

© 2022 Joshua Daniel Loyal

DIMENSION REDUCTION METHODS FOR QUANTIFYING LOCAL VARIABLE
IMPORTANCE AND THE STATISTICAL ANALYSIS OF NETWORK DATA

BY

JOSHUA DANIEL LOYAL

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Statistics
in the Graduate College of the
University of Illinois Urbana-Champaign, 2022

Urbana, Illinois

Doctoral Committee:

Professor Yuguo Chen, Chair and Co-Director of Research
Assistant Professor Ruqing Zhu, Co-Director of Research
Professor Feng Liang
Professor Douglas G. Simpson

Abstract

Many scientific problems require statistical models that satisfy two competing goals: the models must be flexible enough to explain the data while remaining amiable to interpretation and scrutiny. Dimension reduction techniques, which use low-dimensional representations to distill complex data into essential components, suggest a way to achieve these modeling goals. This dissertation develops various dimension reduction methods for problems in predictive modeling and the analysis of complex network data that result in flexible and interpretable statistical inference.

Chapter 2 uses dimension reduction methods to quantify local variable importance in random forests. Random forests are one of the most popular machine learning methods due to their accuracy; however, they only provide variable importance in a global sense. There is an increasing need for such assessments at a local level, motivated by applications in personalized medicine, policy-making, and bioinformatics. Chapter 2 proposes a new nonparametric estimator that pairs the flexible random forest kernel with local sufficient dimension reduction to adapt to a regression function’s local structure. This allows us to estimate a meaningful directional local variable importance measure at each prediction point. We develop a computationally efficient fitting procedure and provide sufficient conditions for the recovery of the splitting directions. We demonstrate significant accuracy gains of our proposed estimator over competing methods on simulated and real regression problems. Finally, we apply the proposed method to seasonal particulate matter concentration data collected in Beijing, China, which yields meaningful local importance measures.

The rest of the dissertation focus on problems related to complex network data. Chapter 3 studies the evolution of communities in dynamic (time-varying) network data, which is a prominent topic of interest. A popular approach to understanding these dynamic networks is to embed the dyadic relations into a latent metric space. While methods for clustering with this approach exist for dynamic networks, they all assume a static community structure. Chapter 3 presents a Bayesian nonparametric model for dynamic networks that can model networks with evolving community structures. Our model extends existing latent space approaches by explicitly modeling the additions, deletions, splits, and mergers of groups with a hierarchical Dirichlet process hidden Markov model. Our proposed approach, the hierarchical Dirichlet process latent position cluster model (HDP-LPCM), incorporates transitivity, models both individual and group level aspects of the data, and avoids the computationally expensive selection of the number of groups required by most popular methods. We provide a Markov chain Monte Carlo estimation algorithm and demonstrate its ability to detect evolving community structure in a network of military alliances during the Cold War and a narrative network constructed from the Game of Thrones television series.

Dynamic multilayer networks frequently represent the structure of multiple co-evolving relations; however, statistical models are not well-developed for this prevalent network type. In Chapter 4, we propose a new latent space model for dynamic multilayer networks. The key feature of our model is its ability to identify common time-varying structures shared by all layers while also accounting for layer-wise variation and degree

heterogeneity. We establish the identifiability of the model’s parameters and develop a structured mean-field variational inference approach to estimate the model’s posterior, which scales to networks previously intractable to dynamic latent space models. We demonstrate the estimation procedure’s accuracy and scalability on simulated networks. We apply the model to two real-world problems: discerning regional conflicts in a data set of international relations and quantifying infectious disease spread throughout a school based on the student’s daily contact patterns.

In Chapters 3 and 4, we used latent space models (LSMs) to model network data by embedding a network’s nodes into a low-dimensional latent space. Correctly choosing the dimension of this space remains a challenge. Chapter 5 proposes a new Bayesian LSM for dynamic networks that not only fixes parameter identifiability issues that have previously impeded dimension selection but also models a larger class of dynamic networks than previous approaches. With these issues resolved, we propose a Bayesian approach to dimension selection for static and dynamic LSMs based on an ordered spike-and-slab prior that provides improved dimension estimation and satisfies several appealing theoretical properties. In particular, we show that the static model’s posterior concentrates on low-dimensional models near the truth. These models are accompanied by a novel parameter expansion scheme that allows for efficient Markov chain Monte Carlo estimation using a Metropolis-within-Gibbs sampler with Hamiltonian Monte Carlo proposals. We demonstrate our approach’s versatility and consistent dimension selection on simulated networks. Lastly, we use the static and dynamic models to study a static protein interaction network and the global arms trade’s dynamics during the Cold War.

