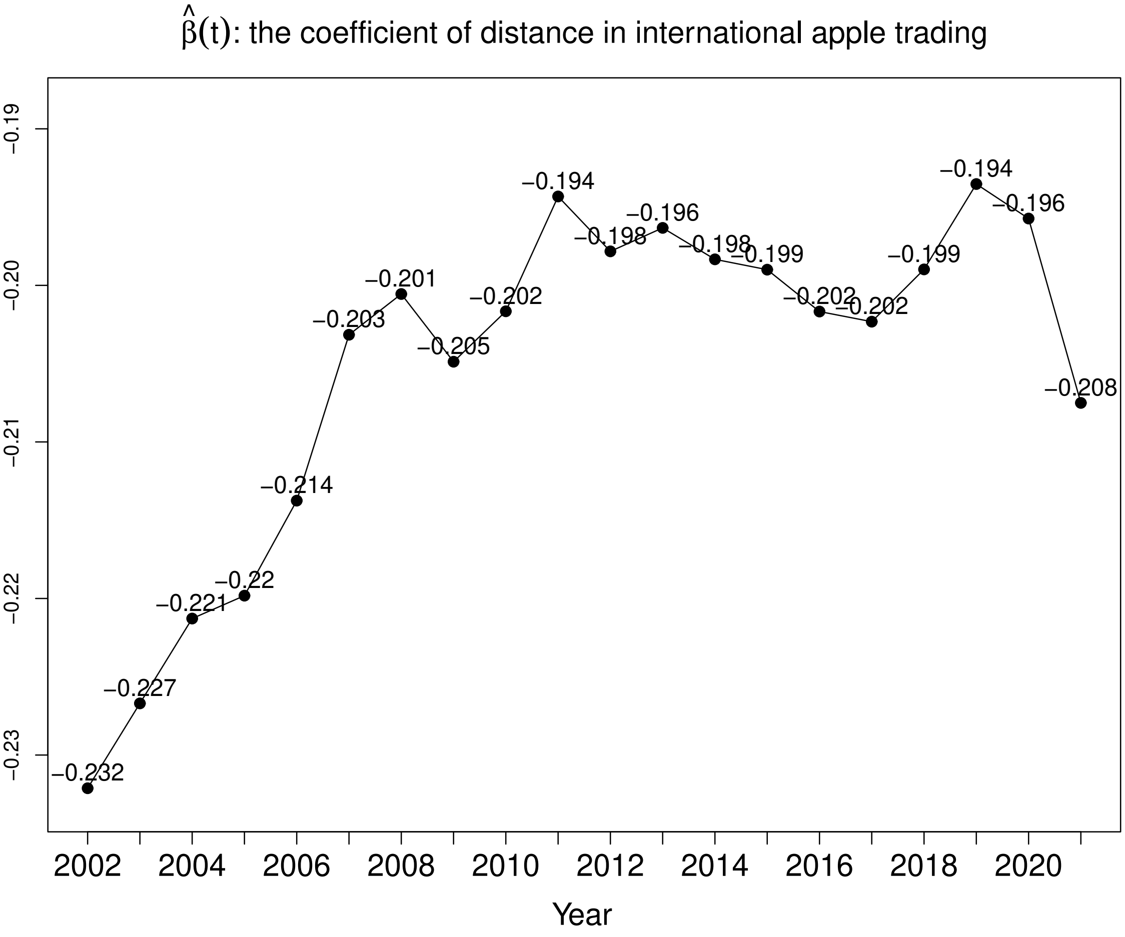


Figure 4.5: Estimated covariate coefficient $\hat{\beta}(t)$ for $\lambda^* = 0.1$.

Community	Country
1	France, United States, Italy, Chile, Belgium, New Zealand, Netherlands, China, South Africa, Argentina, Poland, Spain, Germany, Brazil, Austria
2	Iceland, Dominican Republic, Ukraine, Botswana, Jamaica, Lebanon, Estonia, Georgia, Latvia, Moldova, Azerbaijan, Uruguay, Belarus, Guatemala, North, Macedonia, Switzerland, Slovak, Republic, Kyrgyz Republic, Luxembourg, Slovenia, Costa Rica, Croatia, Bulgaria, Trinidad and Tobago, Hungary, Japan, Australia, Korea Rep, Czech Republic
3	Vietnam, Thailand, Singapore, Denmark, Ireland, Malaysia, Sweden, Jordan, Russian, Federation, Saudi, Arabia, Lithuania, Egypt Arab Rep., Romania, Norway, Finland, Portugal, Canada, United Kingdom, Turkey, Greece, Oman, India

Table 4.6: Community detection results of 66 countries in international apple trading.

Chapter 5

Cox Proportional Hazard Model in Community Detection

5.1 Introduction

5.1.1 Background

In political science, we seek to understand the duration of existing commitments between nations, the longevity of wars, and the sustainability of peace between rivals. Such event history data, where the event of interest is typically the occurrence of a specific event or endpoint over time, is frequently analyzed using survival models, as exemplified by Box-Steffensmeier and Jones (1997). Several reasons make survival models popular in studying these problems. First, censored observations are common in political science data. For instance, the exact time of the violation of a peace agreement is not known but only that it has not occurred up to a certain point in time, which can be considered as the right censoring. In addition, the study of time-to-event data in political science often involves time-dependent covariates. In a notable study by Leeds et al. (2009), the primary objective is to assess the probability of an existing military alliance ending in violation of its terms at a specific time. This analysis considers the influence of variables that evolve over time, such as changes in the leader's societal coalition and leadership transitions. Moreover, political events can exhibit varying effects over time. Consequently, the instantaneous risk of an event occurring at a given time, conditioned on survival up to that point, becomes a topic of significant interest in political science. Understanding the dynamic nature of this risk helps comprehend the temporal dynamics of political processes and policy changes.

The Cox proportional hazards model is known for its capability to model the risk of an event occurring as the product of a baseline hazard function and the exponential term representing the effect of covariates, treated separately. The hazard function is modeled for a subject i with covariate vector x_i as

$$\lambda(t|X_i = x_i) = \lambda_0(t)\exp(x_i^\top \beta), \tag{5.1}$$

where β is a vector of unknown regression parameter, and $\lambda_0(t)$ is the baseline hazard function which represents the hazard rate in the case when the covariate is zero. In the standard Cox proportional hazard model (5.1), there is no intercept term within the exponential component, since a constant will be absorbed into $\lambda_0(t)$. The name "proportional hazard" comes from the fact that the ratio of the hazard rates of two subjects only depends on the discrepancy of their covariates

$$\frac{\lambda(t|X_i = x_i)}{\lambda(t|X_j = x_j)} = \exp\left((x_i^\top - x_j^\top) \cdot \beta\right). \tag{5.2}$$

The Cox proportional hazards model is increasingly utilized in various fields of political science, particularly in the study of international relationships. Examples include the war termination (Weisiger, 2013), the duration of peace after civil conflict (Loyle and Appel, 2017), and the cease-fire duration (Fortna, 2018).

However, there remains a notable gap in accounting for the network structures and latent community dynamics within these interaction times. As the events mentioned above primarily involve relations between different nations, typically characterizing interactions or ties between two political entities, overlooking such network structures may introduce deficiencies in the models. The biggest issue is the violation of the most significant assumption in the Cox proportional hazard model—that the hazard ratio between any two individuals is constant over time given covariates. In the context of international relations, the timing of one interaction on the network may affect the timing of other interactions. For instance, when a major power establishes diplomatic relations with a country, it can prompt other countries to follow suit. For example, when the United States recognized the People's Republic of China in the 1970s, it led to a wave of other countries also establishing diplomatic relations with China, while simultaneously severing ties with Taiwan. This violates the assumption of the Cox proportional hazard model that events are independent given the covariates.

Therefore, it is important to incorporate community structures into the Cox proportional hazards model when analyzing international relations. Countries exhibit a similar propensity to establish diplomatic relations driven by shared political ideologies, cultural affinities, or common economic interests. We integrate the community structures into the parametric component of the Cox proportional model. It becomes more reasonable to assume the independence of the timing of interactions given covariates along with the communities of countries. This adjustment can reduce the risk of model misspecification by capturing more of the variability in the international relations data.

In this chapter, we aim to extend the Cox proportional hazard framework to incorporate the latent community memberships of the nodes.

5.1.2 Existing Work

Understanding the temporal dynamics of interactions between nodes helps gain insights into the formation of networks. Various studies have employed counting processes to model the time of events in dynamic networks. Perry and Wolfe (2013) examined the impact of node characteristics and behaviors of the nodes on the time of directed interactions, considering homophily in network dynamics. They utilized the Cox model, assuming conditional independence of directed interactions given the covariate history, with baseline rates influenced by observable traits like ethnicity or gender. Sit et al. (2021) also delved into dynamic directed networks, employing multivariate counting processes. In contrast to Perry and Wolfe (2013), they assumed that the included covariates could capture all dependencies between future and past events, shaping the intensity function of interactions over time. This nuanced approach provides insights into the evolving nature of directed interactions in networks.

The incorporation of community structures into the modeling framework has been a focus of research, as seen in the work of Matias et al. (2018). They assumed that individuals belong to latent groups, and conditional interactions between two individuals follow an inhomogeneous Poisson process. The cumulative intensity function is driven by the latent groups of interacting nodes, contributing to a more comprehensive understanding of network dynamics. Taking a different approach, He et al. (2023) modeled community labels in a static latent space component, incorporating a time-varying node-specific baseline component. This model was designed to capture the distribution of the number of undirected interactions at a given time, considering both node-specific features and latent space components.

Simultaneously modeling the distribution and timing of interactions, Song et al. (2023) integrated a mixed membership SBM with a cure rate model. This approach allows for the consideration of community memberships of nodes alongside the temporal dynamics of events, offering a more holistic perspective on network interactions.

In a parallel line of research, Cai et al. (2022) propose a new class of nonstationary Hawkes processes to learn latent network structures from large-scale multivariate point process data, allowing for excitatory and inhibitory effects. They establish non-asymptotic error bounds, facilitating parameter estimation and testing temporal constancy of background intensity. Fang et al. (2024) explored the temporal dynamics of interactions using point processes, considering both traditional Poisson and Hawkes processes. Their work accounted for bursty occurrences in interactions and proposed a fast online variational inference algorithm for estimating latent structures underlying dynamic event arrivals.

These studies collectively contribute to the evolving field of modeling time-dependent interactions on networks, providing valuable insights into the complex interplay of community structures, covariates, and latent variables in political science research. Additionally, the extensions by Larsen (2005) and Pei et al. (2024) in using latent variables in Cox proportional hazard models further enrich the literature on this topic.

5.1.3 Outline

The rest of the chapter is organized as follows. Section 5.2 initially introduces the vanilla Cox model incorporating community structures, followed by the introduction of a model that additionally considers covariate effects. Section 5.3 outlines a modified EM algorithm utilized for model estimation, while Section 5.4 presents the results of simulation studies. Lastly, Section 5.5 applies the proposed vanilla model to analyze the time-to-event network depicting diplomatic relations among nations.

5.2 Methodology

In this section, we introduce a Cox proportional hazard model tailored for heterogeneous time-to-event network data. In Section 5.2.1, we start by presenting a vanilla model, where the formation and duration of the network connection are influenced by the community label of the node. Building upon this foundation, in Section 5.2.2, we delve into a more comprehensive model that accommodates the impact of time-invariant covariates on the network dynamics.

5.2.1 Vanilla Model

In network modeling, the event of interest is the connection between nodes. We consider pairwise interaction and censoring time among n nodes. Specifically, let T_{ij} denote the time when node i and node j interact and C_{ij} the censoring time of the pair respectively. For each dyad between nodes i and j, we observe (Y_{ij}, δ_{ij}), where $Y_{ij} = \min\{T_{ij}, C_{ij}\}$, and $\delta_{ij} = I(T_{ij} \leq C_{ij})$ is the censoring indicator to characterize the whether the pair $i - j$ experiences the event of interest during the observational period. We assume that each node belongs to one of K communities, and the nodes in the same community have similar patterns in terms of interaction times. We denote the community label of node i as c_i, and $c_i \in \{1, \cdots, K\}$ follows a multinomial distribution with the probability $(\pi_1, \cdots, \pi_K)$. Our interest lies in the community effects on the distribution of survival times. Following the classic Cox proportional hazards model (Cox, 1972), we consider the following model with community-specific effects:

$$\lambda(t_{ij}|\alpha, c_i = k, c_j = l) = \lambda_0(t_{ij}) \exp(\alpha_{kl}), \;\; k, l \in \{1, \cdots, K\} \tag{5.3}$$

where $\lambda_0(\cdot)$ is the unspecified baseline hazard function, and $\alpha \in \mathbb{R}^{K \times K}$ is a symmetric matrix. We are interested in estimating the community label of each node.

5.2.2 Model with Covariates

In numerous real-world scenarios, it is useful to recognize the influential roles that both community labels and covariates play in shaping the distribution of interaction times. Beyond the impact of alliances or communities, the geographical proximity also emerges as a pivotal factor influencing the timing of diplomatic ties between nations. Considering the effect of edge-wise covariates enhances the ability of the model to navigate and comprehend more complex social phenomena. For this purpose, we assume that both the community labels and the covariates have impacts on the interaction times. Suppose that in addition to the survival time of each edge Y_{ij}, a collection of p symmetric covariate matrices $X^{(1)}, ..., X^{(p)} \in \mathbb{R}^{n \times n}$ may also be available, where the (i, j)-th entry $x_{ij}^{(u)}$ of each $X^{(u)}$ represents a pair-wise covariate containing some information about the connection between node i and node j, and $x_{ii}^{(u)} = 0$ for all $1 \leq i \leq n$ and $u = 1, ..., p$. We can extend the vanilla model in Section 5.2.1 to formulate

$$\lambda(t_{ij}|c_i = k, c_j = l, \alpha, \beta, X_{ij} = x_{ij}) = \lambda_0(t_{ij}) \exp(\alpha_{kl} + x_{ij}^\top \beta), \;\; k, l \in \{1, \cdots, K\}. \tag{5.4}$$

Given a data set $D = \{Y, \delta, X^{(1)}, ..., X^{(p)}\}$, we focus exclusively on the parametric part, omitting the non-parametric terms present in (5.4). Adapting the partial likelihood function from the classic Cox proportional hazards model in (Cox, 1972) to our context, which incorporates the joint probability of community assignments for all nodes in the network, yields the following partial likelihood function of the unknown parameter vector $\theta = (\alpha, \beta, \pi)$

$$\begin{aligned}
L(\theta; D, c) &= \prod_{i=1}^{n} \pi_{c_i} \prod_{(i,j)\in E} \frac{\exp(\alpha_{c_i c_j} + x_{ij}^{\top}\beta)}{\sum\limits_{(r,s)\in R(y_{ij})} \exp(\alpha_{c_r c_s} + x_{rs}^{\top}\beta)} \\
&= \prod_{i=1}^{n}\prod_{k=1}^{K} \pi_k^{\mathbb{1}(c_i=k)} \prod_{(i,j)\in E}\prod_{k,l=1}^{K} \exp(\alpha_{kl} + x_{ij}^{\top}\beta)^{\mathbb{1}(c_i=k,c_j=l)} \\
&\times \prod_{(i,j)\in E}\prod_{k_r,k_s,\cdots=1}^{K} [\sum_{(r,s)\in R(y_{ij})} \exp(\alpha_{k_r k_s} + x_{rs}^{\top}\beta)]^{-\mathbb{1}(c_r=k_r,c_s=k_s,\cdots)} \qquad (5.5)
\end{aligned}$$

where the uncensored set $E = \{(i,j)|\delta_{ij} = 1\}$ and the risk set at the time y, $R(y) = \{(i,j)|y_{ij} \geq y\}$, represents the interaction set experiencing the event of interest and the interaction set with survival time greater or equal to y respectively. The likelihood function of the vanilla model in Section 5.2.1 is a special case of (5.5).

5.3 Computation

In this section, we develop an EM algorithm to estimate π, α, β and infer the latent community labels c for the models in Section (5.2), when the number of communities K is fixed. In the E step, we compute the posterior probability of the community labels and obtain the corresponding expected values via Gibbs sampling. In the M step, we estimate other parameters of interest.

We work on the following log-likelihood function

$$\begin{aligned}
\ell(\theta; D, c) &= \sum_{i=1}^{n}\sum_{k=1}^{K} \mathbb{1}(c_i = k)\pi_k + \sum_{(i,j)\in E}\sum_{k,l=1}^{K} \mathbb{1}(c_i = k, c_j = l)(\alpha_{kl} + x_{ij}^{\top}\beta) \\
&- \sum_{(i,j)\in E}\sum_{k_r,k_s,\cdots=1}^{K} \mathbb{1}(c_r = k_r, c_s = k_s, \cdots)\log[\sum_{(r,s)\in R(y_{ij})} \exp(\alpha_{k_r k_s} + x_{rs}^{\top}\beta)]. \qquad (5.6)
\end{aligned}$$

5.3.1 Expectation Step

The E step computes the expectation of the log-likelihood(5.6) given the observed data $D = \{Y, \delta, X^{(1)}, ..., X^{(p)}\}$ and the current estimated parameters from the previous h^{th} step, $\theta^{(h)} = (\pi^{(h)}, \alpha^{(h)}, \beta^{(h)})$. The resulting conditional expectations of the log-likelihood take the form

$$\begin{aligned}\mathbb{E}[\ell(\theta; D, c)|D, \theta^{(h)}] &= \sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}[\mathbb{1}(c_i = k)|D, \theta^{(h)}]\pi_k \\ &\quad + \sum_{(i,j)\in E}\sum_{k,l=1}^{K}\mathbb{E}[\mathbb{1}(c_i = k, c_j = l)|D, \theta^{(h)}](\alpha_{kl} + x_{ij}^{\top}\beta) \\ - \sum_{(i,j)\in E}\sum_{k_r,k_s,\cdots=1}^{K}&\mathbb{E}[\mathbb{1}(c_r = k_r, c_s = k_s, \cdots)|D, \theta^{(h)}]\log[\sum_{(r,s)\in R(y_{ij})}\exp(\alpha_{k_r k_s} + x_{rs}^{\top}\beta)].\end{aligned}\tag{5.7}$$

We need to calculate three expected values for certain quantities: $\mathbb{E}[\mathbb{1}(c_i = k)|D, \theta^{(h)}]$, $\mathbb{E}[\mathbb{1}(c_i = k, c_j = l)|D, \theta^{(h)}]$, and $\mathbb{E}[\mathbb{1}(c_r = k_r, c_s = k_s, \cdots)|D, \theta^{(h)}]$. The first two are commonly employed in the classic SBM. However, the last term, which represents the expected value of indicators for nodes involved in interactions occurring after a certain moment, is unique to our specific problem setting, introducing additional complexity to the problem.

The presence of the third term reduces the effectiveness of the variational EM method and underscores our choice of EM algorithm. In the variational E step of the variational EM algorithm, after approximating $\mathbb{P}(c_i|D; \theta)$ with the variational distribution $c_i \in MUL(\tau_{i1}, \cdots, \tau_{iK})$, at a specific moment y where there are N_y nodes engaged in interactions occurring after y, the third term requires the computation of various combinations of community labels for these N_y nodes,

$$\sum_{k_r,k_s,\cdots=1}^{K}\tau_{r,k_r}\tau_{s,k_s}\cdots\log[\sum_{(r,s)\in R(y_{ij})}\exp(\alpha_{k_r k_s} + x_{rs}^{\top}\beta)]$$

resulting in a total of N_y^K terms to be calculated. As a result, we opt for Gibbs sampling to efficiently compute the expected values of the entire third term that will be used in the subsequent M step as

$$\mathbb{E}\Bigg[\log[\sum_{(r,s)\in R(y_{ij})}\exp(\alpha_{c_rc_s}+x_{rs}^{\top}\beta)]\Bigg],$$

where the number of terms to be calculated is reduced to the number of Gibbs iterations we implemented.

Let c_{-i} denote the community labels of all nodes except for i. The Gibbs sampling samples the community label of each node in order via the following distribution

$$\begin{aligned} c_1 &\sim \mathbb{P}(c_1|\theta^{(h)}, c_{-1}; D),\\ c_2 &\sim \mathbb{P}(c_2|\theta^{(h)}, c_{-2}; D),\\ &\vdots\\ c_n &\sim \mathbb{P}(c_n|\theta^{(h)}, c_{-n}; D). \end{aligned}$$

The above sampling distribution of node i is multinomial, where the probability of it belonging to each community k conditional on the current network structure and community assignments of other nodes is specified by

$$\mathbb{P}(c_i = k|\theta^{(h)}, c_{-i}; D) = \frac{L(\theta^{(h)}; D, c = (c_1, \cdots, c_{i-1}, k, c_{i+1}, \cdots, c_n))}{\sum\limits_{l=1}^{K} L(\theta^{(h)}; D, c = (c_1, \cdots, c_{i-1}, l, c_{i+1}, \cdots, c_n))}.$$

We repeat the above sequential sampling process for a predefined number of iterations. Samples are then collected from the final specified iterations to calculate the aforementioned three quantities—$\mathbb{E}[\mathbb{1}(c_i = k)|D, \theta^{(h)}]$, $\mathbb{E}[\mathbb{1}(c_i = k, c_j = l)|D, \theta^{(h)}]$, and $\mathbb{E}\Bigg[\log[\sum\limits_{(r,s)\in R(y_{ij})}\exp(\alpha_{c_rc_s}+ x_{rs}^{\top}\beta)]\Bigg]$.

5.3.2 Maximization Step

In the M step, we update θ by maximizing (5.6). Specifically, we update π subject to the constraint $\sum_{k=1}^{K} \pi_k = 1$, resulting in the closed form

$$\pi_k^{(h+1)} = \frac{\sum\limits_{i=1}^{n} \mathbb{E}[\mathbb{1}(c_i = k)|D, \theta^{(h)}]}{n}, \; k = 1, \cdots, K.$$

The values of α and β are updated directly by maximizing (5.6) using the R function `optim`.

Remark 2. *In (5.4), α_{kl} can be considered as a part of the baseline hazard function for the community $k - l$. α_{kl}'s are not individually identifiable, but their ratios are meaningful. The hazard ratio, denoted as $\exp(\alpha_{kl} - \alpha_{k'l'})$, captures the relative risk of an event occurring in one community compared to the other. To ensure the identifiability in (5.4), we arbitrarily choose one α_{kl} to set it as a reference level for the baseline hazard. Without loss of generality, we assume that $\alpha_{11} = 0$ as the reference level, reducing the model to estimate only other $K(K-1)/2 - 1$ α_{kl}'s.*

5.4 Simulation

In this section, we conduct simulation studies to assess the finite-sample performance of the proposed vanilla model (5.3) and the model with covariates (5.4). In both models, we will assess clustering performance using the NMI. Additionally, for the model with covariates, our emphasis will extend to evaluating the accuracy of estimated covariate effects.

In the simulation to validate our models, we generate the survival time from an exponential distribution. Specifically, the survival time of T_{ij} is simulated as

$$T_{ij} = -\frac{\log u}{\exp(\alpha_{c_i c_j} + x_{ij}^\top \beta)}, \text{ where } u \sim \text{UNIF}(0, 1).$$

For all simulations, we fix the number of communities to be $K = 2$, with prior probabilities $\pi_1 = 0.4$ and $\pi_2 = 0.6$. We consider a symmetric true block matrix α, in which all diagonal entries are identical, and likewise, all off-diagonal entries are identical. In the simulated data, we censor 5% of the top generated survival time.

At the current stage, we mainly focus on verifying the computational methods at this particular rate through the simulation studies. Future work will explore varying censoring rates, such as 3% or 10%, to assess their impact on estimation performance.

We initiate the algorithm with the accurate number of communities; consequently, the resulting community labels may merge, but exceeding the correct number of community labels is not possible. In the EM algorithm, to mitigate the risk of being trapped in suboptimal local optima, we employ multiple initial labels in each run, ultimately selecting the result with the largest log-likelihood.

5.4.1 Simulation of Vanilla Model

In this section, we conduct three distinct simulation experiments using our proposed vanilla model 5.3, each with a specific true value of α. The number of initial labels we start in each run is 5. We summarize in Table 5.1 the averages and the standard errors of the NMI metric over 30 simulation runs. Table 5.1 shows that the EM algorithm accurately infers the community labels under different combinations of sample size and the true values of α. The clustering performance also improves when the sample size increases from 20 to 50.

$(\alpha_{kk}, \alpha_{kl})$	n	NMI
$(0.5, -0.5)$	20	0.7413 (0.058)
	50	0.9626 (0.033)
$(0.75, -0.75)$	20	0.8311 (0.063)
	50	1 (0)
$(1, -1)$	20	0.9406 (0.041)
	50	1 (0)

Table 5.1: Simulation results from vanilla model.

5.4.2 Simulation of Model with Covariates

In this section, we aim to examine the performance of our proposed model with covariates in Section 5.4. We consider $p = 1$, and x_{ij} are generated independently from UNIF$(-1, 1)$. The number of initial community labels in each run is 30. In addition to the NMI, we summarize the mean and standard error of the estimate $\hat{\beta}$ over repeated 30 simulation runs in Table 5.2. The performance of both clustering and the estimation of $\hat{\beta}$ improves with an increase in sample size, as demonstrated in Table 5.2.

$(\alpha_{kk}, \alpha_{kl})$	β	n	NMI	err(β)
$(0.75, -0.75)$	1	20	0.9756 (0.014)	0.0139 (0.004)
		50	1 (0)	8×10^{-4} (0)
	2	20	0.9837 (0.011)	0.029 (0.008)
		50	0.9933 (0.046)	0.013 (0.008)
$(1, -1)$	1	20	0.9371 (0.044)	0.0194 (0.006)
		50	1 (0)	8×10^{-4} (0)
	2	20	0.9917 (0.008)	0.0309 (0.009)
		50	1 (0)	0.0021 (0.001)

Table 5.2: Simulation results from the model with covariates.

5.5 Application

In this section, we apply the proposed method to detect the communities among a set of countries based on their diplomatic relations and the time associated with them. We use the data from the *Correlates of War Diplomatic Exchange* data set (Bayer, 2006). The original data set captures diplomatic representation at the level of chargée d'affaires, minister, and ambassador between different countries. The dataset provides directed dyadic information on the level of diplomatic representation and diplomatic exchange between countries for the years 1817, 1824, 1827, 1832, 1836, and 1840, as well as every five years from 1844 to 1914, 1920 to 1940, and 1950 to 2005 (Bayer, 2006).

Our study focuses on a set of 30 selected countries. The survival time between a pair of countries denoted as i and j, is defined as the earliest year in which both diplomatic representation levels from i to j and from j to i reach either level 2 or level 3. Specifically, level 2 includes titles such as minister, minister plenipotentiary, minister resident, and envoy, while level 3 encompasses titles like ambassador, high commissioner, secretary of Libyan People's bureau, similar designations, high commissioner or ambassador resident elsewhere, and ambassador, high commissioner, or secretary vacant. If the earliest year meeting these criteria doesn't exist in the dataset for the country pair i and j, we regard it as right censoring.

Our models, (5.3) and (5.4), operate under the assumption that the survival time is continuous, implying the absence of ties. In the context of survival analysis, "ties" arise when two or more pairs undergo the event of interest simultaneously. This occurs because the dataset (Bayer, 2006) collects information at discrete time intervals. We break the ties by randomly jittering these non-unique times within the allowed range of 0.5 to introduce small random variations to the discrete data points.