Dedicated to my parents, Daniel and Wendy, for their unconditional love and encouragement, and to my brother, Alexander, for his unwavering support and friendship.

Acknowledgments

My Ph.D. studies at the University of Illinois at Urbana-Champaign were a great learning experience academically, professionally, and personally. Completing my graduate education and this dissertation would not have been possible without the help of many people.

First, I am deeply grateful to my advisors, Dr. Yuguo Chen and Dr. Ruqing Zhu, for their mentorship, guidance, and support over the years. I would like to thank Yuguo for countless discussions, suggestions, and advice, which have always pushed me to conduct better research and tackle new and exciting problems. I am also incredibly grateful for the many opportunities he provided me to develop my research and career. I want to thank Ruqing for his mentorship, especially early in my statistics career. I would always leave our meetings with a deeper understanding of statistics that still influences my thinking today.

I would like to thank my committee members, Dr. Feng Liang and Dr. Douglas G. Simpson, for serving on my committee, providing insightful suggestions on my research, and supporting me during my job search. They have generously contributed to my development in many ways.

Also, I would like to acknowledge Sandia National Laboratories, the Laboratory Directed Research and Development Program, and my collaborators there for generously supporting my work and motivating many of the ideas that make up this dissertation.

Next, I would like to thank all the faculty, staff, and students in the Department of Statistics for creating such a friendly and nurturing environment. Specifically, Aaron Thompson and Joseph Zarnsy for answering my many questions and ensuring the Ph.D. process proceeded smoothly. I would also like to acknowledge Dr. Feng Liang and Dr. Georgios Fellouris for their courses early on in my education that deeply impacted my statistical thinking. Lastly, I am incredibly grateful for my fellow Ph.D. students who made my time here so enjoyable. Special thanks go to Alton Barbehenn, William Biscarri, Mauricio Campos, Eduardo Cardenas-Torres, Anamitra Chaudhuri, Robert Garrett, Trevor Harris, Robin Tu, Austin Warner, and Theren Williams for their friendship over the years.

I am also thankful for my longtime friends, Arun Augustine and Kevin Wang. Their constant friendship went a long way in helping me complete this journey.

Finally, I would like to express my warmest gratitude to my family. Thank you to my parents, Daniel and Wendy, for their unconditional love and encouragement throughout my life. The emotional support they provide means the world to me. Also, I want to thank my brother, Alexander, who has dropped everything to help and encourage me on more than one occasion. His love and friendship are constant motivators in my life. I love you all immensely.

Table of contents

Chapter 1 Introduction	1
1.1 Statistical Network Analysis	1
1.2 Network Models	5
1.3 Overview of Contributions	11
Chapter 2 Dimension Reduction Forests	14
2.1 Introduction	14
2.2 Motivation	15
2.3 Dimension Reduction Forests	17
2.4 Theoretical Results	21
2.5 Simulation Studies	22
2.6 Real Data Applications	26
2.7 Discussion	28
2.8 Supplementary Material	28
Chapter 3 Modeling Evolving Community Structure in Dynamic Networks	41
3.1 Introduction	41
3.2 Related Methods	43
3.3 The HDP Latent Position Cluster Model	44
3.4 Estimation	50
3.5 Simulation Study	52
3.6 Real Data Application	57
3.7 Discussion	61
3.8 Supplementary Material	62
Chapter 4 An Eigenmodel for Dynamic Multilayer Networks	86
4.1 Introduction	86
4.2 An Eigenmodel for Dynamic Multilayer Networks	88
4.3 Variational Inference	93
4.4 Simulation Studies	98
4.5 Real Data Applications	100
4.6 Discussion	107
4.7 Supplementary Material	109
Chapter 5 Spike-and-Slab Priors for Dimension Selection in Network Eigenmodels	133
5.1 Introduction	133
5.2 The Models	135
5.3 Dimension Selection	138
5.4 Estimation	141
5.5 Simulation Studies	145
5.6 Real Data Applications	150
5.7 Discussion	155
5.8 Supplementary Material	157
References	173

Chapter 1

Introduction

The recent popularity of statistics, machine learning, and data science partly stems from their promise to deliver meaningful insights into utterly complex data sets. This complexity is often due to the heterogeneous components that make up the data. As such, successful statistical applications often require models that satisfy dual objectives: the models must be (1) flexible enough to account for the essential heterogeneous structures in the data and (2) simple enough to interpret. In this dissertation, I develop statistical methodology aimed at satisfying these dual objectives for problems in predictive modeling and network science. In particular, I focus on using dimension reduction methods to reduce problems described by seemingly high-dimensional data structures into easier-to-interpret low-dimensional components that maintain some of the original data’s core elements. In Chapter 2, I combine sufficient dimension reduction techniques (K.-C. Li, 1991) with random forests (Breiman, 2001) to tackle the difficult problem of assessing local variable importance, which is an essential part of predictive modeling. The remainder of the dissertation focuses on the statistical modeling of network data. In particular, the methods introduced in Chapters 3 – 5 use low-dimensional latent variables (P. D. Hoff et al., 2002) to model various types of complex networks found in real-world applications.

The rest of this chapter is structured as follows. Since the majority of this dissertation focuses on the statistical analysis of network data, I start by reviewing terminology and ideas from network science that may be unfamiliar to readers with only a statistics background. In particular, Section 1.1 provides a high-level overview of basic concepts in statistical network analysis, and Section 1.2 summarizes foundational network models found in the literature. Finally, Section 1.3 briefly summarizes the contributions of the remaining chapters.

1.1 Statistical Network Analysis

Network data is becoming increasingly commonplace in many scientific applications, including epidemiology, sociology, biology, and other fields.¹ A network is composed of a set of actors (or nodes) and dyadic relationships (or ties) between actors. In this section, I summarize fundamental network terminology and methodology necessary to understand the statistical models developed in Chapters 3 – 5.

¹This chapter is based on research published as: Loyal, J. D., and Chen, Y. (2020). Statistical network analysis: A review with applications to the coronavirus disease 2019 pandemic. *International Statistical Review*, 88(2), 419–440.

1.1.1 Network Representations

Real-world networks are often represented by graphs. Depending on the network, the graph may be undirected, directed, weighted, time-dependent, or various combinations of these. A graph G consists of a pair (V, E) , where V is a set of vertices (or actors in the network context), and E is a set of dyads, $E \subseteq V \times V$, known as edges (or relations in the network context). A directed graph treats each edge pair (i, j) and (j, i) as distinct, while undirected graphs do not. Often we focus on parts of the full graph G called subgraphs. A graph $H = (V_H, E_H)$ is a subgraph of $G = (V, E)$ if $V_H \subseteq V$ and $E_H \subseteq E$. In network science, a graph is not synonymous with a network, but rather one possible mathematical representation.

An adjacency matrix is another popular representation for finite networks with n actors. In this case, the vertex set is taken as $V = \{1, \dots, n\}$. The adjacency matrix $\mathbf{Y}$ corresponding to the graph $G = (V, E)$ is the $n \times n$ binary matrix with elements

$$Y_{ij} = \begin{cases} 1, & \text{if } (i, j) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

The adjacency matrix uniquely determines a finite graph. The matrix $\mathbf{Y}$ is symmetric for undirected graphs. Note that real-valued edge weights replace the binary entries of the adjacency matrix for relations represented by arbitrary real numbers, e.g., the amount of trade in dollars between nation i and nation j . We refer to such graphs as weighted graphs. Unless otherwise stated, we assume all graphs are undirected and unweighted in this work. As we show in the following sections, many network characteristics are expressible as functions of the adjacency matrix.

Although considerable effort has been put into understanding static networks, many real-world networks are dynamic (time-dependent) in nature. A dynamic network is a collection of time-indexed graphs $G_t = (V_t, E_t)$, where time t takes values in a continuous or discrete time interval $\mathcal{T}$. Here, V_t and E_t indicate the set of vertices and edges present in the network at time t , respectively. Many analyses transform a dynamic network into a static network by combining the vertex and edge sets over all time points, i.e., $G = (\bigcup_{t \in \mathcal{T}} E_t, \bigcup_{t \in \mathcal{T}} V_t)$. However, this conversion removes temporal characteristics that are often of primary interest. For example, the analysis of dynamic networks reveals how friendships form and dissolve, infectious diseases spread, and economic and political partnerships change over time. As such, the majority of this dissertation, Chapters 3 – 5, develops statistical models for dynamic networks data.