While this jittering method is effective, its suitability may diminish in the presence of covariates. Therefore, in future work, we intend to explore additional techniques tailored to handle interval-censored data. These may include established methods such as the Breslow Method and Efron's Method, as well as extensions of the traditional Cox proportional hazards model to data with interval-censoring structure.

Figure 5.1 shows how these countries are clustered into three communities by our model. The block patterns are clear in the permuted adjacency matrix. The first community comprises primarily European nations, along with the United States and two South American countries, where diplomatic relations were established early among its members. The second cluster seems to be a diverse mix of countries from various regions, which suggests that they may have formed diplomatic ties during a period characterized by global shifts in political dynamics, economic changes, or emerging international organizations. The third community likely comprises countries that established diplomatic relations more recently compared to the other two communities.

After demonstrating the effectiveness of our first vanilla model in clustering countries according to when their strong diplomatic relations are built, future work will focus on exploring suitable covariates to facilitate the application of the method (5.4) in real-world scenarios. Future work also needs to include more countries, while also selecting them wisely. For instance, in our current network, we consider Czechoslovakia, Czech Republic, and Slovakia. However, Czechoslovakia peacefully split into the Czech Republic and Slovakia in 1993. In our future work, we will collaborate closely with researchers in political science to ensure the selection of countries aligns with the evolving geopolitical landscape.

Community	Country
1	United States, United Kingdom, France, Germany, Italy, Sweden, Denmark, Spain, Belgium, Netherlands, Argentina, Brazil
2	Japan, Finland, Norway, Thailand, Switzerland, Hungary, Greece, Chile, Mexico, Peru, Dominican Republic, Poland, Czechoslovakia, Romania, Bulgaria
3	Canada, Iceland, South Korea, India, Malaysia, Singapore, Australia, New Zealand, Austria, Ireland, Luxembourg, China, North Korea, Czech Republic, Slovakia, Croatia

Table 5.3: Community detection results of 43 countries in diplomatic relations.

5.6 Conclusion

In conclusion, this chapter presents a novel framework integrating the Cox proportional hazard model with time-to-event data on networks, incorporating community structures. Notably, our model also accommodates the influence of nodal and edge information. The application of our model to international relational data yields valuable insights into the formation of diplomatic relations, shedding light on the intricate dynamics within the community of nations. This framework offers a versatile approach for analyzing time-varying network data, contributing to the nuanced understanding of relational structures and their temporal evolution.

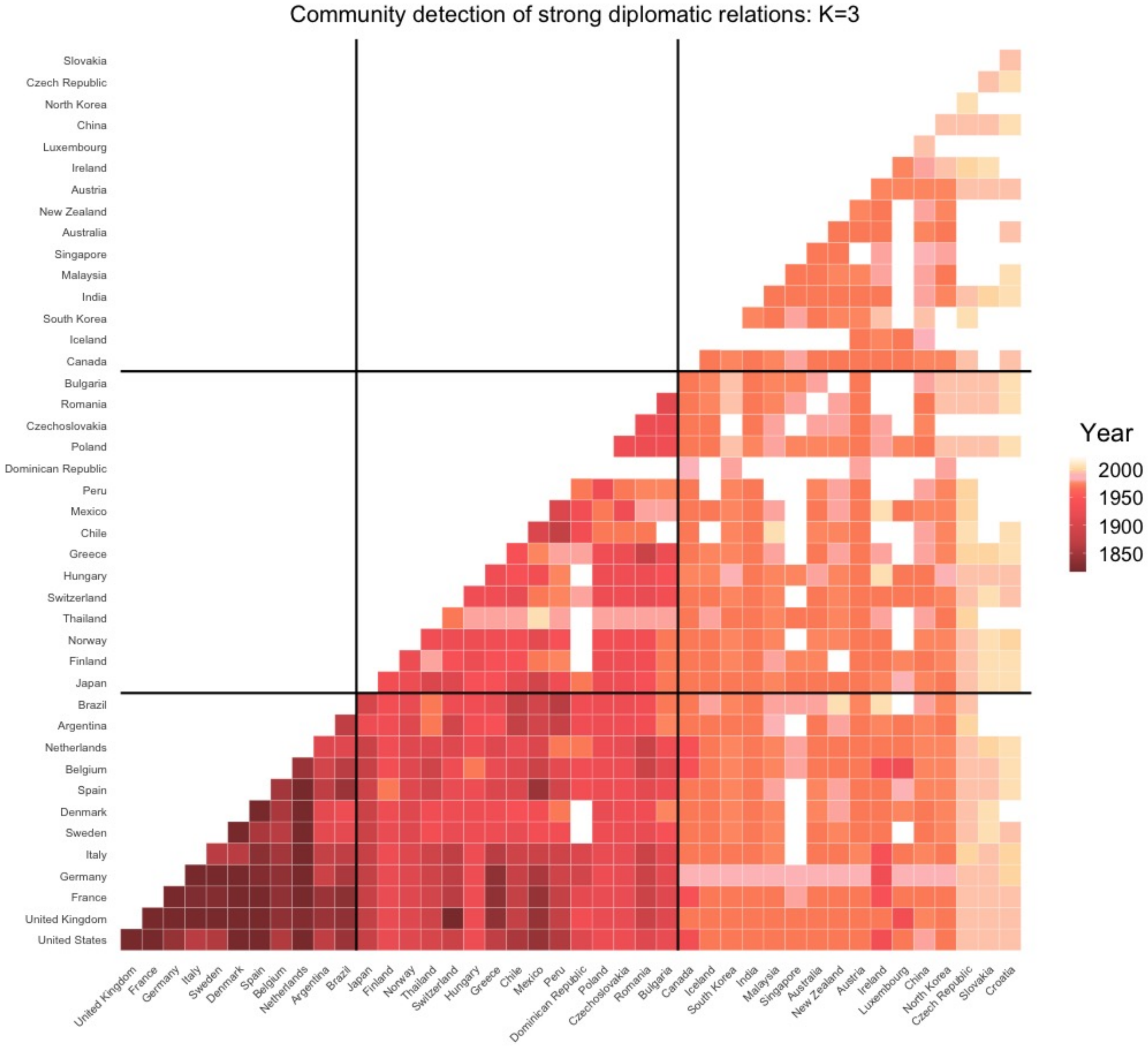


Figure 5.1: The survival time matrix with rows and columns having been permuted according to the inferred community labels. Due to symmetry, only the lower half of the matrix is shown, where color shades correspond to the survival time of each pair, and white cells denote censoring.

Chapter 6

Summary and Future Work

This thesis has explored the estimation of time-varying networks and the detection of communities within them across diverse network types. This chapter serves to consolidate our findings and discusses potential ways for extending and refining the methodologies developed in this thesis.

6.1 Summary

The rapid advancement in data collection and storage has led to the widespread availability of dynamic network data. However, conventional statistical methodologies primarily focus on static network structures, creating a significant gap between the evolving nature of network data and the methodologies used. This thesis investigates diverse types of time-varying network data and addresses various network-related challenges.

In Chapter 2, an overview of the two primary network problems studied in subsequent chapters is presented: firstly, the estimation of conditional dependency structures among a set of random variables using GGM, and secondly, the clustering of network nodes based on their connections through SBM. Additionally, this chapter introduces essential computational tools necessary for addressing these challenges.

Identifying the conditional dependence among a set of random variables is very important in many applications. This problem is often solved via GGM. Many approaches have been developed to model a static GGM, which can be primarily categorized into two types, namely, the l_1-penalized Gaussian likelihood method, also known as GLASSO, and the penalized regression method, referred to as SPACE. However, network structures often change over time, posing challenges for static GGMs. This has led to a demand for new methods to estimate dynamic networks, with most existing approaches building upon the GLASSO due to its computational convenience. Yet the regression framework SPACE can have some advantages: 1) It can be generalized to non-Gaussian data more easily; 2) It estimates the partial correlations directly which are more interpretable to characterize the conditional dependencies; 3) It is computationally more efficient when only a small portion of random variables' network structure is of interest. In Chapter 3, we propose time-varying GGMs, where statistical methods are developed to identify the associations and their dynamic changes in discrete time-varying networks based on SPACE. This goal is achieved under the assumption that the changes in the temporal network from one time point to the next are smooth, which encourages the regularization on the difference of partial correlations between adjacent time points. The penalty we impose on the smoothness of partial correlation networks includes both l_1 and l_2 regularization, leading to two different approaches namely GEN and GFL. To overcome the non-trivial computational challenges in the resulting optimization problems, we exploit the special structures of the symmetric block tri-diagonal matrix. These computational tricks greatly speed up the inversion of large scale matrices. With the computational tricks embedded, the ADMM provides a unifying framework for solving both the GEN and the GLF problems. Tuning parameter selection is an important research question in machine learning. In Chapter 3, two tuning parameters in both GEN and GFL control the sparsity and smoothness of the networks respectively, which makes the tuning more challenging. To adopt the BIC, we derived the degrees of freedom in GEN and GFL under the framework of Stein's unbiased risk estimation. We apply our methods to fMRI data of human brains in both healthy individuals and those suffering from the ADHD. The time-varying networks on 18 cerebellum regions of interest are constructed for healthy and ADHD groups respectively and the most markedly differences between the two groups are noted. We also estimate the dynamic climatic connectivity among some major Canadian cities to illuminate previously unobserved but crucial spatiotemporal patterns of climatic relevance between spatially dispersed grid points.

Given the observed or estimated connections among nodes, the task of clustering nodes within a network into distinct groups based on shared connection patterns is referred to as community detection. This network problem is fundamental in statistical machine learning, with wide applications including marketing and recommendation systems. One prominent method for community detection is a generative probabilistic graph model known as the SBM. When working with relational data in finance, such as detecting latent community structures among countries based on the dollar amounts of their international trading values, three main difficulties arise: 1) Existing SBMs are unable to accommodate networks with non-negative continuous edge weights including the possibility of zero counts; 2) The stochastic equivalence assumption in traditional SBMs makes incorporating nodal information hard; 3) Effectively capturing dynamic network data requires special attention and methodology. In Chapter 4, we introduce a novel SBM that addresses the first challenge aforementioned by utilizing the restricted Tweedie distribution, a compound Poisson-Gamma distribution, to model the distribution of each edge weight. Moreover, we extend our approach to tackle the second and third challenges by incorporating edge information as covariates into the restricted Tweedie parameter and allowing the covariates effects to change over time smoothly. Notably, we also prove an interesting result that, asymptotically, the covariate effects in our model can be estimated irrespective of how community labels are assigned to each node. This result allows us to use an efficient two-step algorithm, separating the estimation of the covariate effects and that of the other parameters—including the unknown community labels in the variational inference. The application of our models to an international trading network, focusing on the dynamic effects of geographic distance between countries, has provided valuable insights into the complexities of global economic relationships.

While Chapter 4 focuses on understanding community detection within time-varying networks observed at each temporal time point, Chapter 5 studies the timestamps of the observed interactions. In this context, the entries in the adjacency matrix represent the time when the interactions among nodes occur. Modeling the survival times of connections among nodes, such as the formation of diplomatic relations and the duration of wars, is an important problem in political science. The Cox proportional hazard model has been extensively employed in political science to analyze the event history of interactions among nations. However, the inherent but latent community structure among countries can violate the independence assumption of events given the covariates. We consider the community structures when modeling the survival time of the interactions in the Cox proportional hazard model. Our model assumes that the hazard ratio between two interactions depends on the community labels of the nodes involved and the effect of the covariate discrepancy, and it remains constant over time. The methods are applied to the community detection of the time when two countries establish a strong diplomatic relationship.

6.2 Future Work

In future work, we will explore several directions stemming from the research conducted in Chapters 3 to 5.

In Chapter 3, the adaptability and interpretability of regression-based approaches open avenues for further exploration. Our future trajectory involves relaxing the Gaussian assumption. Real-world multivariate data seldom adheres strictly to Gaussian distributions, often showcasing heavy-tailed behaviors with outliers. To address this generally, we can incorporate robust loss functions, such as using generalized M-estimators as proposed in Wang et al. (2023b), enhancing the adaptability of our models to diverse data structures. Additionally, the extension of our models to longitudinal data could yield practical benefits, given that our methods rely on independent observations at each time point, a requirement that may not align with the characteristics of many real-world datasets. For example, in Section 3.5.1, where we analyze Canadian temperature data, the observations consist of repeated measurements taken from the same set of weather stations at specific hours of the day over consecutive days. While we have explored the independence assumption through residual plots, including Durbin-Watson statistics and ACF plots, which suggest it is reasonable to treat them as almost independent, there remains a necessity to develop new methods that can adapt to multivariate longitudinal data and address violations of independence assumptions. Techniques such as generalized estimating equations may offer potential solutions to navigate this challenge. Another further refinements involve addressing the assumptions underlying degrees of freedom in dynamic models. Current derivations, based on Stein's unbiased risk estimation, assume homoscedasticity. A future goal is to untangle this assumption, providing more realistic insights into time-varying models.

In Chapter 4, we discuss the interesting asymptotic property of the Tweedie SBM that facilitates the separate estimation of covariate effects and other model parameters. However, the absence of such a property in the Bernoulli SBM poses a puzzle, as we have yet to uncover the underlying mechanism that determines its validity or lack thereof. This intriguing observation motivates further exploration into the behaviors of latent variables and mixture models within the Tweedie distribution. Our findings underscore the potential of this distribution, prompting deeper investigation, particularly in the realm of finite mixtures of generalized linear models and other latent variable models, such as the latent space model. Another direction based on the current work in Chapter 4 is to extend our models to accommodate dynamic community labels, as exemplified in previous works by Xu and Hero (2014) and Matias and Miele (2017). Drawing inspiration from works using Markov chains, this extension promises to enhance the adaptability and realism of our models.

Extending our models in Chapter 4 to address zero-inflated data commonly encountered in real-world scenarios, where there is an excess probability of observing zero values compared to the current distributions, is another important direction. Our initial goal is to apply our model to datasets generated from true zero-inflated distributions and then develop methodologies to effectively handle and analyze such data. This extension will enhance the versatility of our models in analyzing zero-inflated datasets.

Many directions can be considered based on the current work in Chapter 5. First, addressing computational efficiency is a priority. Our current model computation involves applying the EM algorithm, with the E-step executed using Gibbs sampling. While effective, this approach operates at a relatively slow pace. To address this constraint, we are actively exploring some novel gradient-based discrete MCMC methods as an alternative to traditional Gibbs sampling. Recent methods in state-of-the-art publications (Sun et al., 2023a,b, 2022) offer potential paths. These methods generalize the Langevin Monte Carlo, a widely recognized gradient-based MCMC sampler that utilizes Langevin dynamics to iteratively sample from a target distribution until reaching a stationary state, to discrete space. Second, beyond computational advancements, our future work aims to address one significant yet sometimes unrealistic assumption in our current model—that all pairs of nodes will eventually experience the interactions. To mitigate this limitation, we propose integrating the proportional hazards cure model, an extension of the standard proportional hazards model, which incorporates a compound distribution to account for the presence of interactions. Relevant studies exploring this approach can be found in Song et al. (2023) and Williford (2021). Further justifying the rate of occurrences of the interaction will allow us to analyze the political datasets like those in Weisiger (2016), Leeds et al. (2009), and Bennett (1997). Third, one natural direction of the work in Chapter 5 is to incorporate the time-varying covariate in the current Cox proportional hazard model, as the time-varying covariate is common in political science.

In conclusion, this chapter outlines the trajectories of our work to estimate the edges, cluster the nodes, and analyze the timestamps of interactions in time-varying networks. As we move forward, the objective is to contribute to the advancement of relational data, particularly using the Tweedie distribution in latent models and analyzing the latent community structures in the survival models.

References

Ahmed, A. and Xing, E. P. (2009). Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences*, 106(29):11878–11883.

Aicher, C., Jacobs, A. Z., and Clauset, A. (2013). Adapting the stochastic block model to edge-weighted networks. In *ICML Workshop on Structured Learning (SLG)*.

Aicher, C., Jacobs, A. Z., and Clauset, A. (2015). Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248.

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(65):1981–2014.

Amini, A. A., Chen, A., Bickel, P. J., and Levina, E. (2013). Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics*, 41(4):2097–2122.

Bakhthemmat, A. and Izadi, M. (2021). Communities detection for advertising by futuristic greedy method with clustering approach. *Big Data*, 9(1):22–40.

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

Bartlett, T. E., Kosmidis, I., and Silva, R. (2021). Two-way sparsity for time-varying networks with applications in genomics. *The Annals of Applied Statistics*, 15(2):856–879.

Bayer, R. (2006). Diplomatic Exchange Data set, v2006.1. http://correlatesofwar.org.

Bedi, P. and Sharma, C. (2016). Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(3):115–135.

Bennett, D. S. (1997). Testing alternative models of alliance duration, 1816-1984. *American Journal of Political Science*, 41(3):846–878.

Berquin, P., Giedd, J., Jacobsen, L., Hamburger, S., Krain, A., Rapoport, J., and Castellanos, F. (1998). Cerebellum in attention-deficit hyperactivity disorder: a morphometric MRI study. *Neurology*, 50(4):1087–1093.

Bhattacharjee, M., Banerjee, M., and Michailidis, G. (2020). Change point estimation in a dynamic stochastic block model. *Journal of Machine Learning Research*, 21(107):1–59.

Bickel, P. J. and Chen, A. (2009). A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073.

Bickel, P. J., Chen, A., and Levina, E. (2011). The method of moments and degree distributions for network models. *The Annals of Statistics*, 39(5):2280 – 2301.

Bickel, P. J. and Levina, E. (2008). Covariance regularization by thresholding. *The Annals of Statistics*, 36(6):2577–2604.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Box-Steffensmeier, J. M. and Jones, B. S. (1997). Time is of the essence: Event history models in political science. *American Journal of Political Science*, 41(4):1414–1461.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122.

Cai, B., Zhang, J., and Guan, Y. (2022). Latent network structure learning from high-dimensional multivariate point processes. *Journal of the American Statistical Association*, 119(545):95–108.

Chen, K. and Lei, J. (2018). Network cross-validation for determining the number of communities in network data. *Journal of the American Statistical Association*, 113(521):241–251.

Chen, L.-P. and Yi, G. Y. (2021). Analysis of noisy survival data with graphical proportional hazards measurement error models. *Biometrics*, 77(3):956–969.

Chen, S., Witten, D. M., and Shojaie, A. (2015). Selection and estimation for mixed graphical models. *Biometrika*, 102(1):47–64.

Chen, Y. and Mo, D. (2022). Community detection for multilayer weighted networks. *Information Sciences*, 595:119–141.

Choi, D. S., Wolfe, P. J., and Airoldi, E. M. (2012). Stochastic blockmodels with a growing number of classes. *Biometrika*, 99(2):273–284.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 34(2):187–202.

Danaher, P., Wang, P., and Witten, D. M. (2014). 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.

Danon, L., Diaz-Guilera, A., Duch, J., and Arenas, A. (2005). Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008.

Daudin, J.-J., Picard, F., and Robin, S. (2008). A mixture model for random graphs. *Statistics and Computing*, 18(2):173–183.

Decelle, A., Krzakala, F., Moore, C., and Zdeborová, L. (2011). Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical review E*, 84:066106.

Dempster, A. P. (1972). Covariance selection. *Biometrics*, 28(1):157–175.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39(1):1–22.

Dohlman, A. B. and Shen, X. (2019). Mapping the microbial interactome: Statistical and experimental approaches for microbiome network inference. *Experimental Biology and Medicine*, 244(6):445–458.

Drton, M. and Maathuis, M. H. (2017). Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4:365–393.

Dunn, P. K. and Smyth, G. K. (2005). Series evaluation of Tweedie exponential dispersion model densities. *Statistics and Computing*, 15(4):267–280.

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, 18(1):73–86.

Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, 81(394):461–470.

Emmert-Streib, F., Glazko, G., and De Matos Simoes, R. (2012). Statistical inference and reverse engineering of gene regulatory networks from observational expression data. *Frontiers in Genetics*, 3:8–8.

Environment and Climate Change Canada (2022). Environment and Climate Change Canada Historical Climate Data website. https://climate.weather.gc.ca/index_e.html. Accessed: 2022-10-16.

Fang, G., Ward, O. G., and Zheng, T. (2024). Online estimation and community detection of network point processes for event streams. *Statistics and Computing*, 34(1):35.

Ferreira, L. N., Ferreira, N. C., Macau, E. E., and Donner, R. V. (2021). The effect of time series distance functions on functional climate networks. *The European Physical Journal Special Topics*, 230(14):2973–2998.

Fortna, V. P. (2018). *Peace time: Cease-fire agreements and the durability of peace*. Princeton University Press.

Fortunato, S. (2010). Community detection in graphs. *Physics reports*, 486(3-5):75–174.

Foygel, R. and Drton, M. (2010). Extended bayesian information criteria for gaussian graphical models. *Advances in Neural Information Processing Systems*, 23.

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

Friston, K. J. (1994). Functional and effective connectivity in neuroimaging: a synthesis. *Human brain mapping*, 2(1-2):56–78.

Fu, W., Song, L., and Xing, E. P. (2009). Dynamic mixed membership blockmodel for evolving networks. In *Proceedings of the 26th annual international conference on machine learning*, pages 329–336.

Gasparetti, F., Sansonetti, G., and Micarelli, A. (2021). Community detection in social recommender systems: a survey. *Applied Intelligence*, 51(6):3975–3995.

Gibberd, A. J. and Nelson, J. D. (2017). Regularized estimation of piecewise constant gaussian graphical models: The group-fused graphical lasso. *Journal of Computational and Graphical Statistics*, 26(3):623–634.

Glowinski, R. (2014). *On Alternating Direction Methods of Multipliers: A Historical Perspective*. Springer Netherlands, Dordrecht.

Green, P. J. and Silverman, B. W. (1993). *Nonparametric regression and generalized linear models: a roughness penalty approach*. Crc Press, New York.

Guerrero-Solé, F. (2017). Community detection in political discussions on twitter: An application of the retweet overlap network method to the catalan process toward independence. *Social science computer review*, 35(2):244–261.

Haj, A. E., Slaoui, Y., Louis, P.-Y., and Khraibani, Z. (2022). Estimation in a binomial stochastic blockmodel for a weighted graph by a variational expectation maximization algorithm. *Communications in Statistics-Simulation and Computation*, 51(8):4450–4469.

Hallac, D., Park, Y., Boyd, S., and Leskovec, J. (2017). Network inference via the time-varying graphical lasso. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 205–213.

Han, I., Malioutov, D., Avron, H., and Shin, J. (2017). Approximating spectral sums of large-scale matrices using stochastic Chebyshev approximations. *SIAM Journal on Scientific Computing*, 39(4):A1558–A1585.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. New York: springer.

He, Y., Sun, J., Tian, Y., Ying, Z., and Feng, Y. (2023). Semiparametric modeling and analysis for longitudinal network data. *arXiv preprint arXiv:2308.12227*.

Hecker, M., Lambeck, S., Toepfer, S., Van Someren, E., and Guthke, R. (2009). Gene regulatory network inference: data integration in dynamic models — a review. *Biosystems*, 96(1):86–103.

Hoefling, H. (2010). A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137.

Huang, S., Sun, J., and Feng, Y. (2023). PCABM: Pairwise covariates-adjusted block model for community detection. *Journal of the American Statistical Association*, 0(0):1–26.

Jaakkola, T. S. and Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10(1):25–37.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., and Barabási, A.-L. (2000). The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

Kang, X., Deng, X., Tsui, K.-W., and Pourahmadi, M. (2020). On variable ordination of modified cholesky decomposition for estimating time-varying covariance matrices. *International Statistical Review*, 88(3):616–641.

Kittel, T., Ciemer, C., Lotfi, N., Peron, T., Rodrigues, F., Kurths, J., and Donner, R. V. (2021). Evolving climate network perspectives on global surface air temperature effects of enso and strong volcanic eruptions. *The European Physical Journal Special Topics*, 230(14):3075–3100.

Kolar, M., Song, L., Ahmed, A., and Xing, E. P. (2010). Estimating time-varying networks. *The Annals of Applied Statistics*, 4(1):94–123.

Kolar, M. and Xing, E. P. (2011). On time varying undirected graphs. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pages 407–415, Fort Lauderdale, FL, USA.

Larsen, K. (2005). The cox proportional hazards model with a continuous latent variable measured by multiple binary indicators. *Biometrics*, 61(4):1049–1055.

Lauritzen, S. L. (1996). *Graphical Models.* The Clarendon Press, Oxford.

Lee, C. and Wilkinson, D. J. (2019). A review of stochastic block models and extensions for graph clustering. *Applied Network Science*, 4(1):1–50.

Leeds, B. A., Mattes, M., and Vogel, J. S. (2009). Interests, institutions, and the reliability of international commitments. *American Journal of Political Science*, 53(2):461–476.

Levina, E., Rothman, A., and Zhu, J. (2008). Sparse estimation of large covariance matrices via a nested lasso penalty. *The Annals of Applied Statistics*, 2(1):245–263.

Lian, Y., Yang, A. Y., Wang, B., Shi, P., and Platt, R. W. (2023). A Tweedie compound Poisson model in reproducing kernel Hilbert space. *Technometrics*, 65(2):281–295.

Loh, P.-L. (2017). Statistical consistency and asymptotic normality for high-dimensional robust M-estimators. *The Annals of Statistics*, 45(2):866 – 896.

Loyle, C. E. and Appel, B. J. (2017). Conflict recurrence and postconflict justice: Addressing motivations and opportunities for sustainable peace. *International Studies Quarterly*, 61(3):690–703.

Lu, Z., Wahlström, J., and Nehorai, A. (2018). Community detection in complex networks via clique conductance. *Scientific reports*, 8(1):1–16.

Ludkin, M. (2020). Inference for a generalised stochastic block model with unknown number of blocks and non-conjugate edge models. *Computational Statistics & Data Analysis*, 152:107051.

Lurie, D. J., Kessler, D., Bassett, D. S., Betzel, R. F., Breakspear, M., Kheilholz, S., Kucyi, A., Liégeois, R., Lindquist, M. A., and McIntosh, A. R. (2020). Questions and controversies in the study of time-varying functional connectivity in resting fMRI. *Network Neuroscience*, 4(1):30–69.

Ma, Z. and Ma, Z. (2017). Exploration of large networks with covariates via fast and universal latent space model fitting. *arXiv preprint arXiv:1705.02372*.

MacDonald, P. W., Levina, E., and Zhu, J. (2022). Latent space models for multiplex networks with shared structure. *Biometrika*, 109(3):683–706.

Mariadassou, M., Robin, S., and Vacher, C. (2010). Uncovering latent structure in valued graphs: a variational approach. *The Annals of Applied Statistics*, 4(2):715–742.

Matias, C. and Miele, V. (2017). Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 79(4):1119–1141.

Matias, C., Rebafka, T., and Villers, F. (2018). A semiparametric extension of the stochastic block model for longitudinal networks. *Biometrika*, 105(3):665–680.

Meinshausen, N. (2008). A note on the lasso for gaussian graphical model selection. *Statistics & Probability Letters*, 78(7):880–884.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473.

Menon, S. S. and Krishnamurthy, K. (2019). A comparison of static and dynamic functional connectivities for identifying subjects and biological sex using intrinsic individual brain connectivity. *Scientific reports*, 9(1):1–11.

Meyer, M. and Woodroofe, M. (2000). On the degrees of freedom in shape-restricted regression. *Annals of Statistics*, 28(4):1083–1104.

Monti, R. P., Hellyer, P., Sharp, D., Leech, R., Anagnostopoulos, C., and Montana, G. (2014). Estimating time-varying brain connectivity networks from functional MRI time series. *NeuroImage*, 103:427–443.

Mostofsky, S. H., Reiss, A. L., Lockhart, P., and Denckla, M. B. (1998). Evaluation of cerebellar size in attention-deficit hyperactivity disorder. *Journal of Child Neurology*, 13(9):434–439.

Motalebi, N., Stevens, N. T., and Steiner, S. H. (2021). Hurdle blockmodels for sparse network modeling. *The American Statistician*, 75(4):383–393.

Neal, R. M. and Hinton, G. E. (1998). A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer.

Newman, M. E. and Clauset, A. (2016). Structure and inference in annotated networks. *Nature communications*, 7(1):1–11.

Ng, T. L. J. and Murphy, T. B. (2021). Weighted stochastic block model. *Statistical Methods & Applications*, 30:1365–1398.

Pei, Y., Peng, H., and Xu, J. (2024). A latent class cox model for heterogeneous time-to-event data. *Journal of Econometrics*, 239(2).

Peixoto, T. P. (2018). Nonparametric weighted stochastic block models. *Physical Review E*, 97(1):012306.

Peng, J., Wang, P., Zhou, N., and Zhu, J. (2009). Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746.

Perry, P. O. and Wolfe, P. J. (2013). Point process modelling for directed interaction networks. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 75(5):821–849.

Raghavan, U. N., Albert, R., and Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106.

Roy, S., Atchadé, Y., and Michailidis, G. (2019). Likelihood inference for large scale stochastic blockmodels with covariates based on a divide-and-conquer parallelizable algorithm with communication. *Journal of Computational and Graphical Statistics*, 28(3):609–619.

Saggar, M. and Uddin, L. Q. (2019). Pushing the boundaries of psychiatric neuroimaging to ground diagnosis in biology. *ENeuro*, 6(6).

Sarma, D., Alam, W., Saha, I., Alam, M. N., Alam, M. J., and Hossain, S. (2020). Bank fraud detection using community detection algorithm. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 642–646. IEEE.

Sato, T., Yamanishi, Y., Horimoto, K., Kanehisa, M., and Toh, H. (2006). Partial correlation coefficient between distance matrices as a new indicator of protein–protein interactions. *Bioinformatics*, 22(20):2488–2492.

Schwikowski, B., Uetz, P., and Fields, S. (2000). A network of protein–protein interactions in yeast. *Nature biotechnology*, 18(12):1257–1261.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 47(1):1–21.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *Journal of computational and graphical statistics*, 22(2):231–245.

Sit, T., Ying, Z., and Yu, Y. (2021). Event history analysis of dynamic networks. *Biometrika*, 108(1):223–230.

Snijders, T. A. and Nowicki, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*, 14(1):75–100.

Song, F., Chu, J., Ma, S., and Wei, Y. (2023). Survival mixed membership blockmodel. *Journal of the American Statistical Association*, pages 1–19.

Stein, C. M. (1981). Estimation of the mean of a multivariate normal distribution. *Annals of Statistics*, 9(6):1135–1151.

Sun, H., Dai, H., Dai, B., Zhou, H., and Schuurmans, D. (2023a). Discrete langevin samplers via wasserstein gradient flow. In *International Conference on Artificial Intelligence and Statistics*, pages 6290–6313. PMLR.

Sun, H., Dai, H., and Schuurmans, D. (2022). Optimal scaling for locally balanced proposals in discrete spaces. *Advances in Neural Information Processing Systems*, 35:23867–23880.

Sun, H., Goshvadi, K., Nova, A., Schuurmans, D., and Dai, H. (2023b). Revisiting sampling for combinatorial optimization. In *International Conference on Machine Learning*, pages 32859–32874. PMLR.

Tallberg, C. (2004). A bayesian approach to modeling stochastic blockstructures with covariates. *Journal of Mathematical Sociology*, 29(1):1–23.

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

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(1):91–108.

Tibshirani, R. J. and Taylor, J. (2011). The solution path of the generalized lasso. *The Annals of Statistics*, 39(3):1335–1371.

Tibshirani, R. J. and Taylor, J. (2012). Degrees of freedom in lasso problems. *The Annals of Statistics*, 40(2):1198–1232.

Tweedie, M. C. (1984). An index which distinguishes between some important exponential families. In *Statistics: Applications and New Directions: Proc. Indian Statistical Institute Golden Jubilee International Conference*, volume 579, pages 579–604.

Voorman, A., Shojaie, A., and Witten, D. (2014). Graph estimation with joint additive models. *Biometrika*, 101(1):85–101.

Vu, D. Q., Hunter, D. R., and Schweinberger, M. (2013). Model-based clustering of large networks. *The Annals of Applied Statistics*, 7(2):1010.

Wang, H. (2012). Bayesian graphical lasso models and efficient posterior computation. *Bayesian Analysis*, 7(4):867–886.

Wang, J., Zhang, J., Liu, B., Zhu, J., and Guo, J. (2023a). Fast network community detection with profile-pseudo likelihood methods. *Journal of the American Statistical Association*, 118(542):1359–1372.

Wang, S., Xie, C., and Kang, X. (2023b). A novel robust estimation for high-dimensional precision matrices. *Statistics in Medicine*, 42(5):656–675.

Wang, Y. R. and Bickel, P. J. (2017). Likelihood-based model selection for stochastic block models. *The Annals of Statistics*, 45(2):500–528.

Wei, G. C. and Tanner, M. A. (1990). A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704.

Weisiger, A. (2013). *Logics of war: Explanations for limited and unlimited conflicts*. Cornell University Press.

Weisiger, A. (2016). Exiting the coalition: When do states abandon coalition partners during war? *International Studies Quarterly*, 60(4):753–765.

Whittaker, J. (1990). *Graphical Models in Applied Multivariate Statistics*. John Wiley & Sons Ltd., Chichester.

Williford, G. (2021). *Using the Proportional Hazards Cure Model to Improve the Study of International Relations*. PhD thesis, University of Georgia.

World Integrated Trade Solution (2023). International merchandise trade, tariff and non-tariff measures (NTM) data. https://wits.worldbank.org/.

Xin, B., Kawahara, Y., Wang, Y., Hu, L., and Gao, W. (2016). Efficient generalized fused lasso and its applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(4):1–22.

Xin, L., Zhu, M., and Chipman, H. (2017). A continuous-time stochastic block model for basketball networks. *The Annals of Applied Statistics*, 11(2):553–597.

Xing, E. P., Fu, W., and Song, L. (2010). A state-space mixed membership blockmodel for dynamic network tomography. *The Annals of Applied Statistics*, 4(2):535–566.

Xu, K. S. and Hero, A. O. (2014). Dynamic stochastic blockmodels for time-evolving social networks. *IEEE Journal of Selected Topics in Signal Processing*, 8(4):552–562.

Xue, L., Shu, X., and Qu, A. (2020). Time-varying estimation and dynamic model selection with an application of network data. *Statistica Sinica*, 30(1):251–284.

Yang, E., Ravikumar, P., Allen, G. I., and Liu, Z. (2015a). Graphical models via univariate exponential family distributions. *The Journal of Machine Learning Research*, 16(1):3813–3847.

Yang, J. and Peng, J. (2020). Estimating time-varying graphical models. *Journal of Computational and Graphical Statistics*, 29(1):191–202.

Yang, S., Lu, Z., Shen, X., Wonka, P., and Ye, J. (2015b). Fused multiple graphical lasso. *SIAM Journal on Optimization*, 25(2):916–943.

Yang, T., Chi, Y., Zhu, S., Gong, Y., and Jin, R. (2011). Detecting communities and their evolutions in dynamic social networks—a bayesian approach. *Machine learning*, 82(2):157–189.

Ye, G. and Xie, X. (2011). Split Bregman method for large scale fused Lasso. *Computational Statistics & Data Analysis*, 55(4):1552–1569.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35.

Zhang, J. and Cao, J. (2017). Finding common modules in a time-varying network with application to the *drosophila melanogaster* gene regulation network. *Journal of the American Statistical Association*, 112(519):994–1008.

Zhang, J. and Chen, Y. (2020). Modularity based community detection in heterogeneous networks. *Statistica Sinica*, 30(2):601–629.

Zhang, J., Kucyi, A., Raya, J., Nielsen, A. N., Nomi, J. S., Damoiseaux, J. S., Greene, D. J., Horovitz, S. G., Uddin, L. Q., and Whitfield-Gabrieli, S. (2021). What have we really learned from functional connectivity in clinical populations? *NeuroImage*, 242:118466.

Zhang, J., Sun, W. W., and Li, L. (2020). Mixed-effect time-varying network model and application in brain connectivity analysis. *Journal of the American Statistical Association*, 115(532):2022–2036.

Zhang, M., Zhang, J., and Dai, W. (2023). Fast community detection in dynamic and heterogeneous networks. *Journal of Computational and Graphical Statistics*, pages 1–14.

Zhao, Y., Levina, E., and Zhu, J. (2011). Community extraction for social networks. *Proceedings of the National Academy of Sciences*, 108(18):7321–7326.

Zhou, S., Lafferty, J., and Wasserman, L. (2010). Time varying undirected graphs. *Machine Learning*, 80(2-3):295–319.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

Zou, H., Hastie, T., and Tibshirani, R. (2007). On the"degrees of freedom" of the lasso. *The Annals of Statistics*, 35(5):2173–2192.

APPENDICES

Appendix A

Appendix for Chapter 3

A.1 Fast Computation of $\boldsymbol{\theta}$-update in ADMM

This section introduces the details of fast updating $\boldsymbol{\theta}$ in the ADMM algorithm discussed in Section 3.1. In the $\boldsymbol{\theta}-$update identified at Equation (15), $\boldsymbol{\mathcal{X}}^\top \boldsymbol{\mathcal{X}}$ is a diagonal block matrix with the kth diagonal block $\boldsymbol{\mathcal{X}}^\top(t_k)\boldsymbol{\mathcal{X}}(t_k)$, and $\boldsymbol{D}^\top \boldsymbol{D}$ is a $Tp(p-1)/2 \times Tp(p-1)/2$ block tridiagonal matrix given by

$$\boldsymbol{D}^\top \boldsymbol{D} = \begin{bmatrix} I & -I & 0 & 0 & 0 & 0 \\ -I & 2I & -I & 0 & 0 & 0 \\ 0 & -I & 2I & -I & 0 & 0 \\ & & & \ddots & & \\ 0 & 0 & 0 & -I & 2I & -I \\ 0 & 0 & 0 & 0 & -I & I \end{bmatrix},$$

where each block matrix has dimension $p(p-1)/2 \times p(p-1)/2$.

It follows that

$$\frac{2}{n}\boldsymbol{\mathcal{X}}^\top \boldsymbol{\mathcal{X}} + 2\lambda_2 \boldsymbol{D}^\top \boldsymbol{D} + aI = 2\lambda_2 \begin{bmatrix} A_1 & -I & 0 & 0 & 0 & 0 \\ -I & A_2 & -I & 0 & 0 & 0 \\ 0 & -I & A_3 & -I & 0 & 0 \\ & & & \ddots & & \\ 0 & 0 & 0 & -I & A_{T-1} & -I \\ 0 & 0 & 0 & 0 & -I & A_T \end{bmatrix}, \tag{A.1}$$

where

$$A_i = \begin{cases} \frac{\boldsymbol{\mathcal{X}}(t_i)^\top \boldsymbol{\mathcal{X}}(t_i)}{n\lambda_2} + (1 + \frac{a}{2\lambda_2})I, \text{ if } i = 1 \text{ and } T \\ \frac{\boldsymbol{\mathcal{X}}(t_i)^\top \boldsymbol{\mathcal{X}}(t_i)}{n\lambda_2} + (2 + \frac{a}{2\lambda_2})I. \text{ otherwise} \end{cases}.$$

Denote by H the matrix on the right-hand side of (A.1). Updating $\boldsymbol{\theta}$ in (15) is to solve the linear system $H\boldsymbol{\theta} = \big(2n^{-1}\boldsymbol{\mathcal{Y}}^\top \boldsymbol{\mathcal{X}} + a(\boldsymbol{z} - \boldsymbol{u})\big)$. We provide an efficient approach to find the inverse of H.

We first multiply H by a sequence of lower triangular matrices, denoted by L_i's, on its left to convert H to an upper triangular matrix. In particular, we take $T = 4$, i.e., four time points, as an example to illustrate this procedure. Simple algebra yields the following three steps:

$$(1) \quad L_1 H = \begin{bmatrix} I & 0 & 0 & 0 \\ A_1^{-1} & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} A_1 & -I & 0 & 0 \\ -I & A_2 & -I & 0 \\ 0 & -I & A_3 & -I \\ 0 & 0 & -I & A_4 \end{bmatrix} = \begin{bmatrix} A_1 & -I & 0 & 0 \\ 0 & A_2 - A_1^{-1} & -I & 0 \\ 0 & -I & A_3 & -I \\ 0 & 0 & -I & A_4 \end{bmatrix}.$$

$$\begin{aligned} (2) \quad L_2 L_1 H &= \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & (A_2 - A_1^{-1})^{-1} & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} A_1 & -I & 0 & 0 \\ 0 & A_2 - A_1^{-1} & -I & 0 \\ 0 & -I & A_3 & -I \\ 0 & 0 & -I & A_4 \end{bmatrix} \\ &= \begin{bmatrix} A_1 & -I & 0 & 0 \\ 0 & A_2 - A_1^{-1} & -I & 0 \\ 0 & 0 & A_3 - (A_2 - A_1^{-1})^{-1} & -I \\ 0 & 0 & -I & A_4 \end{bmatrix}. \end{aligned}$$

$$\begin{aligned} (3) \quad L_3 L_2 L_1 H &= \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & (A_3 - (A_2 - A_1^{-1})^{-1})^{-1} & I \end{bmatrix} \begin{bmatrix} A_1 & -I & 0 & 0 \\ 0 & A_2 - A_1^{-1} & -I & 0 \\ 0 & 0 & A_3 - (A_2 - A_1^{-1})^{-1} & -I \\ 0 & 0 & -I & A_4 \end{bmatrix} \\ &= \begin{bmatrix} A_1 & -I & 0 & 0 \\ 0 & A_2 - A_1^{-1} & -I & 0 \\ 0 & 0 & A_3 - (A_2 - A_1^{-1})^{-1} & -I \\ 0 & 0 & 0 & A_4 - (A_3 - (A_2 - A_1^{-1})^{-1})^{-1} \end{bmatrix}. \end{aligned}$$

Let $B_1 = A_1^{-1}$ and $B_i = (A_i - B_{i-1})^{-1}$. The formula above can be rewritten as:

$$\begin{aligned} L_3L_2L_1H &= \begin{bmatrix} I & 0 & 0 & 0 \\ B_1 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & B_2 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & B_3 & I \end{bmatrix} H \\ &= \begin{bmatrix} B_1^{-1} & -I & 0 & 0 \\ 0 & B_2^{-1} & -I & 0 \\ 0 & 0 & B_3^{-1} & -I \\ 0 & 0 & 0 & B_4^{-1} \end{bmatrix}. \end{aligned}$$

To eliminate upper off-diagonal blocks, we define a sequence of upper triangular matrices:

$$U_1 = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & B_4 \\ 0 & 0 & 0 & I \end{bmatrix}, U_2 = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & B_3 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}, U_3 = \begin{bmatrix} I & B_2 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}.$$

Sequentially multiplying by U_i's on the left of $L_3L_2L_1H$ yields

$$U_3U_2U_1L_3L_2L_1H = \begin{bmatrix} B_1^{-1} & 0 & 0 & 0 \\ 0 & B_2^{-1} & 0 & 0 \\ 0 & 0 & B_3^{-1} & 0 \\ 0 & 0 & 0 & B_4^{-1} \end{bmatrix}.$$

Lastly, define

$$\tilde{B}_i = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & B_i & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}$$

for $i = 1, 2, 3$. Then $\tilde{B}_4\tilde{B}_3\tilde{B}_2\tilde{B}_1U_3U_2U_1L_3L_2L_1H$ is an identity matrix.

In summary, an updated $\boldsymbol{\theta}$ can be obtained through

$$\begin{aligned}
\boldsymbol{\theta}^{k+1} &= \left(\frac{2}{n}\boldsymbol{\mathcal{X}}^\top\boldsymbol{\mathcal{X}} + \frac{2\lambda_2}{n}\boldsymbol{D}^\top\boldsymbol{D} + aI\right)^{-1}\left[\frac{2}{n}\boldsymbol{\mathcal{X}}^\top\boldsymbol{\mathcal{Y}} + \alpha(z^k - u^k)\right] \\
&= \frac{n}{2\lambda_2}\cdot H^{-1}\cdot\left[\frac{2}{n}\boldsymbol{\mathcal{X}}^\top\boldsymbol{\mathcal{Y}} + a(z^k - u^k)\right] \\
&= \tilde{B}_4\tilde{B}_3\tilde{B}_2\tilde{B}_1U_3U_2U_1L_3L_2L_1\frac{n}{2\lambda_2}\cdot\left[\frac{2}{n}\boldsymbol{\mathcal{X}}^\top\boldsymbol{\mathcal{Y}} + a(z^k - u^k)\right].
\end{aligned}$$

This sequence of operations enables us to quickly find the inverse of $2n^{-1}\boldsymbol{\mathcal{X}}^\top\boldsymbol{\mathcal{X}}+2\lambda_2n^{-1}\boldsymbol{D}^\top\boldsymbol{D}+aI$, thus the computational efficiency of the ADMM algorithm is greatly enhanced.

A.2 Degrees of Freedom in GEN

In this section, we derive the degrees of freedom in GEN under the framework of Stein's unbiased risk estimation (SURE) (Stein, 1981). Zou et al. (2007) provides a theoretical justification of the degrees of freedom in the standard LASSO problem, and we will use it to derive an unbiased estimate of the degrees of freedom in the GEN problem with a homoscedastic assumption. We state the main theorem as the following.

Theorem S2. *Suppose $\boldsymbol{y}_{n\times 1} \sim N(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$, where $\boldsymbol{\mu} \in \mathbb{R}^n$ denotes the mean vector and σ^2 denotes the common variance of each component. Given a design matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ and two tuning parameters, λ_1 and λ_2, we consider the generalized elastic net problem*

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\left\{\frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{D}\boldsymbol{\beta}\|_2^2\right\}, \tag{A.2}$$

where $\boldsymbol{D}_{m\times p}$ is defined as in Equation (9). If definition of degrees of freedom is given by Equation (A.4), then

$$df(\boldsymbol{X}\hat{\boldsymbol{\beta}}) = Tr\left(\boldsymbol{X}_{\mathcal{A}}\left(\boldsymbol{X}_{\mathcal{A}}^\top\boldsymbol{X}_{\mathcal{A}} + n\lambda_2\boldsymbol{D}_{\mathcal{A}}^\top\boldsymbol{D}_{\mathcal{A}}\right)^{-1}\boldsymbol{X}_{\mathcal{A}}^\top\right).$$

Here $\mathcal{A} = \{j : \hat{\boldsymbol{\beta}}_j \neq 0\}$ denotes the collection of column indices of $\boldsymbol{X}$ corresponding to the active features, and $B_{\mathcal{A}}$ denotes a submatrix of B that contains only the columns indexed by $\mathcal{A}$.

To prove Theorem S2, we first introduce definition of effective degrees of freedom for a general fitted function as described in Lemma S1. Furthermore, Lemma S2 indicates that if the fitted function is almost differentiable, the effective degrees of freedom can be simplified as the gradient of the fitted function. In Lemma S3, we give an explicit form of the fitted function in GEN. Lemma S4 shows that the fitted function of GEN is uniformly Lipschitz, and thus by Lemma S5 the fitted function in GEN is almost differentiable.

One natural definition of effective degrees of freedom comes from the well-known identity of *optimism* in Efron (1986).

Lemma S1 (Optimism theorem (Efron, 1986))**.** *Suppose* $\boldsymbol{y}_{n\times 1} \sim (\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$*, where* $\boldsymbol{\mu}$ *is the true mean vector and* σ^2 *is the common variance of each component. Let* $\hat{\boldsymbol{\mu}} = \delta(\boldsymbol{y})$ *denote the fitted function of some fitting technique* δ*, and* $\boldsymbol{y}^{new}$ *be the new response vector generated from the distribution* $(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$*. Then*

$$\mathbb{E}\left\{\|\boldsymbol{y}^{new} - \hat{\boldsymbol{\mu}}\|^2\right\} - E\left\{\|\boldsymbol{y} - \hat{\boldsymbol{\mu}}\|^2\right\} = 2\sum_{i=1}^{n} \operatorname{cov}\left(y_i, \hat{\mu}_i\right). \tag{A.3}$$

The right-hand side of (A.3) is referred to as *the optimism* of the estimator $\hat{\boldsymbol{\mu}}$. Based on (A.3), the degrees of freedom can be defined as

$$df\left(\hat{\boldsymbol{\mu}}\right) = \frac{1}{\sigma^2}\sum_{i=1}^{n} \operatorname{cov}\left(y_i, \hat{\mu}_i\right). \tag{A.4}$$

Stein's Lemma (Stein, 1981) can further simplify the right-hand side of (A.4).

Lemma S2 (Stein's Lemma)**.** *Suppose that* $\hat{\boldsymbol{\mu}} : \mathbb{R}^n \to \mathbb{R}^n$ *is almost differentiable and let* $\nabla \cdot \hat{\boldsymbol{\mu}} = \sum_{i=1}^{n} \frac{\partial \hat{\mu}_i}{\partial y_i}$*. If* $\boldsymbol{y} \sim N(\boldsymbol{\mu}, \sigma^2 I)$*, then*

$$\frac{1}{\sigma^2}\sum_{i=1}^{n} \operatorname{cov}\left(y_i, \hat{\mu}_i\right) = \mathbb{E}(\nabla \cdot \hat{\boldsymbol{\mu}}). \tag{A.5}$$

Therefore, (A.4) and (A.5) imply that

$$\hat{df}\left(\boldsymbol{\mu}\right) = \nabla \cdot \hat{\boldsymbol{\mu}} \tag{A.6}$$

is an unbiased estimate of the degrees of freedom if $\hat{\boldsymbol{\mu}}$ is almost differentiable. Next, we will first find the fitted function $\hat{\boldsymbol{\mu}}$ in the GEN problem (A.2), and then show it is almost differentiable. Lastly, we find an explicit form of $\nabla \cdot \hat{\boldsymbol{\mu}}$ as the unbiased estimator of the (effective) degrees of freedom.

Lemma S3 (Solution to GEN)**.** *The GEN problem* (A.2) *can be written as a lasso-type problem with the augmented dataset* $\tilde{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y}_{n\times 1} \\ \boldsymbol{0}_{m\times 1} \end{bmatrix}$ *and* $\tilde{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{X}_{n\times p} \\ \sqrt{n\lambda_2}\cdot \boldsymbol{D}_{m\times p} \end{bmatrix}$*:*

$$\hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}} \frac{1}{n}\|\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{X}}\boldsymbol{\beta}\|^2 + \lambda_1\|\boldsymbol{\beta}\|_1.$$

Suppose that λ_1 and λ_2 are not the transition points where the active set $\mathcal{A}$ changes. The coefficient estimate is given by

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{\lambda_1,\lambda_2} &= \left(\tilde{\boldsymbol{X}}_{\mathcal{A}}^\top \tilde{\boldsymbol{X}}_{\mathcal{A}}\right)^{-1}\left(\tilde{\boldsymbol{X}}_{\mathcal{A}}^\top \tilde{\boldsymbol{y}} - \frac{\lambda_1}{2}\operatorname{sign}(\hat{\boldsymbol{\beta}}_{\mathcal{A}})\right) \\ &= \left(\boldsymbol{X}_{\mathcal{A}}^\top \boldsymbol{X}_{\mathcal{A}} + n\lambda_2 \boldsymbol{D}_{\mathcal{A}}^\top \boldsymbol{D}_{\mathcal{A}}\right)^{-1}\left(\boldsymbol{X}_{\mathcal{A}}^\top \boldsymbol{y} - \frac{\lambda_1}{2}\operatorname{sign}(\hat{\boldsymbol{\beta}}_{\mathcal{A}})\right), \end{aligned} \tag{A.7}$$

and the fitted function is

$$\begin{aligned} \hat{\boldsymbol{\mu}}(\boldsymbol{y}) &= \boldsymbol{X}_{\mathcal{A}}\hat{\boldsymbol{\beta}}_{\lambda_1,\lambda_2} \\ &= \boldsymbol{X}_{\mathcal{A}}\left(\boldsymbol{X}_{\mathcal{A}}^\top \boldsymbol{X}_{\mathcal{A}} + n\lambda_2 \boldsymbol{D}_{\mathcal{A}}^\top \boldsymbol{D}_{\mathcal{A}}\right)^{-1}\left(\boldsymbol{X}_{\mathcal{A}}^\top \boldsymbol{y} - \frac{\lambda_1}{2}\operatorname{sign}(\hat{\boldsymbol{\beta}}_{\mathcal{A}})\right). \end{aligned} \tag{A.8}$$

Lemma S4 (Lipschitz continuity of $\hat{\boldsymbol{\mu}}$)**.** *The GEN fitted function $\hat{\boldsymbol{\mu}}$ in* (A.8) *is 1-Lipschitz i.e., $\|\hat{\boldsymbol{\mu}}(\boldsymbol{y}+\Delta\boldsymbol{y}) - \hat{\boldsymbol{\mu}}(\boldsymbol{y})\| \leq \|\Delta\boldsymbol{y}\|$ for sufficiently small $\Delta\boldsymbol{y}$.*

Proof. Define a mapping $\tau : \mathbb{R}^n \to \mathbb{R}^{n+m}$ such that

$$\begin{aligned} \tau(\boldsymbol{y}) &= \tilde{\boldsymbol{X}}_{\mathcal{A}}\hat{\boldsymbol{\beta}}_{\lambda_1,\lambda_2} \\ &= \tilde{\boldsymbol{X}}_{\mathcal{A}}\left(\tilde{\boldsymbol{X}}_{\mathcal{A}}^\top \tilde{\boldsymbol{X}}_{\mathcal{A}}\right)^{-1}\left(\tilde{\boldsymbol{X}}_{\mathcal{A}}^\top \tilde{\boldsymbol{y}} - \frac{\lambda_1}{2}\operatorname{sign}(\hat{\boldsymbol{\beta}}_{\mathcal{A}})\right), \end{aligned} \tag{A.9}$$

where $\tilde{\boldsymbol{X}}_{\mathcal{A}}$ and $\tilde{\boldsymbol{y}}$ have been defined in Lemma S3. Then $\hat{\boldsymbol{\mu}}(\boldsymbol{y})$ is the first n components in $\tau(\boldsymbol{y})$, i.e., $\hat{\boldsymbol{\mu}}(\boldsymbol{y}) = T\cdot\tau(\boldsymbol{y})$ where $T := [\boldsymbol{I}_{n\times n}, \boldsymbol{0}_{n\times m}]$.