1.1.2 Characterizing Networks

When studying real-world networks, such as contact networks in a city, we are unlikely to measure the exact adjacency matrix. Furthermore, any two cities have different adjacency matrices. This heterogeneity might suggest that any attempt to transfer knowledge from one network to another is bound to fail. However, we expect networks generated by a common mechanism to share some characteristics. For example, Holland and Leinhardt (1976) observed a nontrivial prevalence of “triangles” in friendship networks. By identifying the essential structures in a specific type of network, we can better understand the processes behind how these networks formed. In this section, we summarize characteristics commonly measured during network analyses.

Vertex Degree and Degree Distributions

Two of the most ubiquitous network characteristics are vertex degree and the degree distribution. The out-degree $d_{\text{out}}(i)$ of a vertex i is the number of edges starting from it, i.e., $\sum_{j=1}^n Y_{ij}$. Similarly, the in-degree $d_{\text{in}}(i)$ is the number of edges going into it, i.e., $\sum_{j=1}^n Y_{ji}$. For an undirected graph, the in-degree and out-degree are equal. In this case, this number is called the degree of the vertex $d(i)$. The degree sequence of an undirected graph is the collection of all vertex degrees $\mathbf{d} = (d(1), d(2), \dots, d(n))$.

Given a graph G , we define f_d as the fraction of vertices with degree $d(i) = d$. The collection $\{f_d\}_{d \geq 0}$ is called the degree distribution of G . As we show in Chapter 4, the first two moments of the degree distribution inform whether an epidemic occurs on a network. We define the k -th degree moment of a graph as

$$\langle d^k \rangle = \frac{1}{n} \sum_{i=1}^n [d(i)]^k. \quad (1.1)$$

The average degree of a graph is $\langle d \rangle = 2(|E| / |V|)$. The degree variance, defined as $\langle d^2 \rangle - \langle d \rangle^2$, measures a network's degree heterogeneity. When every vertex has the same degree d , so zero degree variance, then we call the graph d -regular or homogeneous.

Assortative and Disassortative Mixing

Despite its popularity, the degree distribution does not capture all aspects of a network. To move beyond the degree distribution, we note that there are usually strong correlations between nodes of different degrees. In assortative networks, high degree nodes tend to connect with other high degree nodes. In disassortative networks, high degree nodes prefer to connect to low degree nodes. Whether a network is assortative or disassortative has major implications for how information propagates through a network. For example, the removal of high degree nodes in disassortative networks massively limits the ability of an infectious disease to spread throughout a contact network (Kiss et al., 2008). Developing a principled statistical approach to modelling assortative and disassortative dynamic networks is a major focus of Chapters 4 and 5 of this dissertation.

The degree correlation function is one way to assess whether a network is assortative or disassortative. Define $f_{d'|d}$ as the relative frequency with which edges from a vertex of degree d connect to another vertex of degree d' . The degree correlation function is the mean of this conditional distribution:

$$\text{Cor}(d) = \sum_{d'} d' f_{d'|d}. \quad (1.2)$$

As a function of d , the degree correlation function increases for assortative networks and decreases for disassortative networks. Other popular measures of degree correlation include visualization of the joint degree distribution and calculation of the degree correlation coefficient introduced in Newman (2002).

In general, assortative (disassortative) mixing refers to the propensity of nodes to connect with other nodes with similar (dissimilar) characteristics. Usually, aspects other than degree are the basis of these correlations. For example, students are often more likely to form friendships with other students in the same grade. Note that assortative mixing is known as homophily in the social network literature. Determining features that govern assortative or disassortative mixing in a statistically principled manner is an essential component of network modeling.

Reciprocity

Reciprocity is a pairwise feature specific to directed networks. Reciprocity measures how often ties are reciprocated in a directed network, i.e., the frequency that both $(i, j) \in E$ and $(j, i) \in E$. The ratio between the number of reciprocated edges over the total number of edges measures the extent of reciprocity in a network. This ratio is zero for networks containing no mutual ties, and it is one for networks where all edges are reciprocated.

The level of reciprocity in a directed network provides insight into the nature of the relation. A high degree of reciprocity indicates that the relationship is often mutual between actors. Such an observation has important modeling implications. For example, it is common to treat a directed relation as undirected due to technical constraints. One can justify this decision for highly reciprocal relations.