Since λ_1 and λ_2 are not the transition points, the active set $\mathcal{A}$ stays constant for sufficient small Δy. Then we have

$$\begin{aligned}\|\tau(\boldsymbol{y}+\Delta\boldsymbol{y})-\tau(\boldsymbol{y})\| &= \|\tilde{\boldsymbol{X}}_{\mathcal{A}}\left(\tilde{\boldsymbol{X}}_{\mathcal{A}}^{\top}\tilde{\boldsymbol{X}}_{\mathcal{A}}\right)^{-1}\left(\tilde{\boldsymbol{X}}_{\mathcal{A}}^{\top}\tilde{\boldsymbol{y}}-\frac{\lambda_1}{2}\operatorname{sign}(\hat{\boldsymbol{\beta}}_{\mathcal{A}})\right)\\ &\quad -\tilde{\boldsymbol{X}}_{\mathcal{A}}\left(\tilde{\boldsymbol{X}}_{\mathcal{A}}^{\top}\tilde{\boldsymbol{X}}_{\mathcal{A}}\right)^{-1}\left(\tilde{\boldsymbol{X}}_{\mathcal{A}}^{\top}(\boldsymbol{y}+\Delta\boldsymbol{y})-\frac{\lambda_1}{2}\operatorname{sign}(\hat{\boldsymbol{\beta}}_{\mathcal{A}})\right)\|\\ &= \|\tilde{\boldsymbol{X}}_{\mathcal{A}}\left(\tilde{\boldsymbol{X}}_{\mathcal{A}}^{\top}\tilde{\boldsymbol{X}}_{\mathcal{A}}\right)^{-1}\tilde{\boldsymbol{X}}_{\mathcal{A}}^{\top}\Delta\boldsymbol{y}\|\\ &\leq \|\Delta\boldsymbol{y}\|, \end{aligned} \tag{A.10}$$

where the last relation holds since $\tilde{\boldsymbol{X}}_{\mathcal{A}}\left(\tilde{\boldsymbol{X}}_{\mathcal{A}}^{\top}\tilde{\boldsymbol{X}}_{\mathcal{A}}\right)^{-1}\tilde{\boldsymbol{X}}_{\mathcal{A}}^{\top}$ is a projection matrix. Therefore, τ is Lipschitz continuous.

Now, we can show that $\hat{\boldsymbol{\mu}}$ is also Lipschitz continuous:

$$\begin{aligned}\|\hat{\mu}(\boldsymbol{y}+\Delta\boldsymbol{y})-\hat{\mu}(\boldsymbol{y})\| &= \|T\cdot\left(\hat{\tau}(\boldsymbol{y}+\Delta\boldsymbol{y})-\hat{\tau}(\boldsymbol{y})\right)\|\\ &\leq \|T\|\cdot\|\Delta\boldsymbol{y}\|\\ &= \|\Delta\boldsymbol{y}\|.\end{aligned}$$

□

Lemma S5 (Lipschitz continuity and differentiability (Meyer and Woodroofe, 2000))**.** *Any Lipschitz continuous function is almost differentiable.*

By local constancy of the active set $\mathcal{A}$ and Equation (A.8), it is straighforward to show

$$\nabla\cdot\hat{\boldsymbol{\mu}}=\sum_{i=1}^{n}\frac{\partial\hat{\mu}_i(\boldsymbol{y})}{\partial\boldsymbol{y}_i}=Tr[\boldsymbol{X}_{\mathcal{A}}\left(\boldsymbol{X}_{\mathcal{A}}^{\top}\boldsymbol{X}_{\mathcal{A}}+n\lambda_2\boldsymbol{D}_{\mathcal{A}}^{\top}\boldsymbol{D}_{\mathcal{A}}\right)^{-1}\boldsymbol{X}_{\mathcal{A}}^{\top}]. \tag{A.11}$$

The above results suggest that it is an unbiased estimator of degrees of freedom of $\hat{\boldsymbol{\mu}}$. Therefore, Theorem S2 is established.

A.3 Degrees of Freedom in GFL

Tibshirani and Taylor (2012) shows that the nullity of a particular penalty matrix is an unbiased estimator of the degrees of freedom defined in Equation (A.4); see Lemma S6. In this section, we provide an explicit form for this quantity in the setting of GFL.

Lemma S6 (Tibshirani and Taylor (2012))**.** *Suppose $\boldsymbol{y}_{n\times 1} \sim N(\mathbf{0}, \sigma^2 \boldsymbol{I})$, where $\boldsymbol{\mu}$ is the mean vector and σ^2 is the common variance of each component. Given a design matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ of full column rank and a tuning parameter λ, we consider the generalized lasso problem*

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{F}\boldsymbol{\beta}\|_1 \right\}, \tag{A.12}$$

where $\boldsymbol{F}_{m\times p}$ is an arbitrary penalty matrix. Then

$$df(\boldsymbol{X}\hat{\boldsymbol{\beta}}) = \mathbb{E}[nullity(\boldsymbol{F}_{-\mathcal{A}})], . \tag{A.13}$$

where $\mathcal{A} = \{i \in \{1, \cdots, m\} : (\boldsymbol{F}\hat{\boldsymbol{\beta}})_i \neq 0\}$.

In GFL, the penalty matrix $\lambda\boldsymbol{F}$ is composed of two parts: the top part is a difference matrix defined in (9) multiplied by the first tuning parameter λ_1 while the lower part is an identity matrix multiplied by another tuning parameter λ_2. Theorem S3 establishes a simple expression of (A.13) in GFL. As row operations preserve the null space, in the following calculations the tuning parameters are not considered for the penalty matrix for simplicity. Let $\beta = \left(\beta(1)^\top, \cdots, \beta(T)^\top\right)^\top$ where each $\beta(k)$ is a column vector of length p, and $\boldsymbol{F}$ be a $p(2T-1)$-by-pT matrix, where the first $p(T-1)$ rows constitute a difference matrix and the second pT rows constitute an identity matrix. We rewrite $\boldsymbol{F}$ as a block matrix using the $p \times p$ identity matrix I:

$$\boldsymbol{F} = \left[\begin{array}{cccccc}
I & -I & 0 & 0 & 0 & 0 \\
0 & I & -I & 0 & 0 & 0 \\
 & & \ddots & \ddots & & \\
0 & 0 & 0 & I & -I & 0 \\
0 & 0 & 0 & 0 & I & -I \\
\hdashline
I & 0 & 0 & 0 & 0 & 0 \\
0 & I & 0 & 0 & 0 & 0 \\
 & & & \ddots & & \\
0 & 0 & 0 & 0 & I & 0 \\
0 & 0 & 0 & 0 & 0 & I
\end{array}\right].$$

Theorem S3. *Let $\beta_j(k)$ denote the jth element of $\beta(k)$, $j = 1, \ldots, p, k = 1, \ldots, T$. If $\boldsymbol{F}_{-\mathcal{A}}$ denotes the submatrix of $\boldsymbol{F}$ after removing the rows indexed by $\mathcal{A} = \{i \in \{1, \cdots, p(2T-1)\} :$*

$(\boldsymbol{F}\hat{\boldsymbol{\beta}})_i \neq 0\}$, *then*

$$nullity(\boldsymbol{F}_{-\mathcal{A}}) = \#fused\ group, \tag{A.14}$$

where

$$\#fused\ group = \sum_{j=1}^{p} \left[\mathbb{1}\{\hat{\beta}_j(1) \neq 0\} + \sum_{k=2}^{T} \mathbb{1}\{\hat{\beta}_j(k) \neq \hat{\beta}_j(k-1),\ \hat{\beta}_j(k) \neq 0\} \right]. \tag{A.15}$$

Proof. The right-hand side of (A.15) can be written as

$$\sum_{j=1}^{p} \left[T - \sum_{k=1}^{T} \mathbb{1}\{\hat{\beta}_j(k) = 0\} - \sum_{k=2}^{T} \mathbb{1}\{\hat{\beta}_j(k) = \hat{\beta}_j(k-1),\ \hat{\beta}_j(k) \neq 0\} \right].$$

Since $nullity(\boldsymbol{F}_{-\mathcal{A}}) = pT - rank(\boldsymbol{F}_{-\mathcal{A}})$, we only need to show

$$rank(\boldsymbol{F}_{-\mathcal{A}}) = \sum_{j=1}^{p} \left[\sum_{k=1}^{T} \mathbb{1}\{\hat{\beta}_j(k) = 0\} + \sum_{k=2}^{T} \mathbb{1}\{\hat{\beta}_j(k) = \hat{\beta}_j(k-1),\ \hat{\beta}_j(k) \neq 0\} \right]. \tag{A.16}$$

To calculate the rank of $\boldsymbol{F}_{-\mathcal{A}}$, we count the maximum number of its independent rows.

If $\hat{\beta}_j(k) = 0$, then the $((k-1)p + j)$th row in the lower identity matrix of $\boldsymbol{F}$ is preserved in $\boldsymbol{F}_{-\mathcal{A}}$, which serves as an independent row as this row with only one component 1 can eliminate any other non-zero entries in the same column.

When $\hat{\beta}_j(k) \neq 0$, the corresponding row in the lower identity matrix of $\boldsymbol{F}$ is removed in $\boldsymbol{F}_{-\mathcal{A}}$, which also takes away its only non-zero component 1 in the $((k-1)p + j)$th place (column). Whether there exists any other non-zero entries in the $((k-1)p + j)$th column depends on the relations of the pairs $\left(\hat{\beta}_j(k-1), \hat{\beta}_j(k)\right)$ and $\left(\hat{\beta}_j(k), \hat{\beta}_j(k+1)\right)$. If $\hat{\beta}_j(k) = \hat{\beta}_j(k-1)$, the $((k-2)p + j)$th row in the top difference matrix of $\boldsymbol{F}$ stays in $\boldsymbol{F}_{-\mathcal{A}}$ providing a -1 in the $((k-1)p + j)$th column. We will count this row as an independent row. Because even if $\hat{\beta}_j(k) = \hat{\beta}_j(k+1)$ which keeps the $((k-1)p + j)$th row of $\boldsymbol{F}$ in $\boldsymbol{F}_{-\mathcal{A}}$, this row will be counted as an independent row when we consider the case of $\hat{\beta}_j(k+1)$ where $\hat{\beta}_j(k+1) \neq 0$ and $\hat{\beta}_j(k+1) = \hat{\beta}_j(k)$.

Hence, the maximum number of the independent rows in $\boldsymbol{F}_{-\mathcal{A}}$ is

$$\sum_{i=1}^{p} \left[\sum_{k=1}^{T} \mathbb{1}\{\hat{\beta}_j(k) = 0\} + \sum_{k=2}^{T} \mathbb{1}\{\hat{\beta}_j(k) = \hat{\beta}_j(k-1),\ \beta_j(k) \neq 0\} \right].$$

Therefore, (A.16) holds. Proof is completed. □

A.4 Inverse of 2-by-2 Block Diagonal Matrix

In this section, we prove that for a $p-$by$-p$ covariance matrix Σ_s where p is a even number, if it takes the form of a 2×2 block matrix, $\Sigma_s = (A, C; C, B)$, where the four block submatrices are all diagonal, then the nonzero entries in its inverse Σ_s^{-1} locate in the same position as the original matrix Σ_s.

Proof. $\begin{bmatrix} A_{p/2\times p/2} & C_{p/2\times p/2} \\ C_{p/2\times p/2} & B_{p/2\times p/2} \end{bmatrix}_{p\times p}^{-1} = \begin{bmatrix} (A - CB^{-1}C)^{-1} & -A^{-1}C(B - CA^{-1}C)^{-1} \\ -B^{-1}C(A - CB^{-1}C)^{-1} & (B - CA^{-1}C)^{-1} \end{bmatrix}.$
If submatrices A, B and C are diagonal matrices, then the four submatrices in the inverse are also diagonal matrices. □

A.5 Additional Results in Simulation 1

A.5.1 Partial Correlation Networks

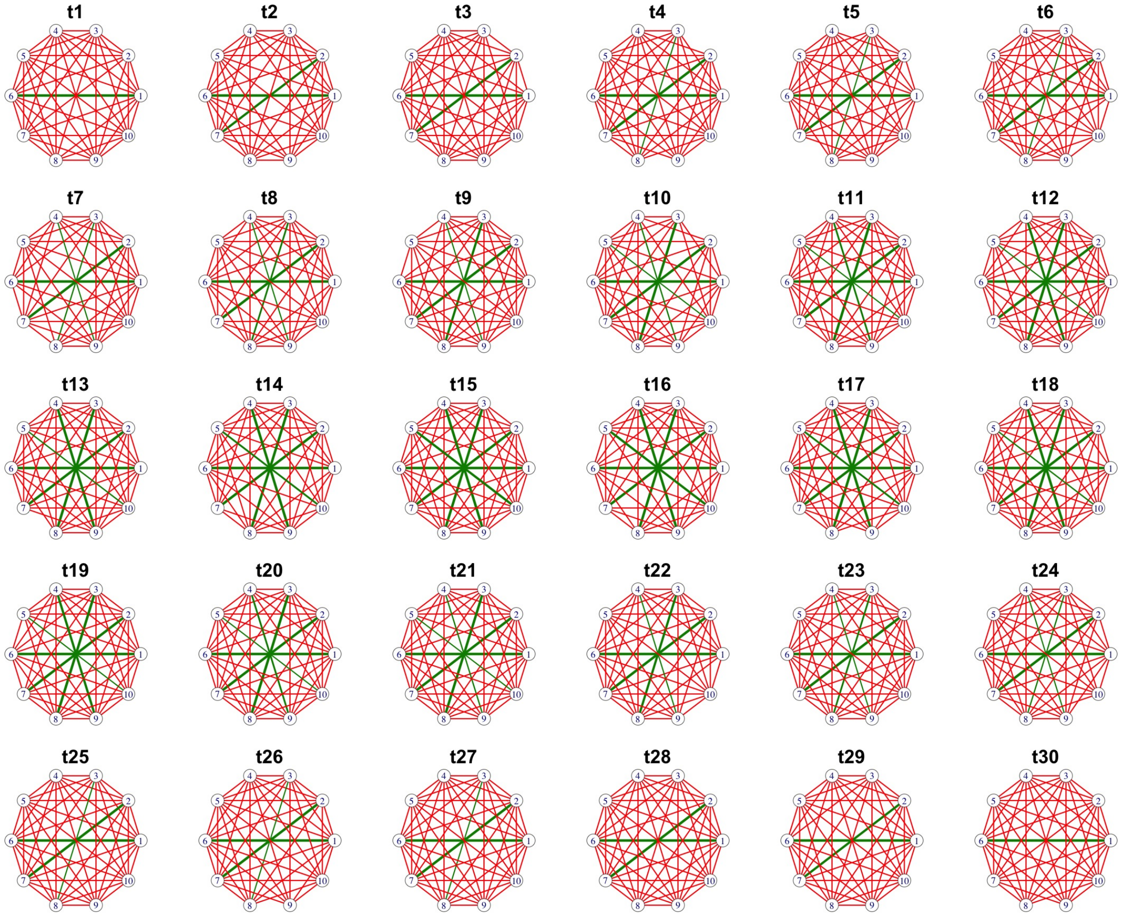


Figure A.1: The GEN-based partial correlation networks at different time points with sample size 200. The green solid lines, the green dashed lines and the red solid lines represent the true positive, false negative and false positive connections, respectively. Thickness of each green solid line represents the magnitude of its underlying true partial correlation.

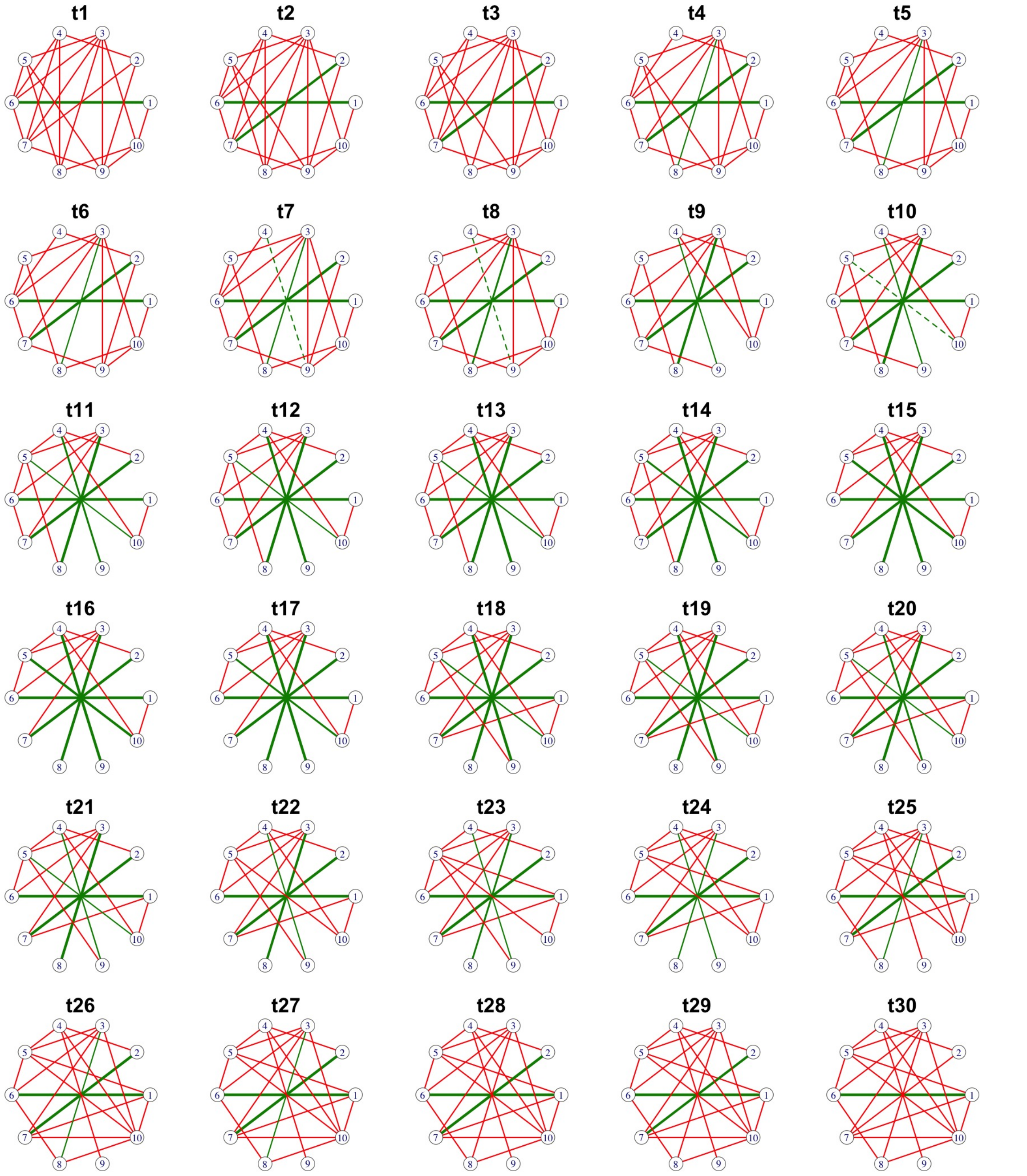
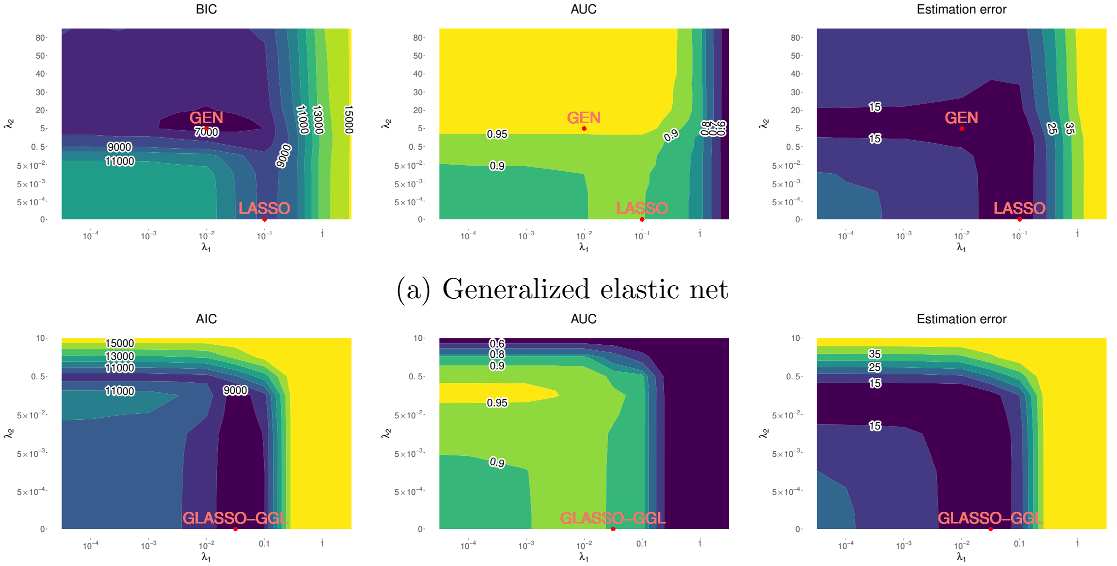


Figure A.2: The GFL-based partial correlation networks at different time points with sample size 200. The green solid lines, the green dashed lines and the red solid lines represent the true positive, false negative and false positive connections, respectively. Thickness of each green solid line represents the magnitude of its underlying true partial correlation.

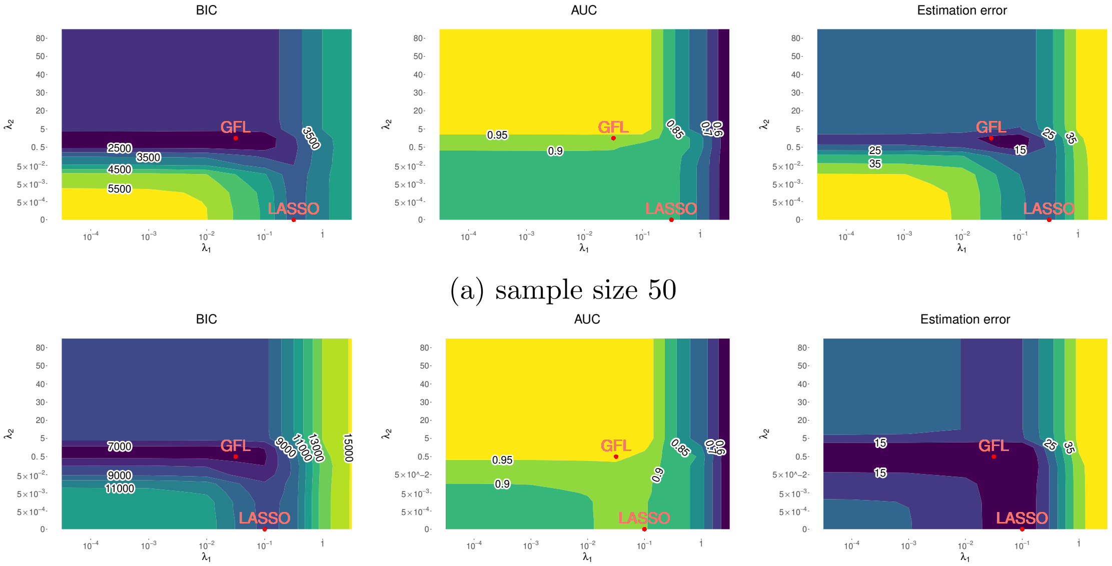
A.5.2 Contours



(a) Generalized elastic net

(b) GLASSO-GGL

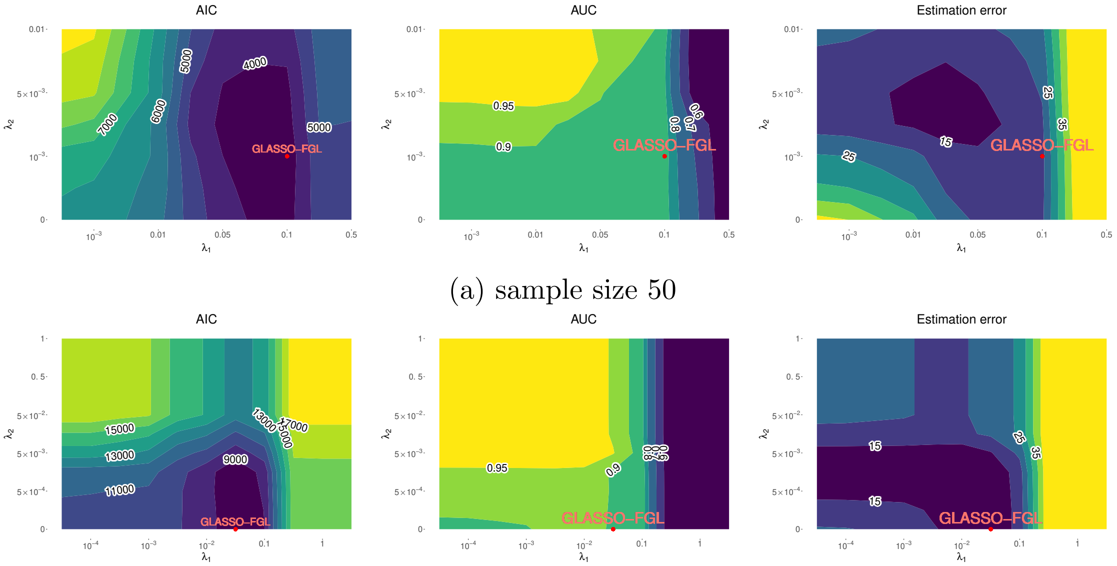
Figure A.3: Simulation performance measurements from generalized elastic net and GLASSO-GGL both with sample size 200. Darker areas represent lower values. For reference, in the simulation from generalized elastic net, the sample AUC is 0.8956, comparing to the GEN AUC 0.9764 in the figure; the sample estimation error is 20.31, comparing to the estimation error of GEN as 12.52 in the figure. In the simulation from GLASSO-GGL, the GLASSO-GGL AUC is 0.8967 in the figure and the GLASSO-GGL estimation error is 10.17 in the figure.



(a) sample size 50

(b) sample size 200

Figure A.4: Simulation performance measurements from generalized fused lasso. Darker areas represent lower values. For reference, in the simulation with sample size 50, the sample AUC is 0.8757, comparing to the GFL AUC 0.9242 in the figure; the sample estimation error is 41.81, comparing to the GFL estimation error as 13.36 in the figure. In the simulation with sample size 200, the sample AUC is 0.8956, comparing to the GFL AUC 0.9692 in the figure; the sample estimation error is 20.31, comparing to the GFL estimation error as 6.5741 in the figure.



(a) sample size 50

(b) sample size 200

Figure A.5: Simulation performance measurements from GLASSO-FGl. Darker areas represent lower values. For reference, in the simulation with sample size 50, the sample AUC is 0.8757, comparing to the GLASSO-FGl AUC 0.8733 in the figure; the sample estimation error is 41.81, comparing to the GLASSO-FGL estimation error as 18.54 in the figure. In the simulation with sample size 200, the sample AUC is 0.8956, comparing to the GLASSO-FGL AUC 0.8967 in the figure; the sample estimation error is 20.31, comparing to the GLASSO-FGL estimation error as 10.17 in the figure.

A.6 Simulation 2 on Complicated Networks

In addition to the simulation in the main text, we perform additional simulation studies, which we call "Simulation 2", to assess the performance of our proposed approaches. We generate data where the network size $p = 50$, the number of time points is 20, and the topological structures are more complicated.