Transitivity and Higher-Order Structures

Pairwise interactions do not uniquely characterize networks. The concept of transitivity, also known as clustering, pertains to triplets of actors. Transitivity quantifies the frequency with which connected triplets (i.e., a subgraph of three vertices connected by two edges) close to form triangles. In a social network context, transitivity quantifies the ‘a friend of my friend is my friend’ phenomenon. The clustering coefficient (Kiss et al., 2017; Newman, 2010) measures the amount of transitivity in a network:

$$\text{cl}(G) = \frac{6\tau_{\Delta}(G)}{\tau_3(G)} = \frac{\text{tr}(\mathbf{Y}^3)}{\|\mathbf{Y}^2\| - \text{tr}(\mathbf{Y}^2)}, \quad (1.3)$$

where $\tau_{\Delta}(G)$ is the number of triangles in the graph G , $\tau_3(G)$ is the number of connected triplets, and $\|\mathbf{Y}\| = \sum_{i=1}^n \sum_{j=1}^n Y_{ij}$. A non-zero entry Y_{ij}^k of $\mathbf{Y}^k$ indicates that nodes i and j are connected by Y_{ij}^k paths of length k . This means $\text{tr}(\mathbf{Y}^3)$ in the numerator measures the amount of closed loops of length three in a graph and the denominator measures all non-backtracking paths of length two.

Higher-order connected subgraphs containing more than three nodes may be relevant in real-world networks. Such repeating subgraphs are called motifs. Common motifs include k -stars, k -cycles, and k -cliques. Depending on the relation of interest, one focuses on motifs that have interpretable meanings, such as feed-back loops in biological networks. Determining whether meaningful higher-order motifs occur more often than by chance is a crucial aspect of network analysis.

Shortest Paths, Connectedness, and Components

The previous sections characterized networks based on local vertex-level properties; however, networks also contain distinguishable global structures. As we show in Chapter 4, models that can detect important global structures have major epidemiological applications. In particular, identifying interventions that can potentially reduce the rate of infectious disease spread by altering the network’s global topology.

A network’s connectedness is an important global characteristic. A directed path connecting vertices $u, v \in V$ is a sequence of vertices $v_0, \dots, v_m \in V$ such that $(v_i, v_{i+1}) \in E$ for all i and $v_0 = u, v_m = v$. We call a graph strongly connected if there is a direct path from any vertex to any other vertex. In the case of undirected graphs, we say the graph is connected. Many graphs are not connected; however, they often have connected components. A connected component is a maximally connected subgraph, which means that the addition of any other vertex to the subgraph would result in an unconnected subgraph. The giant

component is a graph’s largest connected component. In network analysis, it is common to focus on the giant component.

Lastly, we are often interested in the distance between vertices on a graph. The shortest path, geodesic distance, or distance between two vertices is the number of edges in the shortest directed path connecting the two vertices. In addition, the longest geodesic distance between two nodes in a graph is called the diameter. The diameter of real-world social networks is often quite small, scaling logarithmically in the number of vertices (Watts & Strogatz, 1998).

1.2 Network Models

Chapters 3 – 5 develop various network models that allow for statistical inference of complex network data. Formally, a network model is a collection of probability distributions over graphs

$$\{\mathbb{P}_{\boldsymbol{\theta}}(G), G \in \mathcal{G} : \boldsymbol{\theta} \in \Theta\}, \quad (1.4)$$

where $\mathcal{G}$ is a collection of possible graphs, $\mathbb{P}_{\boldsymbol{\theta}}$ is a probability distribution on $\mathcal{G}$, and $\boldsymbol{\theta}$ is a vector of parameters that takes values in Θ . Network models allow for the rigorous quantification of many questions in network science. For example, the specification of a network model is the starting point for formal significance tests of network characteristics, goodness-of-fit tests for mechanisms leading to plausible network formation, models that predict the presence of ties between actors, and forecasts of edge formation or dissolution. We divide network models into two broad categories: (1) mathematical network models and (2) statistical network models. We review each model category in the remainder of this section.

1.2.1 Mathematical Network Models

Mathematical network models provide closed-form analytical statements about network phenomena due to their mathematical tractability. However, their simplicity often makes them poor approximations to real-world networks. Nonetheless, the intuition they provide motivates the development of more complicated statistical network models. This section introduces the essential mathematical network models in the literature.