A.6.1 Data Generation

We simulate the network in Simulation 2 with two clusters, because many real life networks have community structures. There are 30 and 20 nodes in the two clusters respectively, and the probability that there ever exists a connection between two nodes is 0.1 and 0.01 for nodes from the same and different clusters respectively. We follow Peng et al. (2009) to generate temporal precision matrices directly, whose non-vanishing entries are functions of time. We first define the initial matrix $\boldsymbol{A}(t)$ as a symmetric matrix whose diagonal entries are one. If there ever exists an edge between node i and j, then $\boldsymbol{A}[i,j](t)$ takes values of $\pm f(t)$, otherwise 0. Let $f(t)$ be zero everywhere except on its active interval $[T_s, T_e]$ which is randomly located in a continuous period of time between $[0,1]$, where

$$f(t, T_s, T_e) = \frac{1}{2}\left[0.1 + 0.8 \cdot \sin\left(\frac{t - T_s}{T_e - T_s}\pi\right)\right].$$

The off-diagonal entry in $\boldsymbol{A}(t)$ is then divided by 1.5 times the total absolute value of the entries in its row to guarantee the positive definiteness. We take the average of $\boldsymbol{A}(t)$ and its transpose $\boldsymbol{A}(t)^\top$ to assure the symmetry, and then the covariance matrix $\Sigma(t)$ is given by $\boldsymbol{\Sigma}(t)[i,j] = \boldsymbol{A}(t)^{-1}[i,j]/\sqrt{\boldsymbol{A}(t)^{-1}[i,i] \cdot \boldsymbol{A}(t)^{-1}[j,j]}$, from which the concentration matrices and the partial correlation matrices can be calculated. It should be noted that using this more conventional method can only ensure the underlying partial correlations are smooth functions of time, while the actual observations at different time points are independent and not necessarily smooth over time. In this scenario, there are 79 true connections as displayed in Figure A.6. Moreover, Figure S6 shows that there exist two clusters where nodes within the same cluster are considerably more densely connected that those from different clusters. This network structure is thus more complicated than the one we simulated in the main text.

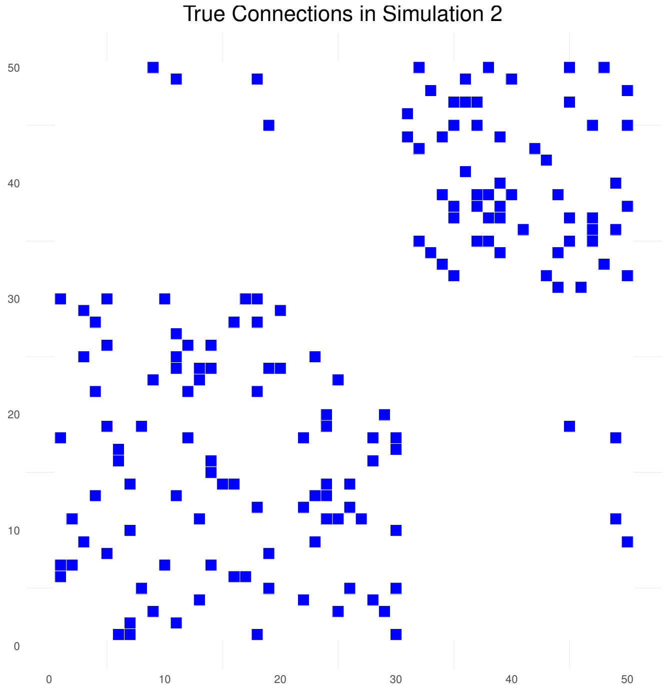


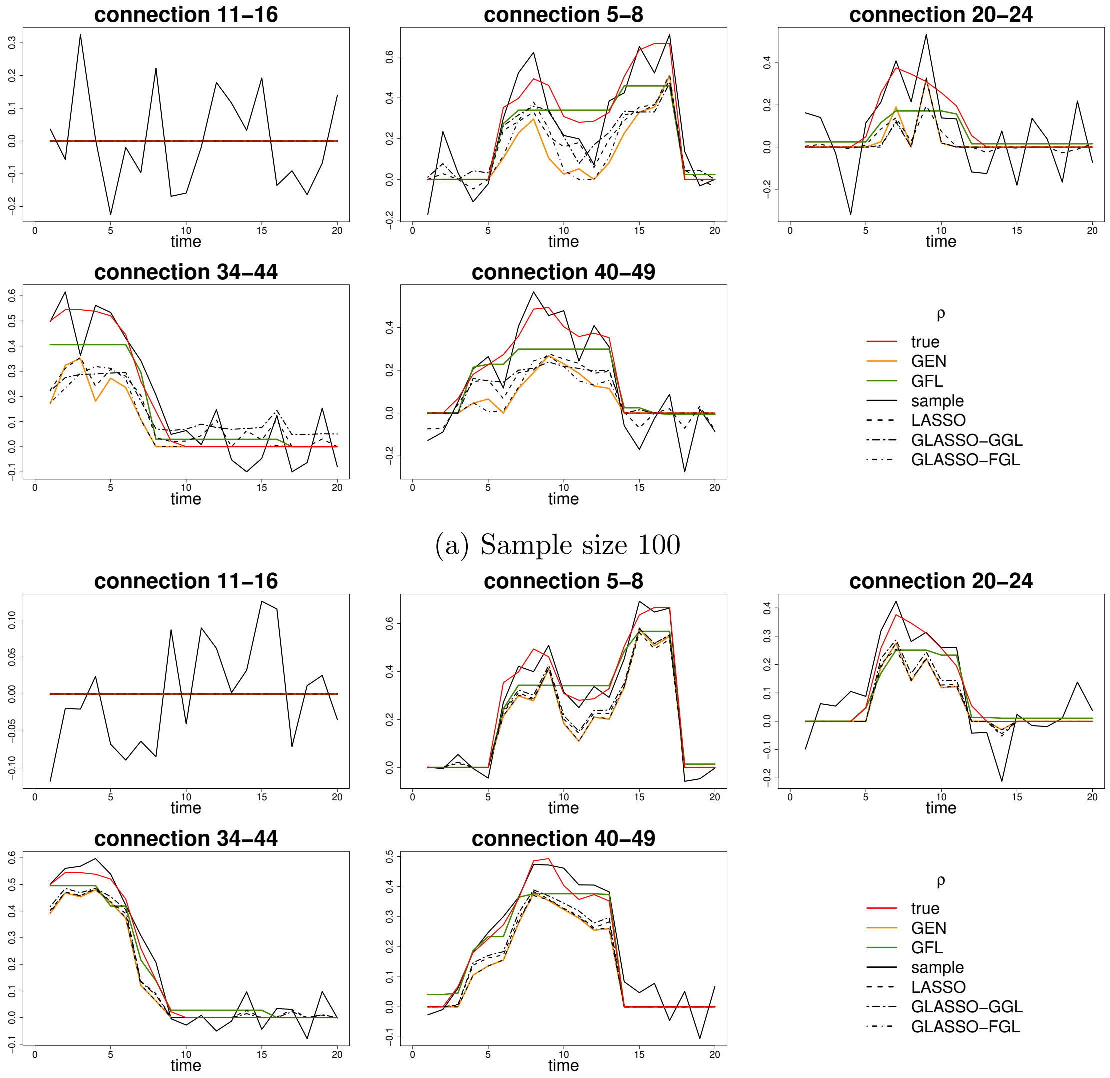
Figure A.6: One snapshot of true connections in the time-varying networks in Simulation 2

A.6.2 Results

We consider $n = 100$ and $n = 300$, and each simulation is repeated 100 times. The estimation error and AUC are presented in Table A.1. To further illustrate the performance of our methods, we plot the profiles of some connections from one single trial in Figure A.7. One of our two methods, GFL, performs well, while the other method, GEN, does not perform well. The reason why GEN is inferior in this scenario might be that the l_2 penalty shrinks the difference of the neighboring coefficients towards zero in a smoother manner. In contrast, the l_1 penalty in GFL encourages zero coefficients and a few non-zero coefficients with jumps or breaks in between; this feature makes GFL preferable for less smooth signals.

Method	Estimation error		AUC	
	$n = 100$	$n = 300$	$n = 100$	$n = 300$
Sample	137.9 (0.19)	62.3 (0.05)	0.8300 (0)	0.9420 (0)
LASSO	36.0 (0.04)	20.0 (0.11)	0.8154 (0)	0.9303 (0)
TS-LASSO	36.0 (0.04)	20.0 (0.03)	0.8162 (0)	0.9293 (0)
GLASSO-GGL	30.1 (0.04)	18.7 (0.03)	0.9526 (0)	0.9517 (0)
GLASSO-FGL	30.7 (0.04)	17.5 (0.03)	0.9238 (0)	0.9466 (0)
GEN	36.4 (0.04)	20.4 (0.03)	0.8162 (0)	0.9297 (0)
GEN-EBIC	47.6 (0.04)	20.4 (0.03)	0.7081 (0)	0.9297 (0)
GFL	20.8 (0.04)	12.2 (0.02)	0.9730 (0)	0.9858 (0)
GFL-EBIC	22.9 (0.04)	17.6 (0.03)	0.9564 (0)	0.9911 (0)

Table A.1: Mean estimation error and AUCs across 100 replicates with standard error in parentheses.



(a) Sample size 100

(b) Sample size 300

Figure A.7: Estimated partial correlations of some selected connections.

A.7 Computational Time

In this section, we present the computational times, averaged over 100 simulation replicates, respectively for each method in both simulation studies (see Tables A.2 and A.3 below).

All experiments were run on an Intel Xeon E5-4650 2.2 GHz processor.

Our two methods, GEN and GFL, are generally quite efficient. For other methods, simply computing the sample covariance matrix is—not surprisingly—the fastest, while the naïve LASSO and GLASSO-FGL typically take more time. However, such comparisons are not truly meaningful, as these others methods were *not* originally meant for time-varying networks.

Readers may also notice that, in simulation 1, the naïve LASSO and our methods are actually slightly faster when $n = 200$ than they are when $n = 50$. This is because, when the total number of partial correlations to be estimated is much larger than the sample size (here, $Tp(p-1)/2 > n$), the estimation problem becomes easier—and the algorithms can require fewer iterations to converge—as more samples become available, and this sometimes more than compensates for the longer running times of each iteration with a larger n.

Method	Time (in seconds)	
	$n = 50$	$n = 200$
Sample	0.01 (0)	0.01 (0)
LASSO	98.68 (4.61)	72.30 (0.60)
TS-LASSO	9.10 (0.04)	10.58 (0.03)
GLASSO-GGL	3.02 (0.04)	3.84 (0.03)
GLASSO-FGL	>1hour	>1hour
GEN	1.32 (0.10)	0.94 (0.02)
GEN-EBIC	1.17 (0.07)	1.09 (0.02)
GFL	5.64 (0.41)	2.70 (0.03)
GFL-EBIC	5.66 (0.53)	2.40 (0.03)

Table A.2: Running time across 100 replicates in Simulation 1 with standard error in parentheses.

Method	Time (in minutes)	
	$n = 100$	$n = 300$
Sample	0.001 (0)	0.001 (0)
LASSO	> 50 mins	>1 hour
TS-LASSO	>1 hour	>1 hour
GLASSO-GGL	0.022 (0)	0.028 (0)
GLASSO-FGL	>1 hour	>1.5 hour
GEN	3.771 (0.044)	6.823 (0.071)
GEN-EBIC	1.877 (0.024)	6.823 (0.071)
GFL	2.808 (0.291)	3.782 (0.032)
GFL-EBIC	2.544 (0.029)	4.156 (0.026)

Table A.3: Running time across 100 replicates in Simulation 2 with standard error in parentheses.

A.8 Additional Results from Canadian Weather Data

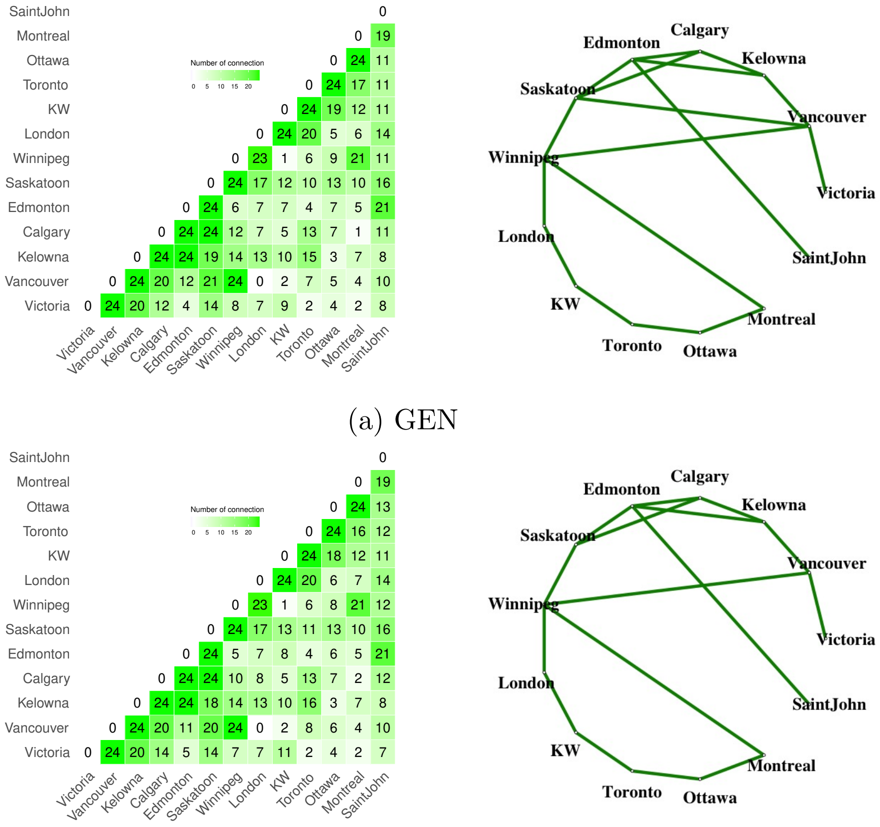
A.8.1 Result 2 from GEN and GFL

This subsection summarizes “Result 2” mentioned in the main manuscript as a comparison to “Result 1”.

		Result 1	Result 2
Generalized elastic net	degrees of freedom	886.95	925.71
	# of connections	888	979

		Result 1	Result 2
Generalized fused LASSO	degrees of freedom	912	867
	# of connections	931	987

Table A.4: Summary of degrees of freedom and number of detected edges based on the GEN and the GFL with two pairs of selected tuning parameters.



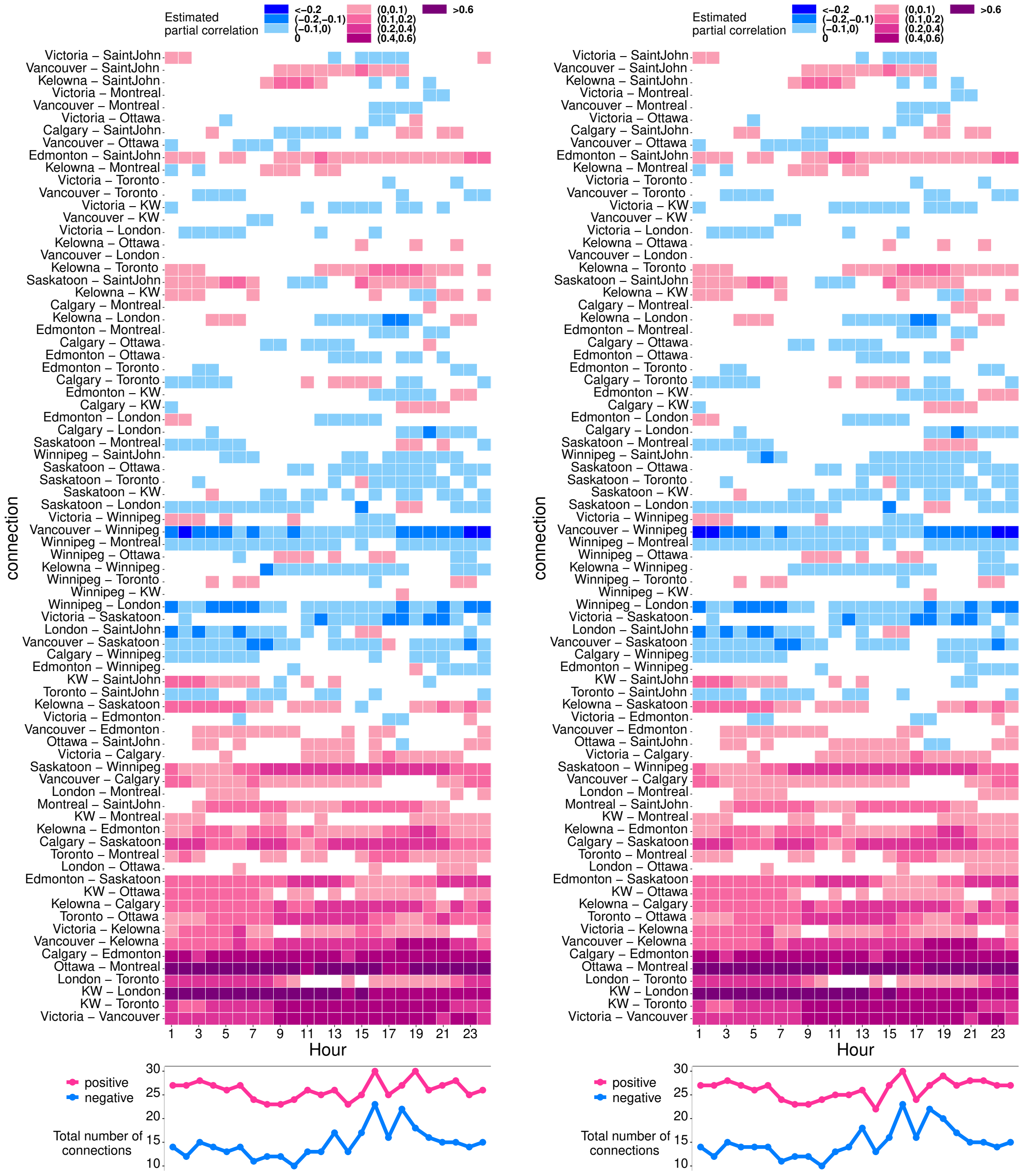
	Victoria	Vancouver	Kelowna	Calgary	Edmonton	Saskatoon	Winnipeg	London	KW	Toronto	Ottawa	Montreal	SaintJohn
SaintJohn													0
Montreal												0	19
Ottawa											0	24	11
Toronto										0	24	17	11
KW									0	24	19	12	11
London								0	24	20	5	6	14
Winnipeg							0	23	1	6	9	21	11
Saskatoon						0	24	17	12	10	13	10	16
Edmonton					0	24	6	7	7	4	7	5	21
Calgary				0	24	24	12	7	5	13	7	1	11
Kelowna			0	24	24	19	14	13	10	15	3	7	8
Vancouver		0	24	20	12	21	24	0	2	7	5	4	10
Victoria	0	24	20	12	4	14	8	7	9	2	4	2	8

(a) GEN

	Victoria	Vancouver	Kelowna	Calgary	Edmonton	Saskatoon	Winnipeg	London	KW	Toronto	Ottawa	Montreal	SaintJohn
SaintJohn													0
Montreal												0	19
Ottawa											0	24	13
Toronto										0	24	16	12
KW									0	24	18	12	11
London								0	24	20	6	7	14
Winnipeg							0	23	1	6	8	21	12
Saskatoon						0	24	17	13	11	13	10	16
Edmonton					0	24	5	7	8	4	6	5	21
Calgary				0	24	24	10	8	5	13	7	2	12
Kelowna			0	24	24	18	14	13	10	16	3	7	8
Vancouver		0	24	20	11	20	24	0	2	8	6	4	10
Victoria	0	24	20	14	5	14	7	7	11	2	4	2	7

(b) GFL

Figure A.8: [Result 2] Aggregated connections between different cities over 24 time points.



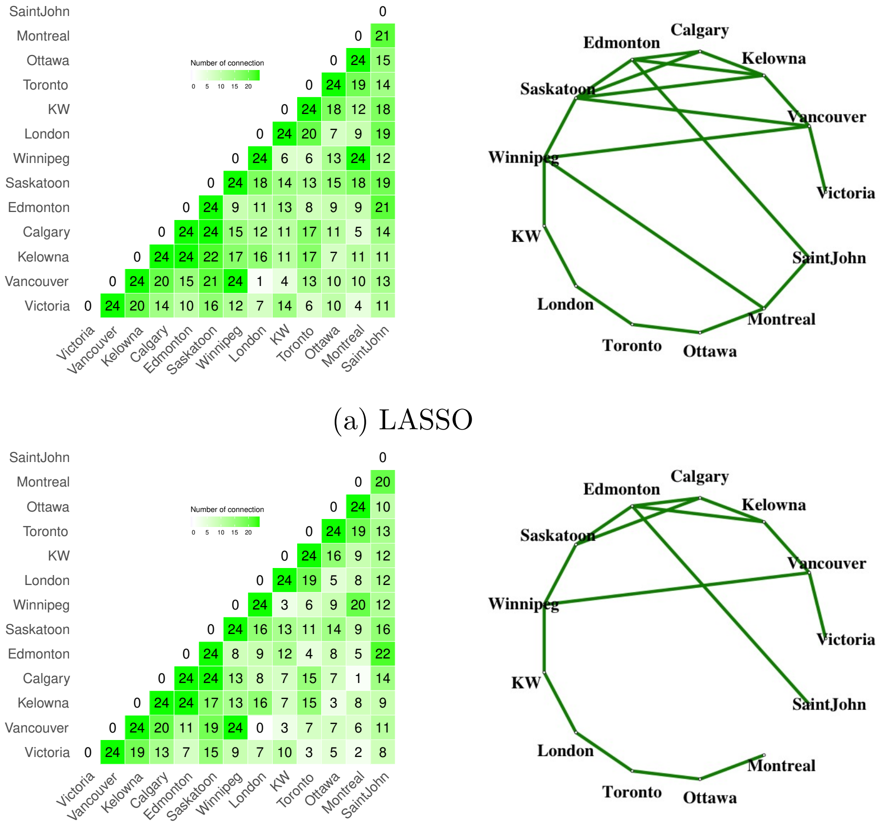
(a) GEN

(b) GFL

Figure A.9: [Result 2] Estimated partial correlations between different cities over 24 hours. Connections are listed in descending order of the distances between cities from top to the bottom. Each cell corresponds to a connection at a given time point, and the color characterizes the value of the estimated partial correlation. The lower line chart marginalizes the total connections at each hour.

A.8.2 Results from LASSO and TS-LASSO

As a baseline benchmark to compare with, we hereby include results from naïve LASSO and TS-LASSO. Looking at the results shown in Figures A.10 and A.11, we see that the two LASSO methods and our proposed methods are similar in terms of the overall network structure and aggregated number of connections. This is not surprising; different methods are definitely supposed to pick up the same information from the underlying data. But there exist some distinct differences in the estimated partial corrleations as well. In particular, compared with Figure 5 in the main text, Figure A.11 indicates that the trajectories of the estimated partial correlations from naïve LASSO and TS-LASSO are more wiggly, with more non-zero ones "popping up" at different time points. Noticeably for the TS-LASSO, there are a large number of "stand-alone" connections showing up 11:00. This further justifies our claim that smoother partial correlations ensue from incorporating $P(\boldsymbol{\theta})$ in Equation (4). Therefore, this penalty is desirable to achieve a smooth time-varying network.



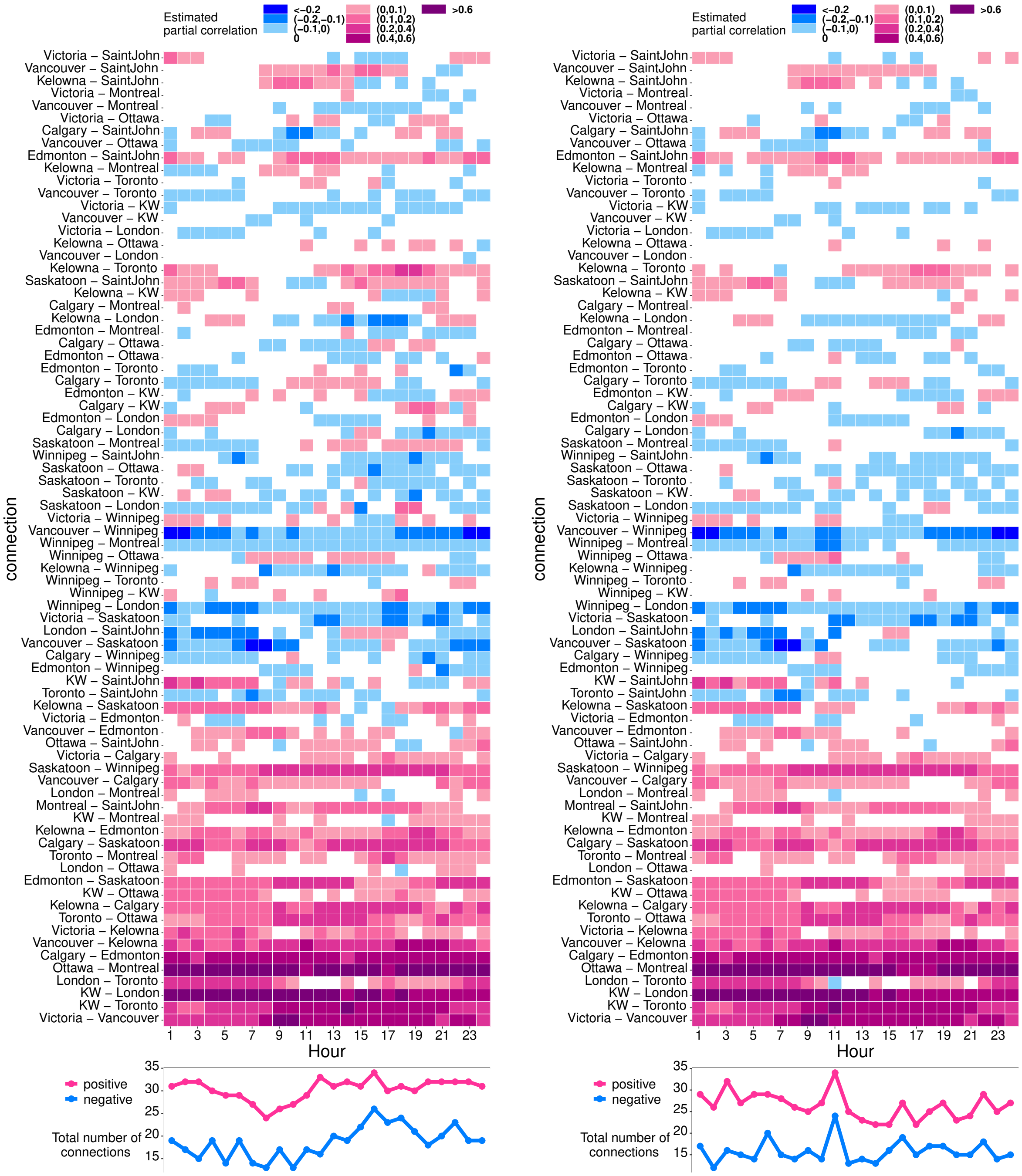
	Victoria	Vancouver	Kelowna	Calgary	Edmonton	Saskatoon	Winnipeg	London	KW	Toronto	Ottawa	Montreal	SaintJohn
SaintJohn													0
Montreal												0	21
Ottawa											0	24	15
Toronto										0	24	19	14
KW									0	24	18	12	18
London								0	24	20	7	9	19
Winnipeg							0	24	6	6	13	24	12
Saskatoon						0	24	18	14	13	15	18	19
Edmonton					0	24	9	11	13	8	9	9	21
Calgary				0	24	24	15	12	11	17	11	5	14
Kelowna			0	24	24	22	17	16	11	17	7	11	11
Vancouver		0	24	20	15	21	24	1	4	13	10	10	13
Victoria	0	24	20	14	10	16	12	7	14	6	10	4	11

(a) LASSO

	Victoria	Vancouver	Kelowna	Calgary	Edmonton	Saskatoon	Winnipeg	London	KW	Toronto	Ottawa	Montreal	SaintJohn
SaintJohn													0
Montreal												0	20
Ottawa											0	24	10
Toronto										0	24	19	13
KW									0	24	16	9	12
London								0	24	19	5	8	12
Winnipeg							0	24	3	6	9	20	12
Saskatoon						0	24	16	13	11	14	9	16
Edmonton					0	24	8	9	12	4	8	5	22
Calgary				0	24	24	13	8	7	15	7	1	14
Kelowna			0	24	24	17	13	16	7	15	3	8	9
Vancouver		0	24	20	11	19	24	0	3	7	7	6	11
Victoria	0	24	19	13	7	15	9	7	10	3	5	2	8

(b) TS-LASSO

Figure A.10: Aggregated connections between different cities over 24 time points.



(a) LASSO

(b) TS-LASSO

Figure A.11: Estimated partial correlations between different cities over 24 hours. Connections are listed in descending order of the distances between cities from the top to the bottom. Each cell corresponds to a connection at a given time point, and the color characterizes the value of the estimated partial correlation. The lower line chart marginalizes the total connections at each hour.

A.8.3 Residual Analysis

In this section, we assess the residuals obtained from fitting the Canadian weather data with our proposed methods, GEN and GFL. More specifically, for each city j at a given hour t, we (i) plot the residual $e_{ij}(t) = X_{ij}(t) - \hat{X}_{ij}(t)$ against the index i, (ii) compute the Durbin-Watson (DW) statistic, and (iii) inspect the autocorrelation function (ACF) of these residuals.

The DW statistic is calculated as $d_j(t) = \sum_{i=2}^{124} (e_{ij}(t) - e_{i-1,j}(t))^2 / \sum_{i=1}^{124} e_{ij}(t)^2$. A value of $d = 2$ indicates no autocorrelation among the residuals, while $d \in [1.5, 2.5]$ is regarded as evidence for weak dependence at best. For the ACF of the residuals, values within the range of $[-0.2, 0.2]$ and close to zero are generally regarded to suggest weak or no linear correlation between observations at different time lags.

Here in Figures A.12, A.13, A.14 and A.15 below, we only include a representative selection of residual assessments for all thirteen cities at $t = 1{:}00$. The complete set of results for other 23 time points can be found at https://github.com/JieJJian/TimeVaryingGGM.

Generally speaking, the residual plot for each city at a given time point show a random pattern with no obvious trends or patterns, and most Durbin-Watson statistics are between 1.5 and 2.5, both of which indicate that any serial correlation that may exist in principle is relatively weak. For instance, in Figures A.12 and A.14 here, the only city with a DW-statistic < 1.5 is Vancouver. Likewise, Figures A.13 and A.15 show that the majority of autocorrelations are within ±0.2, indicating weak or little linear relationships between residuals at different time lags. The one exception is again Vancouver, with a lag-1 autocorrelation exceeding 0.2.

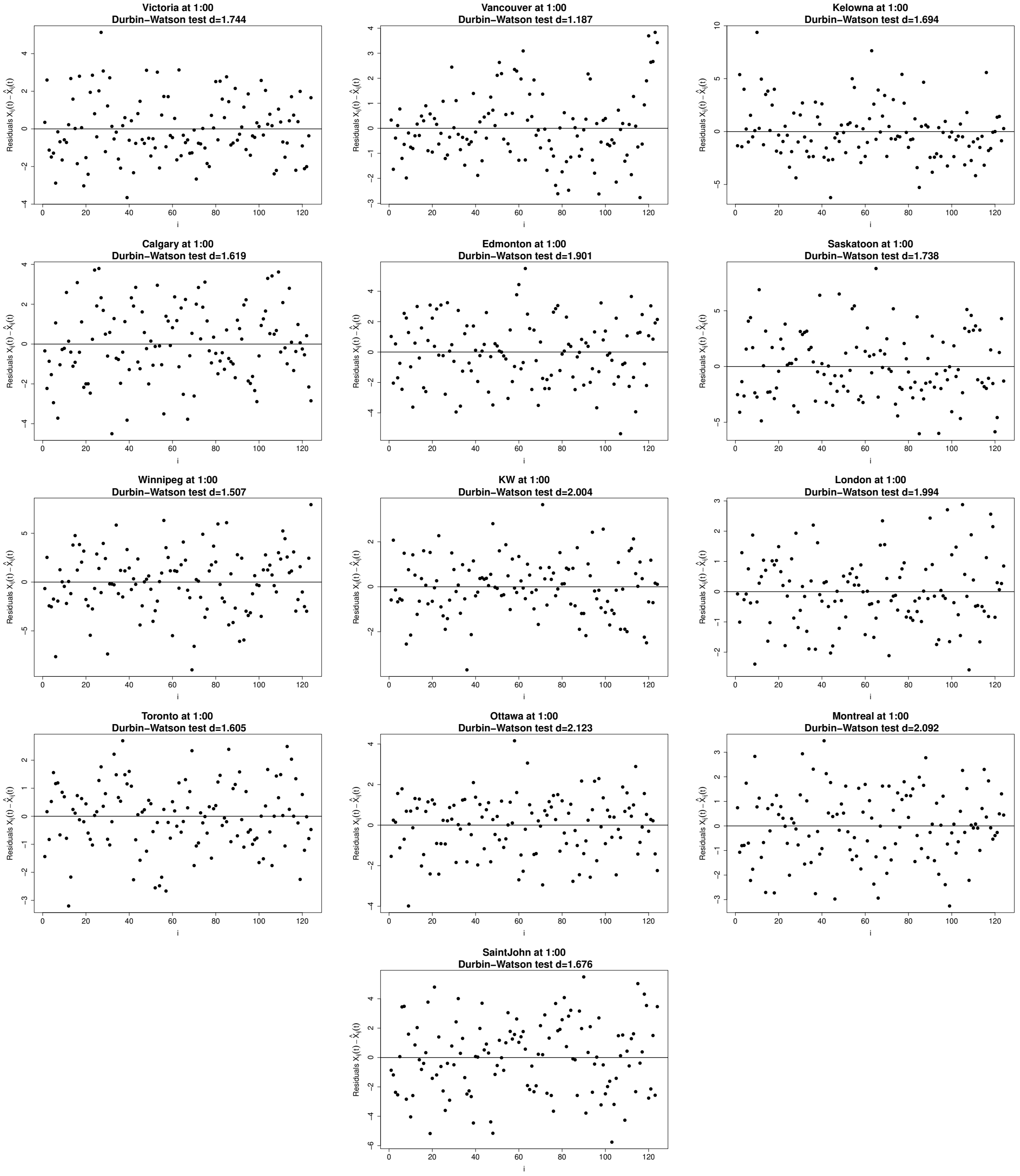


Figure A.12: [GEN, Result 1] Residual plots of each city at 1:00.

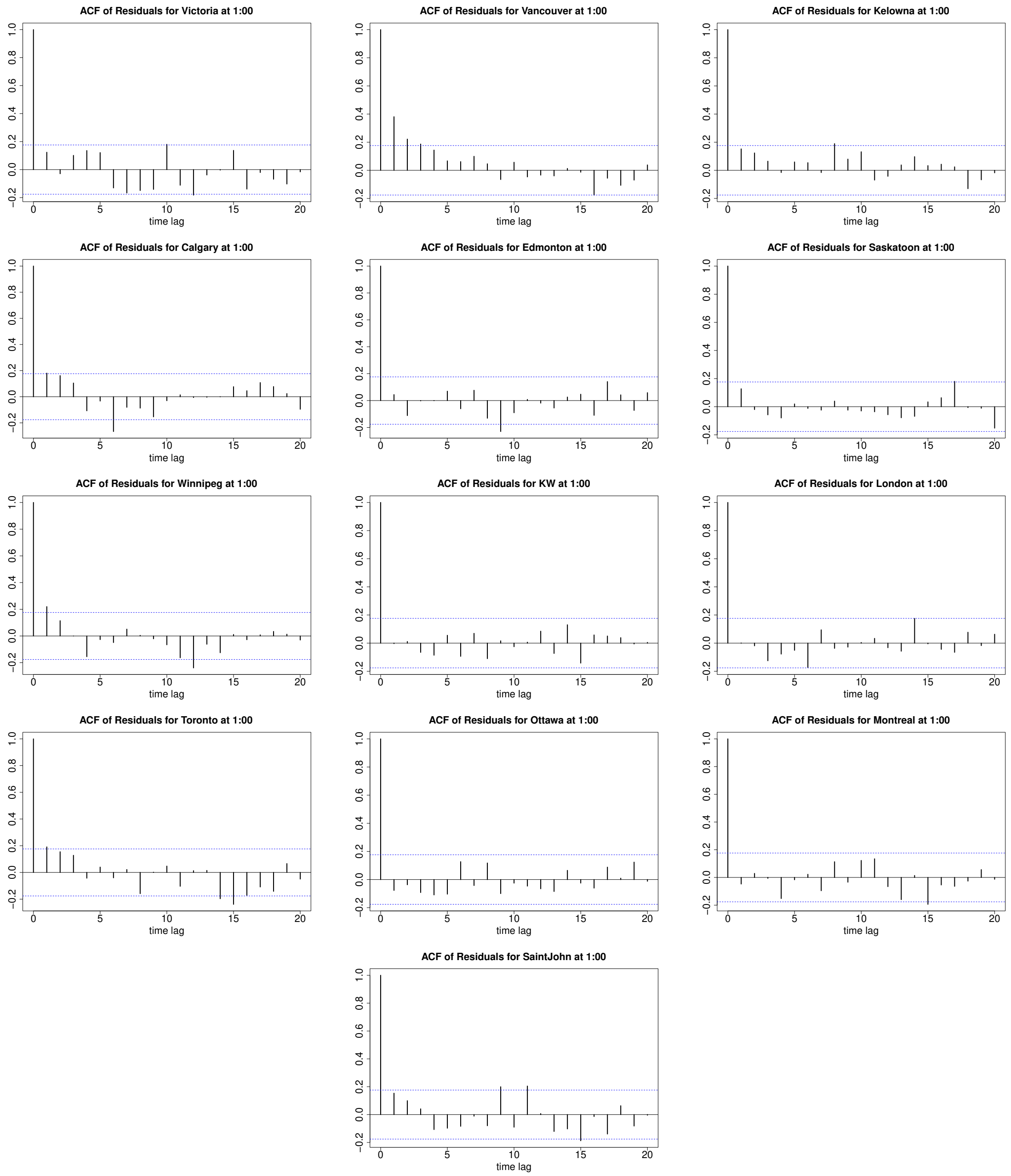


Figure A.13: [GEN, Result 1] Autocorrelation of residual for each city at 1:00.

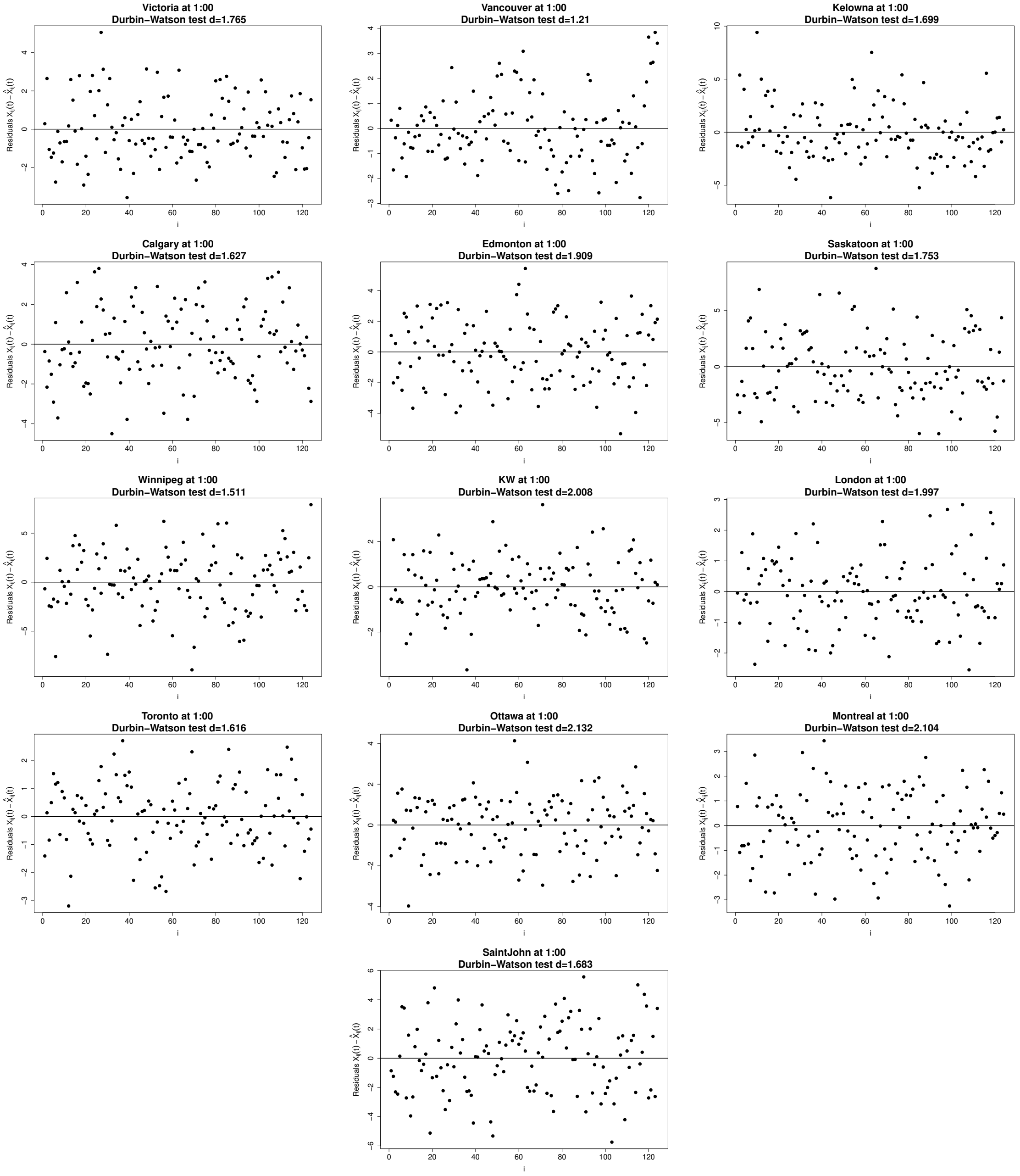


Figure A.14: [GFL, Result 1] Residual plots of each city at 1:00.

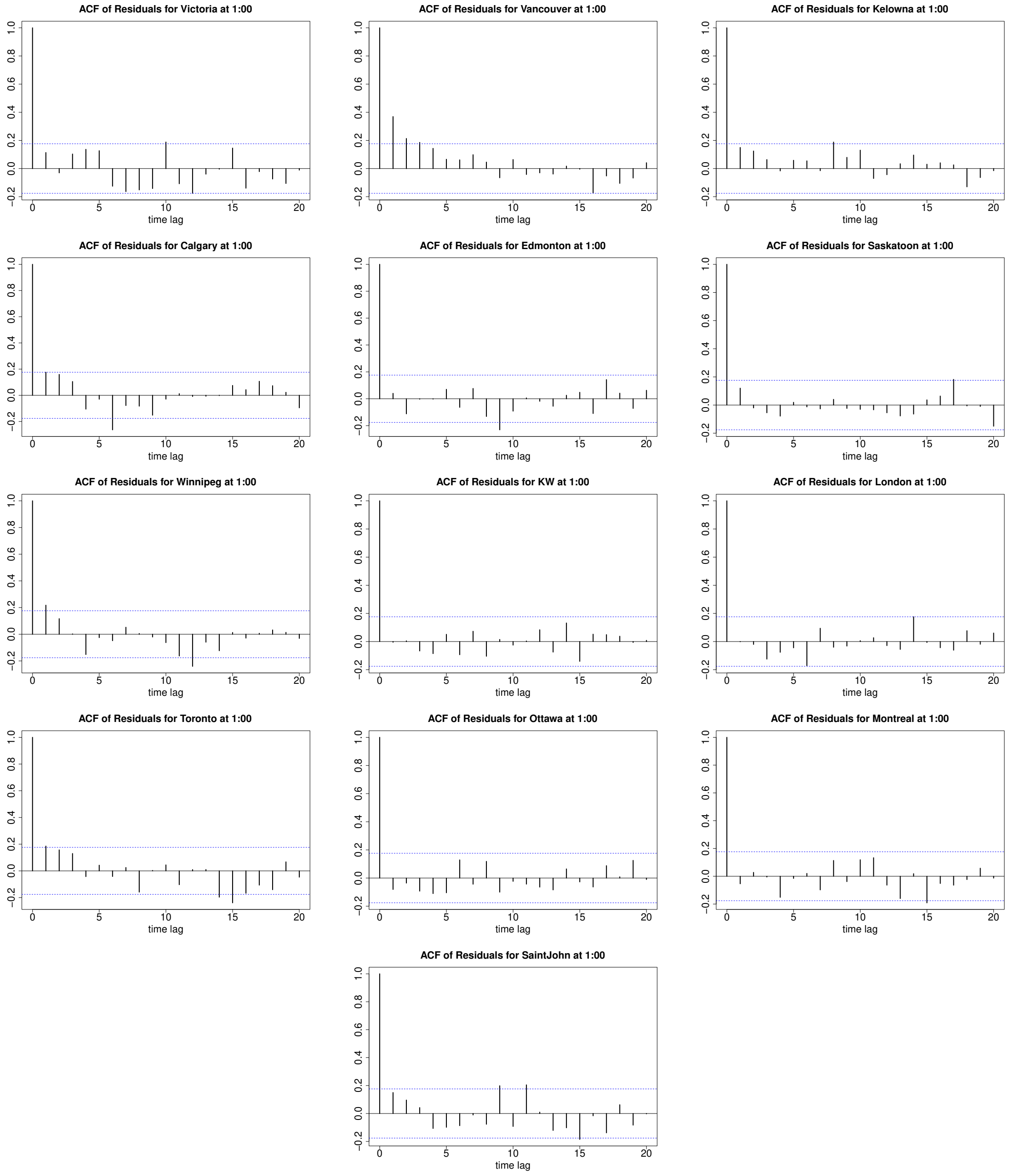
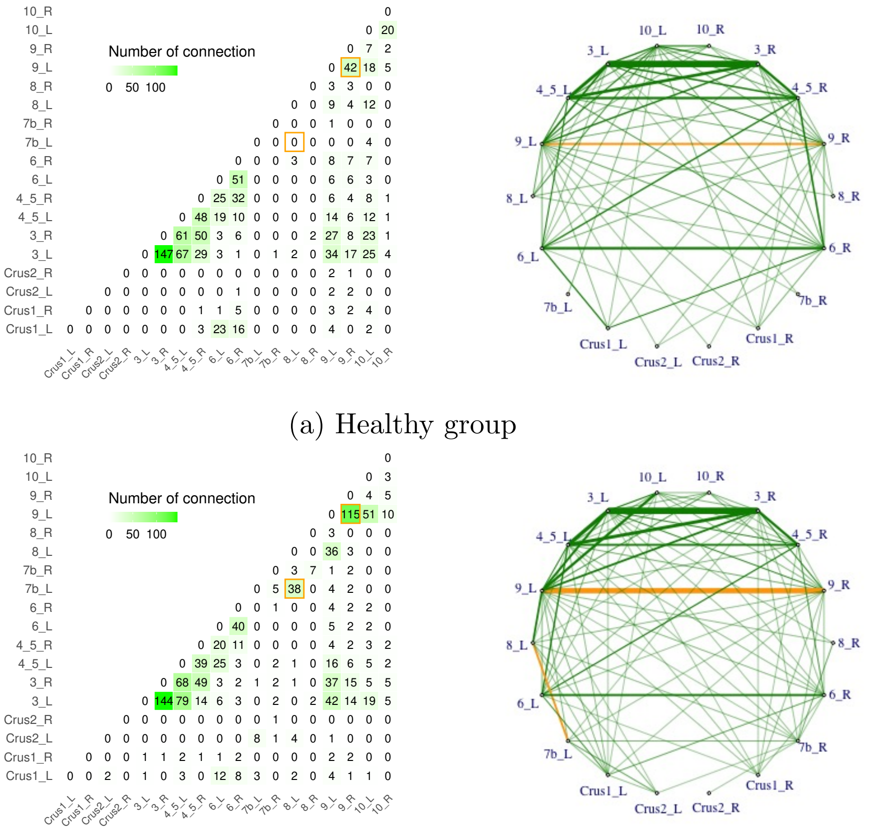


Figure A.15: [GFL, Result 1] Autocorrelation of residual for each city at 1:00.

A.9 Additional Results from ADHD Data



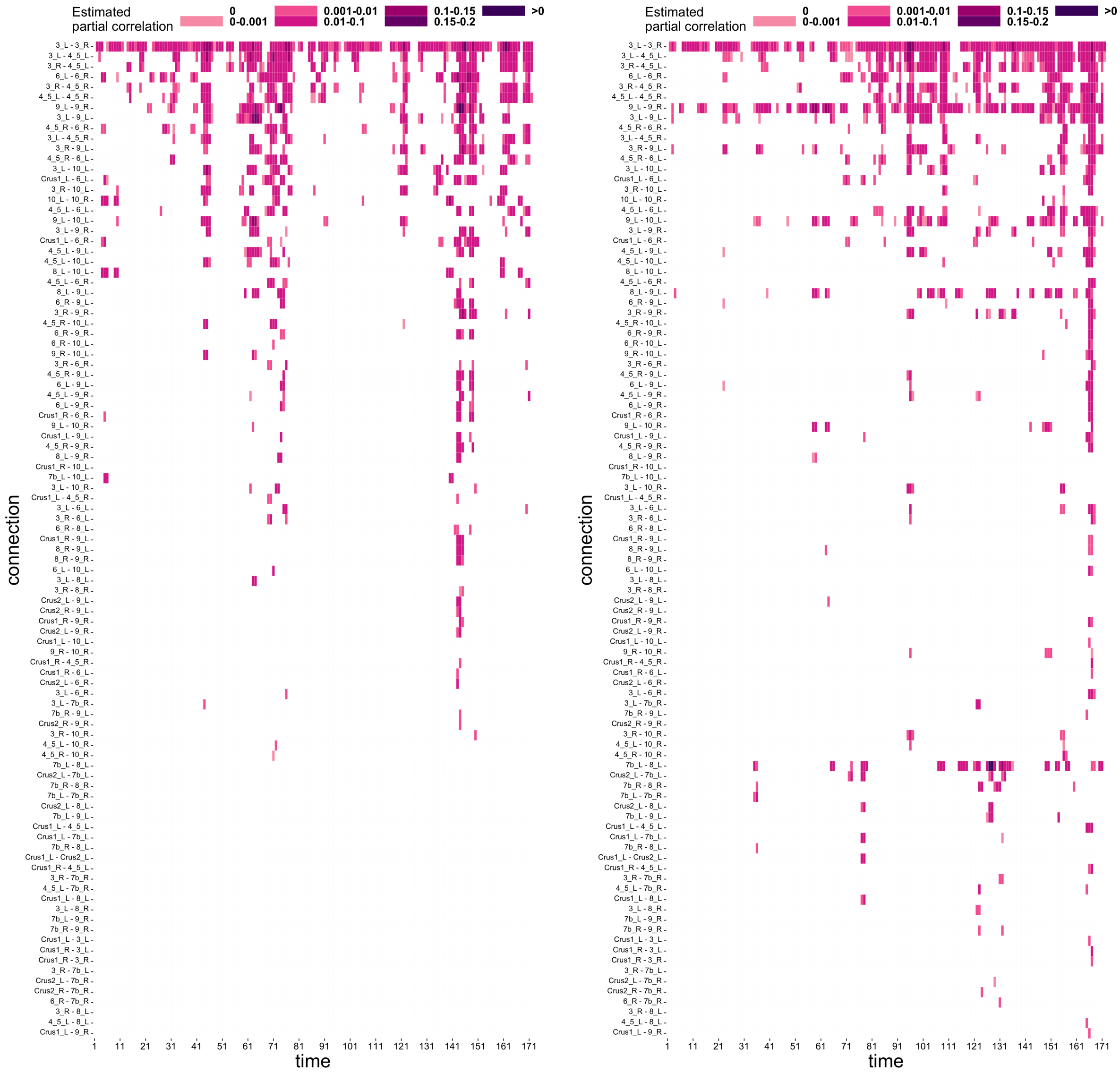
	Crus1_L	Crus1_R	Crus2_L	Crus2_R	3_L	3_R	4_5_L	4_5_R	6_L	6_R	7b_L	7b_R	8_L	8_R	9_L	9_R	10_L	10_R
10_R																		0
10_L																	0	20
9_R																0	7	2
9_L															0	42	18	5
8_R														0	3	3	0	0
8_L													0	0	9	4	12	0
7b_R												0	0	0	1	0	0	0
7b_L											0	0	0	0	0	0	4	0
6_R										0	0	0	3	0	8	7	7	0
6_L									0	51	0	0	0	0	6	6	3	0
4_5_R								0	25	32	0	0	0	0	6	4	8	1
4_5_L							0	48	19	10	0	0	0	0	14	6	12	1
3_R						0	61	50	3	6	0	0	0	2	27	8	23	1
3_L					0	147	67	29	3	1	0	1	2	0	34	17	25	4
Crus2_R				0	0	0	0	0	0	0	0	0	0	0	2	1	0	0
Crus2_L			0	0	0	0	0	0	0	1	0	0	0	0	2	2	0	0
Crus1_R		0	0	0	0	0	0	1	1	5	0	0	0	0	3	2	4	0
Crus1_L	0	0	0	0	0	0	0	3	23	16	0	0	0	0	4	0	2	0

(a) Healthy group

	Crus1_L	Crus1_R	Crus2_L	Crus2_R	3_L	3_R	4_5_L	4_5_R	6_L	6_R	7b_L	7b_R	8_L	8_R	9_L	9_R	10_L	10_R
10_R																		0
10_L																	0	3
9_R																0	4	5
9_L															0	115	51	10
8_R														0	3	0	0	0
8_L													0	0	36	3	0	0
7b_R												0	3	7	1	2	0	0
7b_L											0	5	38	0	4	2	0	0
6_R										0	0	1	0	0	4	2	2	0
6_L									0	40	0	0	0	0	5	2	2	0
4_5_R								0	20	11	0	0	0	0	4	2	3	2
4_5_L							0	39	25	3	0	2	1	0	16	6	5	2
3_R						0	68	49	3	2	1	2	1	0	37	15	5	5
3_L					0	144	79	14	6	3	0	2	0	2	42	14	19	5
Crus2_R				0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Crus2_L			0	0	0	0	0	0	0	0	8	1	4	0	1	0	0	0
Crus1_R		0	0	0	1	1	2	1	1	2	0	0	0	0	2	2	0	0
Crus1_L	0	0	2	0	1	0	3	0	12	8	3	0	2	0	4	1	1	0

(b) ADHD group

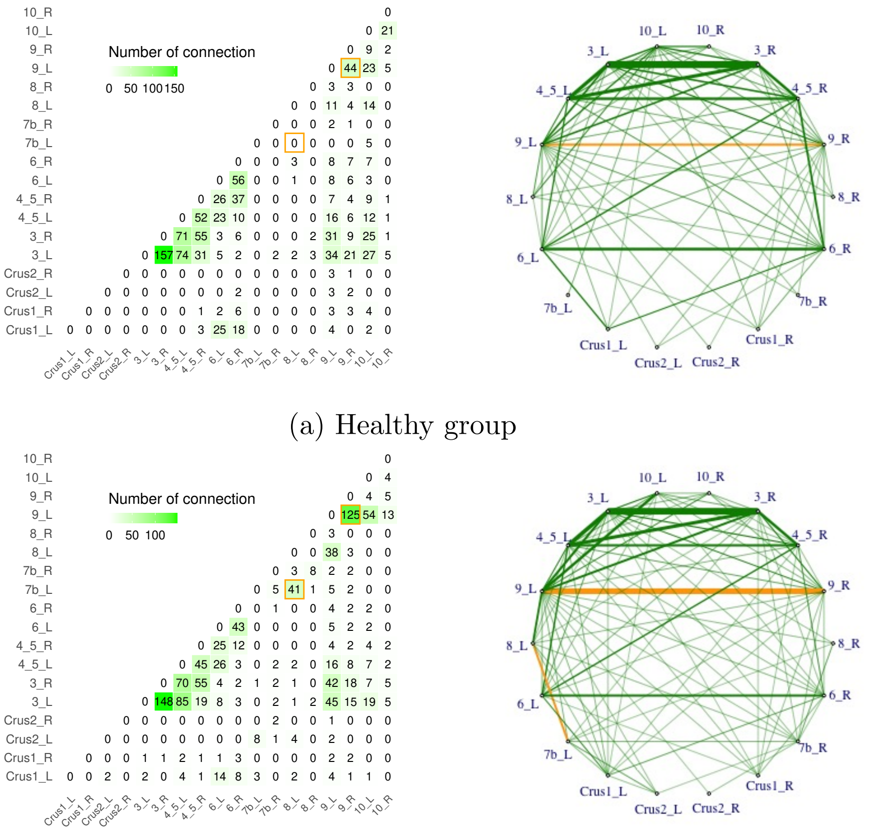
Figure A.16: [Result 1, GEN] Accumulated connections between different cerebellum regions estimated by generalized elastic net, showing the total connections between different cerebellum regions over 172 time points. The yellow squares on the left highlight the number of the edge 7b_L - 8_L and the edge 9_L - 9_R.



(a) Healthy group

(b) ADHD group

Figure A.17: [Result 1, GEN] Estimated partial correlations between different cerebellum regions estimated by generalized elastic net, illustrating how connections change over time. Each cell corresponds a connection at a given time point, and the color represents the magnitude of the estimated partial correlation value.



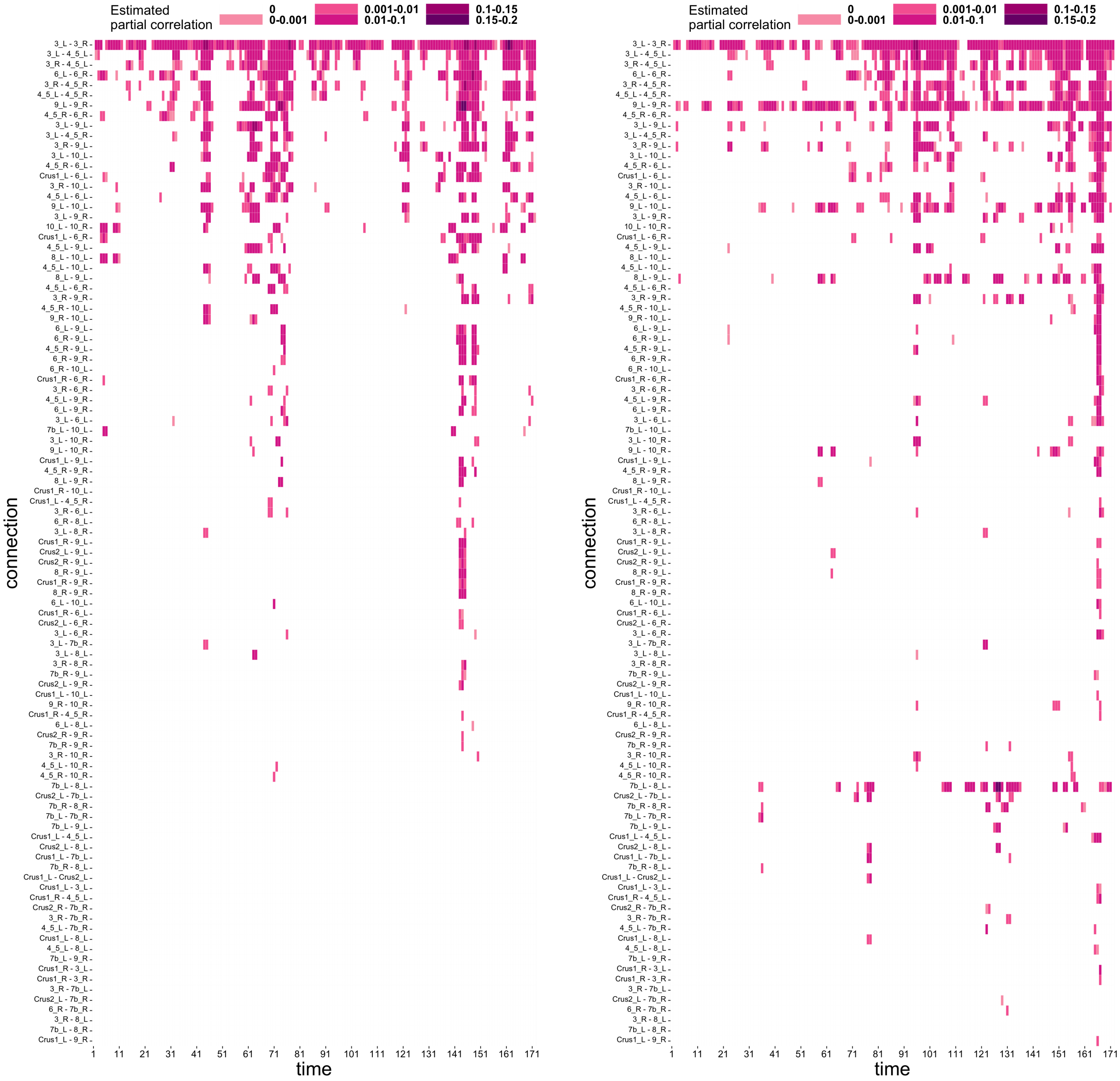
(a) Healthy group

	Crus1_L	Crus1_R	Crus2_L	Crus2_R	3_L	3_R	4_5_L	4_5_R	6_L	6_R	7b_L	7b_R	8_L	8_R	9_L	9_R	10_L	10_R
10_R																		0
10_L																	0	21
9_R																0	9	2
9_L															0	44	23	5
8_R														0	3	3	0	0
8_L													0	0	11	4	14	0
7b_R												0	0	0	2	1	0	0
7b_L											0	0	0	0	0	0	5	0
6_R										0	0	0	3	0	8	7	7	0
6_L									0	56	0	0	1	0	8	6	3	0
4_5_R								0	26	37	0	0	0	0	7	4	9	1
4_5_L							0	52	23	10	0	0	0	0	16	6	12	1
3_R						0	71	55	3	6	0	0	0	2	31	9	25	1
3_L					0	157	74	31	5	2	0	2	2	3	34	21	27	5
Crus2_R				0	0	0	0	0	0	0	0	0	0	0	3	1	0	0
Crus2_L			0	0	0	0	0	0	0	2	0	0	0	0	3	2	0	0
Crus1_R		0	0	0	0	0	0	1	2	6	0	0	0	0	3	3	4	0
Crus1_L	0	0	0	0	0	0	0	3	25	18	0	0	0	0	4	0	2	0

(b) ADHD group

	Crus1_L	Crus1_R	Crus2_L	Crus2_R	3_L	3_R	4_5_L	4_5_R	6_L	6_R	7b_L	7b_R	8_L	8_R	9_L	9_R	10_L	10_R
10_R																		0
10_L																	0	4
9_R																0	4	5
9_L															0	125	54	13
8_R														0	3	0	0	0
8_L													0	0	38	3	0	0
7b_R												0	3	8	2	2	0	0
7b_L											0	5	41	1	5	2	0	0
6_R										0	0	1	0	0	4	2	2	0
6_L									0	43	0	0	0	0	5	2	2	0
4_5_R								0	25	12	0	0	0	0	4	2	4	2
4_5_L							0	45	26	3	0	2	2	0	16	8	7	2
3_R						0	70	55	4	2	1	2	1	0	42	18	7	5
3_L					0	148	85	19	8	3	0	2	1	2	45	15	19	5
Crus2_R				0	0	0	0	0	0	0	0	2	0	0	1	0	0	0
Crus2_L			0	0	0	0	0	0	0	0	8	1	4	0	2	0	0	0
Crus1_R		0	0	0	1	1	2	1	1	3	0	0	0	0	2	2	0	0
Crus1_L	0	0	2	0	2	0	4	1	14	8	3	0	2	0	4	1	1	0

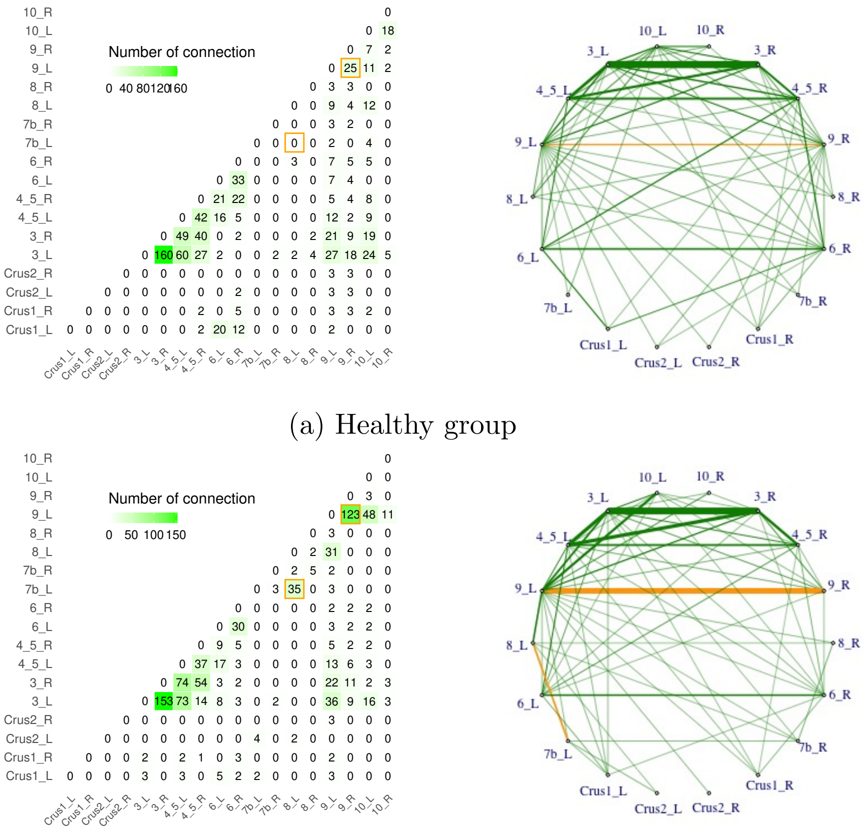
Figure A.18: [Result 2, GEN] Accumulated connections between different cerebellum regions estimated by generalized elastic net, showing the total connections between different cerebellum regions over 172 time points. The yellow squares on the left highlight the number of the edge 7b_L - 8_L and the edge 9_L - 9_R.



(a) Healthy group

(b) ADHD group

Figure A.19: [Result 2, GEN] Estimated partial correlations between different cerebellum regions estimated by generalized elastic net, illustrating how connections change over time. Each cell corresponds a connection at a given time point, and the color represents the magnitude of the estimated partial correlation value.



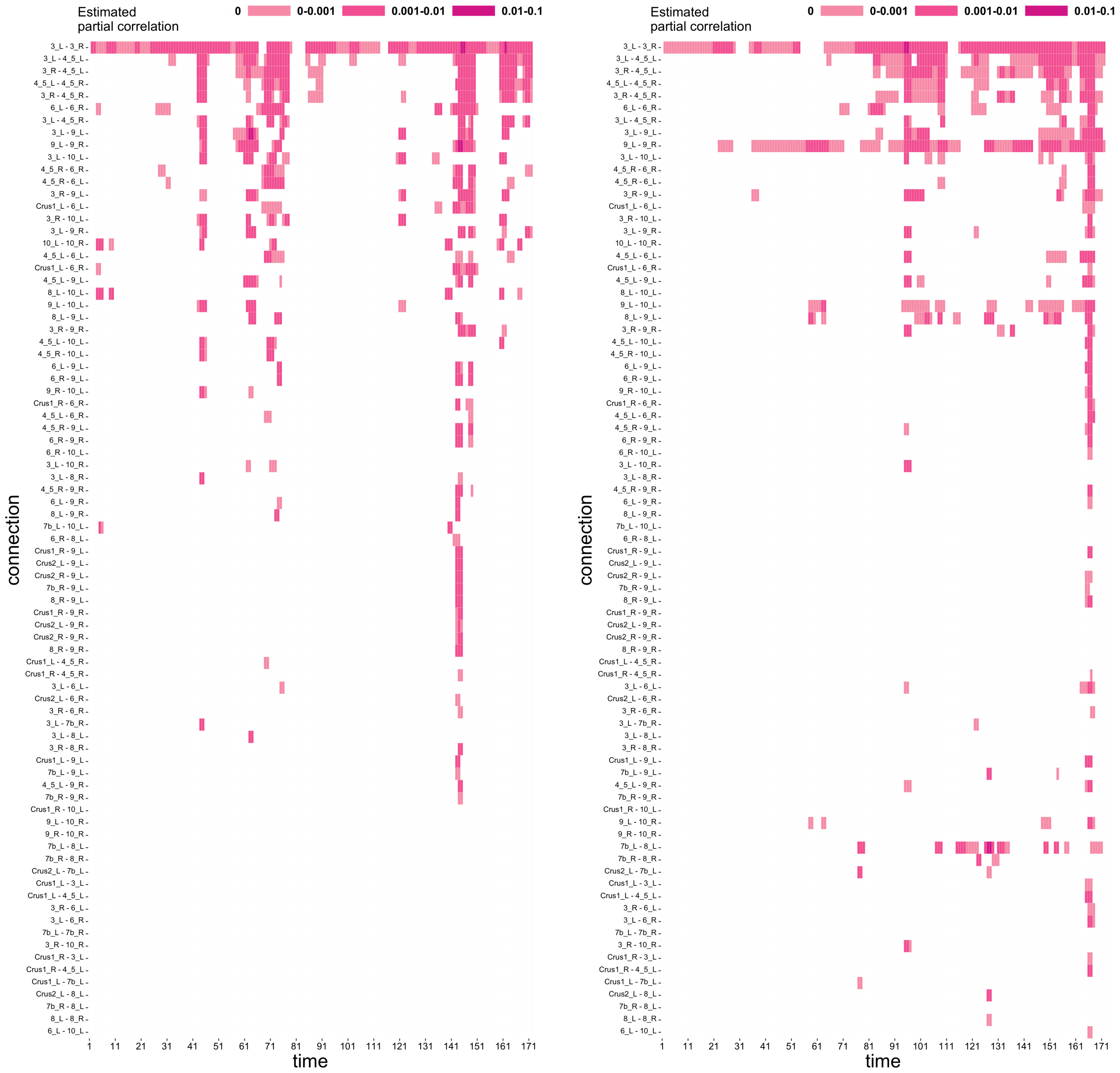
(a) Healthy group

	Crus1_L	Crus1_R	Crus2_L	Crus2_R	3_L	3_R	4_5_L	4_5_R	6_L	6_R	7b_L	7b_R	8_L	8_R	9_L	9_R	10_L	10_R
10_R																		0
10_L																	0	18
9_R																0	7	2
9_L															0	25	11	2
8_R														0	3	3	0	0
8_L													0	0	9	4	12	0
7b_R												0	0	0	3	2	0	0
7b_L											0	0	0	0	2	0	4	0
6_R										0	0	0	3	0	7	5	5	0
6_L									0	33	0	0	0	0	7	4	0	0
4_5_R								0	21	22	0	0	0	0	5	4	8	0
4_5_L							0	42	16	5	0	0	0	0	12	2	9	0
3_R						0	49	40	0	2	0	0	0	2	21	9	19	0
3_L					0	160	60	27	2	0	0	2	2	4	27	18	24	5
Crus2_R				0	0	0	0	0	0	0	0	0	0	0	3	3	0	0
Crus2_L			0	0	0	0	0	0	0	2	0	0	0	0	3	3	0	0
Crus1_R		0	0	0	0	0	0	2	0	5	0	0	0	0	3	3	2	0
Crus1_L	0	0	0	0	0	0	0	2	20	12	0	0	0	0	2	0	0	0

(b) ADHD group

	Crus1_L	Crus1_R	Crus2_L	Crus2_R	3_L	3_R	4_5_L	4_5_R	6_L	6_R	7b_L	7b_R	8_L	8_R	9_L	9_R	10_L	10_R
10_R																		0
10_L																	0	0
9_R																0	3	0
9_L															0	123	48	11
8_R														0	3	0	0	0
8_L													0	2	31	0	0	0
7b_R												0	2	5	2	0	0	0
7b_L											0	3	35	0	3	0	0	0
6_R										0	0	0	0	0	2	2	2	0
6_L									0	30	0	0	0	0	3	2	2	0
4_5_R								0	9	5	0	0	0	0	5	2	2	0
4_5_L							0	37	17	3	0	0	0	0	13	6	3	0
3_R						0	74	54	3	2	0	0	0	0	22	11	2	3
3_L					0	153	73	14	8	3	0	2	0	0	36	9	16	3
Crus2_R				0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
Crus2_L			0	0	0	0	0	0	0	0	4	0	2	0	0	0	0	0
Crus1_R		0	0	0	2	0	2	1	0	3	0	0	0	0	2	0	0	0
Crus1_L	0	0	0	0	3	0	3	0	5	2	2	0	0	0	3	0	0	0

Figure A.20: [Result 2, GFL] Accumulated connections between different cerebellum regions estimated by generalized fused lasso. The figures show the total connections between different cerebellum regions over 172 time points. The yellow squares on the left highlight the number of the edge 7b_L - 8_L and the edge 9_L - 9_R.



(a) Healthy group

(b) ADHD group

Figure A.21: [Result 2, GFL] Estimated partial correlations between different cerebellum regions estimated by generalized fused lasso, illustrating how connections change over time. Each cell corresponds a connection at a given time point, and the color represents the magnitude of the estimated partial correlation value.

Appendix B

Appendix for Chapter 4

B.1 MLE of $\hat{\beta}_0$

In this section, we provide the detailed derivation of $\hat{\beta}_0(\boldsymbol{\beta}(t))$ as defined in (4.9). Subsequently, we substitute this resulting maximum likelihood estimate of $\hat{\beta}_0$ back into (4.9), demonstrating how the equation presented in the first line of (4.10) is established.

The derivative of (4.9) with respect to β_0^{kl} is given by

$$\frac{\partial \ell_n(\beta_0, \boldsymbol{\beta}(t), \phi_0, \rho_0; D, z)}{\partial \beta_0^{kl}} = \frac{1}{\binom{n}{2}} \sum_{\nu=1}^{T} \sum_{1 \le i < j \le n} \frac{\mathbb{1}(z_i = k, z_j = l)}{\phi_0} \times \\ \Big\{ y_{ij}(t_\nu) \cdot \exp[(1-\rho_0)\{\beta_0^{kl} + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}] - \exp[(2-\rho_0)\{\beta_0^{kl} + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}] \Big\}. \quad \text{(B.1)}$$

Thus the second-order derivative is

$$\frac{\partial^2 \ell_n(\beta_0, \boldsymbol{\beta}(t), \phi_0, \rho_0; D, z)}{\partial [\beta_0^{kl}]^2} = \\ \frac{1}{\binom{n}{2}} \sum_{\nu=1}^{T} \sum_{1 \le i < j \le n} \frac{\mathbb{1}(z_i = k, z_j = l)}{\phi_0} \times \Big\{ (1-\rho_0) \cdot y_{ij}(t_\nu) \cdot \exp[(1-\rho_0)\{\beta_0^{kl} + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}] - \\ (2-\rho_0) \cdot \exp[(2-\rho_0)\{\beta_0^{kl} + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}] \Big\} < 0.$$

Therefore, the MLE of β_0^{kl} is given by the zero of (B.1) as

$$\hat{\beta}_0^{kl}(\boldsymbol{\beta}(t)) = \log \frac{\sum_{\nu=1}^{T} \sum_{1\le i<j\le n} y_{ij}(t_\nu)\exp[(1-\rho)\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)]\mathbb{1}(z_i=k, z_j=l)}{\sum_{\nu=1}^{T} \sum_{1\le i<j\le n} \exp[(2-\rho)\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)]\mathbb{1}(z_i=k, z_j=l)}$$
$$= \log \frac{\hat{\theta}_{kl}}{\hat{\gamma}_{kl}}.$$

Plugging $\hat{\beta}_0^{kl}(\boldsymbol{\beta}(t)) = \log \hat{\theta}_{kl}/\hat{\gamma}_{kl}$ into (4.9), we obtain the first line of the equation presented in (4.10):

$$\ell_n(\boldsymbol{\beta}(t), \phi_0, \rho_0; D, z) = \frac{1}{\binom{n}{2}} \sum_{\nu=1}^{T} \sum_{1\le i<j\le n} \sum_{k,l=1}^{K} \frac{\mathbb{1}(z_i=k, z_j=l)}{\phi_0} \times$$
$$\Big[\frac{y_{ij}(t_\nu)\exp[(1-\rho_0)\{\log \hat{\theta}_{kl}/\hat{\gamma}_{kl} + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}]}{1-\rho_0} - \frac{\exp[(2-\rho_0)\{\log \hat{\theta}_{kl}/\hat{\gamma}_{kl} + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}]}{2-\rho_0}\Big]$$
$$= \frac{1}{\binom{n}{2}} \sum_{k,l=1}^{K} \frac{1}{1-\rho_0} (\frac{\hat{\theta}_{kl}}{\hat{\gamma}_{kl}})^{1-\rho_0} \sum_{\nu=1}^{T} \sum_{1\le i<j\le n} \frac{\mathbb{1}(z_i=k, z_j=l)}{\phi_0} \cdot y_{ij}(t_\nu)\exp[(1-\rho_0)\{\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}] -$$
$$\frac{1}{\binom{n}{2}} \sum_{k,l=1}^{K} \frac{1}{2-\rho_0} (\frac{\hat{\theta}_{kl}}{\hat{\gamma}_{kl}})^{2-\rho_0} \sum_{\nu=1}^{T} \sum_{1\le i<j\le n} \frac{\mathbb{1}(z_i=k, z_j=l)}{\phi_0} \cdot \exp[(2-\rho_0)\{\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}(t_\nu)\}]$$
$$= \frac{1}{\phi_0} \sum_{k,l=1}^{K} \frac{1}{1-\rho_0} (\frac{\hat{\theta}_{kl}}{\hat{\gamma}_{kl}})^{1-\rho_0} \cdot \hat{\theta}_{kl} - \frac{1}{\phi_0} \sum_{k,l=1}^{K} \frac{1}{2-\rho_0} (\frac{\hat{\theta}_{kl}}{\hat{\gamma}_{kl}})^{2-\rho_0} \cdot \hat{\gamma}_{kl}$$
$$= \frac{1}{\phi_0} \frac{1}{(1-\rho_0)(2-\rho_0)} \sum_{k,l=1}^{K} \hat{\theta}_{kl}^{2-\rho_0} \cdot \hat{\gamma}_{kl}^{\rho_0 - 1}.$$

B.2 Proof of Theorem S1

In this section, we prove Theorem S1. Before laying out the main proof, we introduce several lemmas first.

Lemma 1. *Under Conditions 4.3.1 to 4.3.2,*

$$\frac{\hat{\gamma}_{kl}}{\sum\limits_{k,l}\hat{\gamma}_{kl}} = p_k p_l + o_p(1).$$

Proof. Proof According to Conditions 4.3.1 and 4.3.2, $\exp[(2-\rho)x_{ij}^{\top}\beta(t)]$ and $\mathbb{1}(z_i = k, z_j = l)$ are iid random variables, with mean γ and $p_k p_l$ respectively. Specifically, γ is a positive constant. By the weak law of large numbers, we have

$$\begin{aligned}
\frac{\hat{\gamma}_{kl}}{\sum\limits_{k,l}\hat{\gamma}_{kl}} &= \frac{2\sum\limits_{\nu=1}^{T}\sum\limits_{1\le i<j\le n}\exp[(2-\rho)\boldsymbol{x}_{ij}^{\top}\boldsymbol{\beta}(t_\nu)]\mathbb{1}(z_i = k, z_j = l)/\{n(n-1)\}}{2\sum\limits_{\nu=1}^{T}\sum\limits_{1\le i<j\le n}\exp[(2-\rho)\boldsymbol{x}_{ij}^{\top}\boldsymbol{\beta}(t_\nu)]/\{n(n-1)\}} \\
&= \frac{\gamma\cdot p_k p_l + o_p(1)}{\gamma + o_p(1)} \\
&= p_k p_l + o_p(1).
\end{aligned}$$

□

Lemma 2. *Under Conditions 4.3.1 to 4.3.2,*

$$\frac{\hat{\theta}_{kl}}{\sum\limits_{k,l}\hat{\theta}_{kl}} = p_k p_l + o_p(1).$$

Proof. Proof The proof is similar to that of Lemma 1. If we can show that, at each time point t_ν, $\nu = 1,\ldots,T$, $y_{ij}(t_\nu)\exp[(1-\rho)\boldsymbol{x}_{ij}^{\top}\boldsymbol{\beta}(t_\nu)]$ for $i,j = 1,\ldots,n$ are iid with a nonzero mean, we complete the proof. For each node pair (i,j), both their pairwise covariate $\boldsymbol{x}_{ij}$ and community labels c_i and c_j are iid. Moreover, $y_{ij}(t_\nu)$ conditional on x_{ij}, c_i and c_j are iid as well. Therefore, $y_{ij}(t_\nu)\exp[(1-\rho)\boldsymbol{x}_{ij}^\top\boldsymbol{\beta}(t_\nu)]$ for $i,j=1,\dots,n$ are iid, with mean

$$\begin{aligned}\mathbb{E}[y_{ij}(t_\nu)\exp\{(1-\rho)\boldsymbol{x}_{ij}^\top\boldsymbol{\beta}(t_\nu)\}] &= \mathbb{E}\Big(\mathbb{E}[y_{ij}(t_\nu)\exp\{(1-\rho)\boldsymbol{x}_{ij}^\top\boldsymbol{\beta}(t_\nu)\}\mid \boldsymbol{x},\boldsymbol{c}]\Big)\\ &= \mathbb{E}\Big[\mathbb{E}\{y_{ij}(t_\nu)|\boldsymbol{x},\boldsymbol{c}\}\cdot\exp\{(1-\rho)\boldsymbol{x}_{ij}^\top\boldsymbol{\beta}(t_\nu)\}\Big]\\ &= \mathbb{E}\Big[\exp\{\beta_0^{c_ic_j}+\boldsymbol{x}_{ij}^\top\boldsymbol{\beta}(t_\nu)\}\cdot\exp\{(1-\rho)\boldsymbol{x}_{ij}^\top\boldsymbol{\beta}(t_\nu)\}\Big]\\ &= \sum_{k,l}\mathbb{E}\Big[\exp\left\{\beta_0^{kl}+(2-\rho)\boldsymbol{x}_{ij}^\top\boldsymbol{\beta}(t_\nu)\right\}\Big]\cdot p_kp_l.\end{aligned}$$

Therefore, the expectation is a nonzero constant. □

Next, we prove Theorem S1.

Proof. Proof of Theorem S1 By Lemmas 1 and 2 and the continuous mapping theorem,

$$\begin{aligned}&\sum_{k,l}\hat{\theta}_{kl}^{2-\rho}\cdot\hat{\gamma}_{kl}^{\rho-1}\\ =&\Big[\sum_{k,l}\Big(\frac{\hat{\theta}_{kl}}{\sum_{k,l}\hat{\theta}_{kl}}\Big)^{2-\rho}\cdot\Big(\frac{\hat{\gamma}_{kl}}{\sum_{k,l}\hat{\gamma}_{kl}}\Big)^{\rho-1}\Big]\cdot(\sum_{k,l}\hat{\theta}_{kl})^{2-\rho}(\sum_{k,l}\hat{\gamma}_{kl})^{\rho-1}\\ =&\Big[\sum_{k,l}\Big\{(p_kp_l)^{2-\rho}+o_p(1)\Big\}\cdot\Big\{(p_kp_l)^{\rho-1}+o_p(1)\Big\}\Big]\cdot(\sum_{k,l}\hat{\theta}_{kl})^{2-\rho}(\sum_{k,l}\hat{\gamma}_{kl})^{\rho-1}\\ =&\Big[\sum_{k,l}\Big(p_kp_l+o_p(1)\Big)\Big]\cdot(\sum_{k,l}\hat{\theta}_{kl})^{2-\rho}(\sum_{k,l}\hat{\gamma}_{kl})^{\rho-1}\\ =&\Big[1+\frac{K(K+1)}{2}o_p(1)\Big]\cdot\Big(\theta+o_p(1)\Big)^{2-\rho}\Big(\gamma+o_p(1)\Big)^{\rho-1} \qquad \text{(B.2)}\\ =&\theta^{2-\rho}\gamma^{\rho-1}+o_p(1).\end{aligned}$$

(B.2) holds because $\sum_{k,l}\hat{\theta}_{kl}=\hat{\theta}=\theta+o_p(1)$ and $\sum_{k,l}\hat{\gamma}_{kl}=\hat{\gamma}=\gamma+o_p(1)$ by the weak law of large numbers. Therefore, we have

$$\begin{aligned}\frac{2}{n(n-1)}l_z(\boldsymbol{\beta}(t)) &= \frac{1}{\phi}\frac{1}{(1-\rho)(2-\rho)}\sum_{k,l}\hat{\theta}_{kl}^{2-\rho}\cdot\hat{\gamma}_{kl}^{\rho-1}\\ &= \frac{1}{\phi}\frac{1}{(1-\rho)(2-\rho)}\Big(\theta^{2-\rho}\cdot\gamma^{\rho-1}+o_p(1)\Big)\\ &= \frac{1}{\phi}\frac{1}{(1-\rho)(2-\rho)}\theta^{2-\rho}\cdot\gamma^{\rho-1}+o_p(1).\end{aligned}$$

□

Remark 3. *By Hölder's inequality, we have*

$$\sum_{k,l}\left(\frac{\hat{\theta}_{kl}}{\sum\limits_{k,l}\hat{\theta}_{kl}}\right)^{2-\rho}\cdot\left(\frac{\hat{\gamma}_{kl}}{\sum\limits_{k,l}\hat{\gamma}_{kl}}\right)^{\rho-1}\leq\left(\sum_{k,l}\frac{\hat{\theta}_{kl}}{\sum\limits_{k,l}\hat{\theta}_{kl}}\right)^{2-\rho}\cdot\left(\sum_{k,l}\frac{\hat{\gamma}_{kl}}{\sum\limits_{k,l}\hat{\gamma}_{kl}}\right)^{\rho-1}=1.$$

Then, it follows

$$\frac{2}{n(n-1)}l_z(\boldsymbol{\beta}(t))\leq\frac{1}{\phi}\frac{1}{(1-\rho)(2-\rho)}\theta^{2-\rho}\cdot\gamma^{\rho-1}. \tag{B.3}$$

In fact, Lemmas 1 and 2 establish the asymptotic equality conditions, which sharpen (B.3) *and lead to the conclusion in Theorem S1.*

B.3 Bspline Estimation in Step 1

In this section, we present the details of estimating the time-varying covariate coefficient $\hat{\boldsymbol{\beta}}(t)$ in accordance to (4.11) as part of Step 1 in our two-step estimation process.

According to Silverman (1985) and Green and Silverman (1993), each optimal $\boldsymbol{\beta}_u(t), u = 1, \cdots, p$ is a natural cubic spline with knots at time points where temporal data is observed. In practice, we use B-spline in the computations of smoothing splines (Hastie et al., 2009). We use $T+4$ B-spline basis functions $\{B_m(t)\}_{m=1}^{T+4}$, so we can represent the scalar $\boldsymbol{\beta}_u(t_\nu)$

as the $(\nu, u)-$th element in the $T-$by$-p$ matrix $\boldsymbol{B\eta}$, where

$$\boldsymbol{B}_{T\times(T+4)} = \begin{bmatrix} B_1(t_1) & \cdots & B_{T+4}(t_1) \\ \vdots & \ddots & \vdots \\ B_1(t_T) & \cdots & B_{T+4}(t_T) \end{bmatrix}.$$

and $\boldsymbol{\eta} \in \mathbb{R}^{(T+4)\times p}$ is the coefficient matrix that needs to be estimated. The p dimensional vector $\boldsymbol{\beta}(t_\nu) = (\boldsymbol{B}_{\nu\cdot}\boldsymbol{\eta})^\top$, where $\boldsymbol{B}_{\nu\cdot}$ represents the ν^{th} row of the matrix $\boldsymbol{B}$.

If we define $\boldsymbol{\Omega} \in \mathbb{R}^{(T+4)\times(T+4)}$ where $\boldsymbol{\Omega}_{ij} = \int B_i''(t)B_j''(t)dt$ and $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_p)^\top$, we can solve the $(T+4)\times p$ matrix $\boldsymbol{\eta}$ by plugging $\boldsymbol{\beta}(t_\nu) = (\boldsymbol{B}_{\nu\cdot}\boldsymbol{\eta})^\top$ in (4.11):

$$\begin{aligned}\hat{\boldsymbol{\eta}} = \arg\max_{\boldsymbol{\eta}} \\ \frac{1}{(1-\rho_0)(2-\rho_0)} \sum_{k,l} \left(\sum_{\nu=1}^{T} \sum_{1\le i<j\le n} y_{ij}(t) \exp[(1-\rho_0)\boldsymbol{B}_{\nu\cdot}\boldsymbol{\eta}x_{ij}] \mathbb{1}(z_i = k, z_j = l) \right)^{2-\rho_0} \times \\ \left(\sum_{\nu=1}^{T} \sum_{1\le i<j\le n} \exp[(2-\rho_0)\boldsymbol{B}_{\nu\cdot}\boldsymbol{\eta}x_{ij}] \mathbb{1}(z_i = k, z_j = l) \right)^{\rho_0 - 1} - \frac{1}{2}\boldsymbol{\lambda}^\top \cdot diag(\boldsymbol{\eta}^T\boldsymbol{\Omega}\boldsymbol{\eta})\end{aligned}$$

Once we have obtain $\hat{\boldsymbol{\eta}}$, we can calculate the estimated $\hat{\boldsymbol{\beta}}(t)$ in Step 1 by $\hat{\boldsymbol{\beta}}(t) = \boldsymbol{B}\hat{\boldsymbol{\eta}}$.

B.4 Additional Simulation Results

B.4.1 Tweedie Parameters Estimated in Simulation

Although our primary interest is to estimate the covariate coefficients and infer the community labels in our model, their estimation is affected by the Tweedie parameters ϕ and ρ. In this section, we provide the simulation results regarding ϕ and ρ in the Section 4.5. We report the estimated bias and standard error (SE) of the estimates of ϕ and ρ over 50 simulation runs in Table B.1, B.2, B.3, B.4 and B.5. To be more specific, we calculate the bias of the estimate $\hat{\phi}$ of ϕ with true value ϕ_0 by bias$(\hat{\phi}) = \sum_{i=1}^{50}(\hat{\phi}_i - \phi_0)/50$ and SE$(\hat{\phi}) = \sqrt{\sum_{i=1}^{50}(\hat{\phi}_i - \bar{\hat{\phi}})^2/49}$ where $\bar{\hat{\phi}}$ is the average of $\hat{\phi}$ over 50 simulation runs. In summary, the simulation results indicate that the estimates of ϕ and ρ are highly accurate.

			Scenario 1		Scenario 2		Scenario 3	
ϕ	ρ	n	Bias	SE	Bias	SE	Bias	SE
2	1.2	50	0.012	0.054	0.014	0.09	0.014	0.082
		100	-0.003	0.027	0.001	0.026	0.004	0.032
	1.5	50	0.014	0.072	0.018	0.085	0.011	0.085
		100	0.004	0.025	0.004	0.034	0	0.032
	1.8	50	-0.003	0.062	-0.004	0.07	-0.001	0.056
		100	0	0.03	0.001	0.028	-0.002	0.028
1	1.2	50	0.008	0.034	0.008	0.028	0.013	0.048
		100	0.002	0.014	-0.002	0.015	-0.001	0.012
	1.5	50	0.008	0.039	0.006	0.033	0.01	0.033
		100	-0.001	0.018	0	0.017	0.002	0.014
	1.8	50	0.001	0.037	0.008	0.034	0.007	0.034
		100	0.001	0.016	0.001	0.016	-0.001	0.016
0.5	1.2	50	0.004	0.015	0	0.022	0.007	0.023
		100	0.002	0.007	0	0.008	0	0.006
	1.5	50	0.005	0.021	0.002	0.022	0.009	0.03
		100	0	0.01	0	0.01	-0.001	0.01
	1.8	50	-0.003	0.016	0.008	0.023	0.002	0.031
		100	-0.002	0.009	-0.001	0.009	-0.001	0.010

Table B.1: Summary of estimated bias and standard error (SE) of estimated ϕ in scenario 1, 2 and 3 over 50 simulation runs.

B.4.2 Sensitivity Analysis of Tuning Parameters in TV-TSBM

In this section, we apply the TV-TSBM on two λ values of 1 and 0.1 respectively to conduct the sensitivity analysis of the simulation in Section 4.5.3.

By and large, the clustering outcomes across the three distinct tuning parameters measured by the NMI are relatively close, and all indicate high-quality clustering. With increasing λ values, the curvature of the estimated $\hat{\beta}(t)$ diminishes. Consequently, when the true curve $\beta(t)$ is linear, larger values of λ yield smaller errors in estimating $\hat{\beta}(t)$. Vice versa, a smaller λ leads to a better estimation of $\hat{\beta}(t)$ when the underlying curve is a sine function. In summary, the consistent clustering outcomes across various distinct λ values, coupled with the choice of a moderately penalized smoothness, substantiates the rationale behind adopting 0.5 as the preferred value for λ.

			Scenario 1		Scenario 2		Scenario 3	
ϕ	ρ	n	Bias	SE	Bias	SE	Bias	SE
2	1.2	50	0	0	0.002	0.014	0.002	0.014
		100	0	0	0	0	0	0
	1.5	50	0	0	0	0	0	0
		100	0	0	0	0	0	0
	1.8	50	0	0	0	0	0	0
		100	0	0	0	0	0	0
1	1.2	50	0.060	0.120	0	0	0.002	0.014
		100	0	0	0	0	0	0
	1.5	50	-0.002	0.014	0	0	0	0
		100	0	0	0	0	0	0
	1.8	50	-0.002	0.014	0	0	0	0
		100	0	0	0	0	0	0
0.5	1.2	50	0	0	-0.002	0.014	0.002	0.014
		100	0	0	0	0	0	0
	1.5	50	0.002	0.037	-0.002	0.014	0.002	0.032
		100	0	0	0	0	0	0
	1.8	50	0.006	0.054	0.008	0.039	0.002	0.037
		100	0.002	0.014	0	0	0	0

Table B.2: Summary of estimated bias and standard error (SE) of estimated ρ in scenario 1, 2 and 3 over 50 simulation runs.

			Weak Effect ($\beta = 1$)				Strong Effect ($\beta = 2$)			
ϕ	ρ	n	$\hat{\phi}$		$\hat{\rho}$		$\hat{\phi}$		$\hat{\rho}$	
			Bias	SE	Bias	SE	Bias	SE	Bias	SE
2	1.2	50	-0.002	0.063	0	0	0.002	0.073	0.002	0.014
		100	0.001	0.026	0	0	-0.006	0.031	0	0
	1.5	50	0.016	0.076	0	0	0.019	0.072	0	0
		100	0.005	0.033	0	0	0.005	0.039	0	0
	1.8	50	0.003	0.065	0	0	-0.007	0.066	0	0
		100	-0.002	0.032	0	0	0.004	0.035	0	0
1	1.2	50	0.003	0.03	0	0	0.002	0.032	0	0
		100	0	0.017	0	0	-0.002	0.013	0	0
	1.5	50	0.003	0.042	-0.002	0.014	0.009	0.052	0	0.02
		100	-0.001	0.013	0	0	0.001	0.016	0	0
	1.8	50	0.001	0.037	0	0	0.006	0.04	0	0
		100	-0.001	0.027	-0.002	0.014	0.003	0.02	0	0
0.5	1.2	50	0.009	0.026	0.004	0.02	0	0.014	0	0
		100	0.001	0.008	0	0	0.001	0.007	0	0
	1.5	50	0.003	0.027	-0.006	0.031	0.002	0.019	-0.006	0.024
		100	0	0.009	0	0	0	0.009	0	0
	1.8	50	0.005	0.028	-0.01	0.054	-0.002	0.025	-0.006	0.042
		100	-0.002	0.01	0	0	0.001	0.009	0	0

Table B.3: Summary of estimated bias and standard error (SE) of estimated ϕ and ρ in the static model with covariates over 50 simulation runs, with $(\beta_0^{kk}, \beta_0^{kl}) = (0.5, -0.5)$.

$(\beta_0^{kk}, \beta_0^{kl})$	ϕ			$\beta(t)$			
		$2t-1$	$\sin(2\pi t)$	$2t$	$\sin(2\pi t)$ +1	$0.5(2t-1)$	$0.5\sin(2\pi t)$
Scenario 1	Bias	0.002	0.004	-0.001	0.007	0.003	0.004
$(1, 0)$	SE	0.008	0.008	0.014	0.017	0.01	0.017
Scenario 2	Bias	0.002	0.005	0.004	0.005	0.002	0.001
$(0.5, -0.5)$	SE	0.008	0.006	0.026	0.008	0.008	0.007
Scenario 3	Bias	0.002	0.005	0	0.005	0.001	0.002
$(0, -1)$	SE	0.007	0.007	0.007	0.009	0.007	0.009
Scenario 4	Bias	0.001	0.003	0.002	0.004	0.001	0.001
$(0.5, 0)$	SE	0.007	0.007	0.007	0.006	0.007	0.007
Scenario 5	Bias	0.001	0.004	0.002	0.005	0.002	0.001
$(0.25, -0.25)$	SE	0.007	0.008	0.007	0.008	0.008	0.007
Scenario 6	Bias	0	0.004	0.001	0.006	0	0.006
$(0, -0.5)$	SE	0.008	0.007	0.009	0.007	0.007	0.032

Table B.4: Summary of estimated bias and standard error (SE) of estimated ϕ (with true value 1) in the time-varying model over 50 simulation runs (using $\lambda = 0.5$).

$(\beta_0^{kk}, \beta_0^{kl})$	ρ			$\beta(t)$			
		$2t-1$	$\sin(2\pi t)$	$2t$	$\sin(2\pi t)$ +1	$0.5(2t-1)$	$0.5\sin(2\pi t)$
Scenario 1	Bias	0	0	-0.002	0.002	0	0.002
$(1, 0)$	SE	0	0	0.014	0.014	0	0.014
Scenario 2	Bias	0	0	0.002	0	0	0
$(0.5, -0.5)$	SE	0	0	0.014	0	0	0
Scenario 3	Bias	0	0	0	0	0	0
$(0, -1)$	SE	0	0	0	0	0	0
Scenario 4	Bias	0	0	0	0	0	0
$(0.5, 0)$	SE	0	0	0	0	0	0
Scenario 5	Bias	0	0	0	0	0	0
$(0.25, -0.25)$	SE	0	0	0	0	0	0
Scenario 6	Bias	0	0	0	0	0	0.002
$(0, -0.5)$	SE	0	0	0	0	0	0.014

Table B.5: Summary of estimated bias and standard error (SE) of estimated ρ (with true value 1.5) in the time-varying model over 50 simulation runs (using $\lambda = 0.5$).

$(\beta_0^{kk}, \beta_0^{kl})$		$\beta(t)$					
		$2t-1$	$\sin(2\pi t)$	$2t$	$\sin(2\pi t)$ $+1$	$0.5(2t-1)$	$0.5\sin(2\pi t)$
Scenario 1	NMI	1 (0)	1 (0)	1 (0)	1 (0)	0.996 (0.004)	1 (0)
$(1, 0)$	$\text{Err}(\hat{\beta}(t))$	0.004 (0)	0.041 (0)	0.004 (0)	0.040 (0)	0.004 (0)	0.021 (0)
Scenario 2	NMI	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
$(0.5, -0.5)$	$\text{Err}(\hat{\beta}(t))$	0.004 (0)	0.048 (0)	0.005 (0)	0.046 (0)	0.005 (0)	0.024 (0)
Scenario 3	NMI	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
$(0, -1)$	$\text{Err}(\hat{\beta}(t))$	0.005 (0)	0.055 (0)	0.005 (0)	0.053 (0)	0.006 (0)	0.028 (0)
Scenario 4	NMI	0.996 (0.004)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
$(0.5, 0)$	$\text{Err}(\hat{\beta}(t))$	0.004 (0)	0.045 (0)	0.004 (0)	0.043 (0)	0.004 (0)	0.023 (0)
Scenario 5	NMI	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
$(0.25, -0.25)$	$\text{Err}(\hat{\beta}(t))$	0.004 (0)	0.048 (0)	0.004 (0)	0.046 (0)	0.004 (0)	0.024 (0)
Scenario 6	NMI	1 (0)	0.996 (0.004)	1 (0)	1 (0)	1 (0)	0.996 (0.004)
$(0, -0.5)$	$\text{Err}(\hat{\beta}(t))$	0.005 (0)	0.052 (0)	0.004 (0)	0.050 (0)	0.005 (0)	0.026 (0)

Table B.6: Summary of clustering and estimation performance (using $\lambda = 1$) from the time-varying model over 50 simulation runs, with $\phi = 1$, $\rho = 1.5$ and $n = 50$.

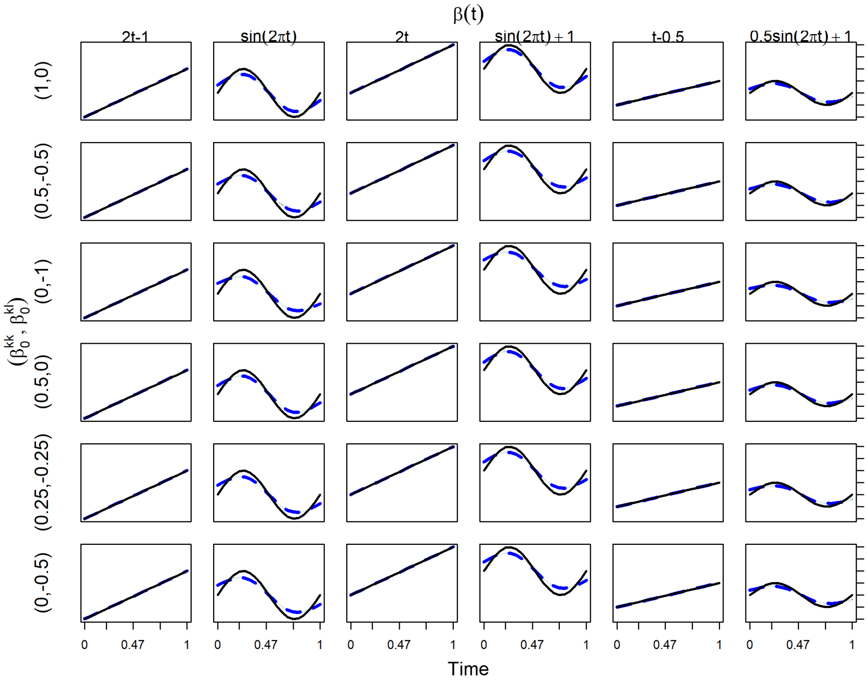


Figure B.1: Estimations of the time-varying coefficients for 36 designs with $\lambda = 1$, i.e. six block matrices by six functions for $\beta(t)$. In each panel, the black solid line represents the true $\beta(t)$ while the blue dashed line denotes the mean curve of $\hat{\beta}(t)$ and the light blue shadow marks the corresponding confidence band.

$(\beta_0^{kk}, \beta_0^{kl})$		$\beta(t)$					
		$2t-1$	$\sin(2\pi t)$	$2t$	$\sin(2\pi t)$ $+1$	$0.5(2t-1)$	$0.5\sin(2\pi t)$
Scenario 1	NMI	1 (0)	1 (0)	1 (0)	1 (0)	0.996 (0.004)	1 (0)
$(1, 0)$	$\mathrm{Err}(\hat{\beta}(t))$	0.005 (0)	0.008 (0)	0.005 (0)	0.008 (0)	0.005 (0)	0.006 (0)
Scenario 2	NMI	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
$(0.5, -0.5)$	$\mathrm{Err}(\hat{\beta}(t))$	0.005 (0)	0.009 (0)	0.006 (0)	0.009 (0)	0.006 (0)	0.007 (0)
Scenario 3	NMI	1 (0)	1 (0)	1 (0)	0.996 (0.004)	1 (0)	1 (0)
$(0, -1)$	$\mathrm{Err}(\hat{\beta}(t))$	0.006 (0)	0.012 (0)	0.006 (0)	0.011 (0)	0.007 (0)	0.008 (0)
Scenario 4	NMI	0.996 (0.004)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
$(0.5, 0)$	$\mathrm{Err}(\hat{\beta}(t))$	0.005 (0)	0.009 (0)	0.006 (0)	0.008 (0)	0.006 (0)	0.007 (0)
Scenario 5	NMI	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
$(0.25, -0.25)$	$\mathrm{Err}(\hat{\beta}(t))$	0.006 (0)	0.009 (0)	0.005 (0)	0.009 (0)	0.005 (0)	0.007 (0)
Scenario 6	NMI	1 (0)	1 (0)	1 (0)	0.996 (0.004)	1 (0)	0.996 (0.004)
$(0, -0.5)$	$\mathrm{Err}(\hat{\beta}(t))$	0.006 (0)	0.010 (0)	0.006 (0)	0.010 (0)	0.006 (0)	0.007 (0)

Table B.7: Summary of clustering and estimation performance (using $\lambda = 0.1$) from the time-varying model over 50 simulation runs, with $\phi = 1$, $\rho = 1.5$ and $n = 50$.

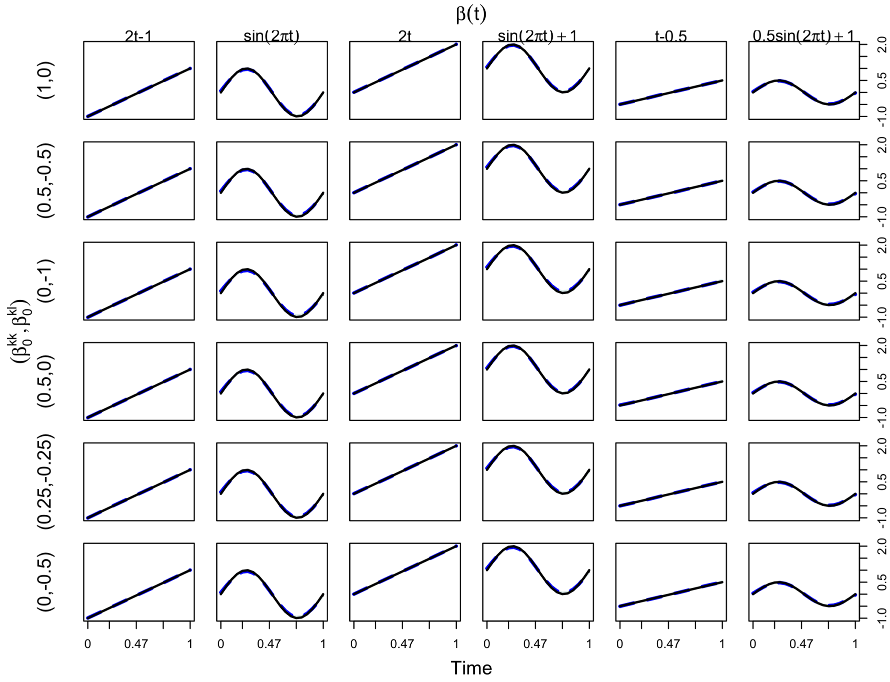


Figure B.2: Estimations of the time-varying coefficients for 36 designs with $\lambda = 0.1$, i.e. six block matrices by six functions for $\beta(t)$. In each panel, the black solid line represents the true $\beta(t)$ while the blue dashed line denotes the mean curve of $\hat{\beta}(t)$ and the light blue shadow marks the corresponding confidence band.

Glossary

ADMM The alternating direction methods of multipliers, an optimization algorithm that addresses constrained optimization problems by iteratively decomposing them into subproblems, facilitating parallel computation and efficient convergence. 12, 14, 15, 30, 33–36, 52, 98, 116, 119

ELBO The evidence lower bound, a quantity in variational inference that serves as a lower bound on the marginal likelihood of observed data in Bayesian models. 22–25

EM The Expectation Maximisation algorithm, an iterative method between the "Expectation" step, where it estimates the values of latent variables, and the "Maximization" step, where it maximizes the likelihood function by updating parameters. xiii, 20, 21, 23, 25, 88, 89, 92, 101

GGM Gaussian Graphical Models, a statistical modeling approach that assumes a Gaussian distribution to represent dependencies among multiple variables. 5–11, 27, 29, 30, 38, 97, 98

MCMC The Markov Chain Monte Carlo, a computational method used for sampling from complex probability distributions by constructing a Markov chain that converges to the desired distribution. 21, 55, 101

SBM Stochastic Block Model, a network modeling approach that groups nodes into blocks, with connection probabilities depending on block assignments, revealing latent community structures in complex systems. 16–18, 20, 21, 24, 54–58, 60, 64, 69, 74, 85, 89, 97, 98, 100

VEM The variational Expectation Maximization algorithm, a technique for approximating maximum likelihood estimates by optimizing a lower bound on the likelihood. xiii, 20–24