Random Graph Models

Random graph models are the simplest – although one of the most useful – classes of network models. A random graph model places a uniform distribution over a pre-specified collection of graphs. The Erdős-Rényi model is the earliest studied random graph model. There are two definitions of this model. The first definition coincides with the model proposed in the original work by Erdős and Rényi (1959). Under this definition, the Erdős-Rényi model places a uniform distribution over graphs with the same number of vertices and edges. The second definition corresponds to a model proposed around the same time by Gilbert (1959). This variant of the Erdős-Rényi model places a distribution over graphs with n vertices that can be obtained by independently connecting two vertices with probability $p \in (0, 1)$. In the network literature, the Gilbert (1959) variant is commonly referred to as the Erdős-Rényi model because it is easier to analyze mathematically. We adopt this definition in this article.

Beyond the Erdős-Rényi model, the most commonly used random graph model places a uniform distribution over graphs with a prescribed degree sequence $\mathbf{d} = (d(1), \dots, d(n))$. This model often serves as the

null distribution in statistical hypothesis tests. For example, Milo et al. (2002) determined specific network motifs are more prevalent in real-world networks than random graphs with the same degree sequence. Although there are a variety of Monte Carlo methods designed to sample uniformly from random graph models with a fixed degree sequence, a popular algorithm called the configuration model is useful for understanding disease spread. The configuration model begins by assigning $d(i)$ stubs to each vertex i . Then two stubs are connected uniformly at random to form an edge. This procedure proceeds on the remaining stubs until all stubs are connected. Note that this algorithm allows for self-loops and multi-edges; however, they represent a small proportion of edges. For this reason, one loses little by discarding them when the number of edges is large.

When studying the asymptotic properties of the configuration model, one often assumes that a known parametric distribution generates the degree sequence (Molloy & Reed, 1995, 1998). For example, the Erdős-Rényi model has a Binomial($n - 1, p$) degree distribution, which is often approximated by a Poisson distribution when n is large. A line of network science research involves fitting parametric distributions to the degree sequence of real-world networks to determine more realistic random graph models.

Scale-Free Networks

In the network literature, one of the most pervasive degree distributions used to mimic real-world networks is the power-law distribution, i.e., $f_d \propto d^{-\gamma}$ where $d > 0$ and $\gamma > 0$. So much so that networks with a power-law degree distribution have garnered the name ‘scale-free’ networks. The scale-free term comes from the scale-free property of the power-law distribution: $f_{a \cdot d} = a^{-\gamma} f_d \propto f_d$. In other words, the shape of the degree distribution is the same regardless of the input’s scale. Typically, γ is taken between two and three. In recent years, researchers have applied rigorous goodness-of-fit tests to networks that were conjectured to be scale-free (Clauset et al., 2009; Khanin & Wit, 2006). These studies found that a pure power-law is quite uncommon, and the degree distribution is typically just heavy-tailed. For this reason, one often restricts the power-law distribution to take values on a finite range of degrees where the power-law provides a good approximation (Pastor-Satorras & Vespignani, 2002).

Preferential Attachment

The preferential attachment (PA) model is a famous mathematical network model used to generate scale-free networks. Unlike random graph models, the preferential attachment model is part of an area of mathematical network modeling that tries to construct realistic networks based on a plausible generation mechanism. The preferential attachment mechanism mimics the ‘rich get richer’ phenomenon. The PA model, proposed by Barabási and Albert (1999), builds up a network sequentially over time. Starting with a graph with node set V_t and edge set E_t at time t , one forms a new graph at time $t + 1$ by adding a vertex with m new edges. These edges are connected to existing vertices in E_t with probability proportional to their degree. This procedure results in new vertices tending to attach to existing high-degree vertices. In the original work, the authors showed that as t goes to infinity, the degree distribution approaches a power-law with exponent $\gamma = 3$.

Small-World Models

The previous models do not capture many high-order network characteristics prevalent in real-world networks. In particular, they lack the small-world property, which states that many networks have small average distances between nodes and high transitivity. Formally, a network model exhibits small-world behavior

if its diameter scales as $\mathcal{O}(\log(n))$ while the clustering coefficient, $\text{cl}(G)$, remains large. In particular, the Erdős-Rényi and preferential attachment models do not exhibit the small-world property since their clustering coefficients scale as n^{-1} (Bollobás & Riordan, 2003; Kolaczyk, 2017). Watts and Strogatz (1998) proposed a famous mathematical model that exhibits small-world behavior. In its basic form, to generate a graph, the model begins by arranging the nodes on a regular ring lattice, where each node is connected to r neighbors on each side. Then, for each edge, the model rewires the edge with probability p to another node chosen uniformly at random while avoiding self-loops and repeated edges. The resulting graphs tend to have high transitivity while maintaining small average distances.

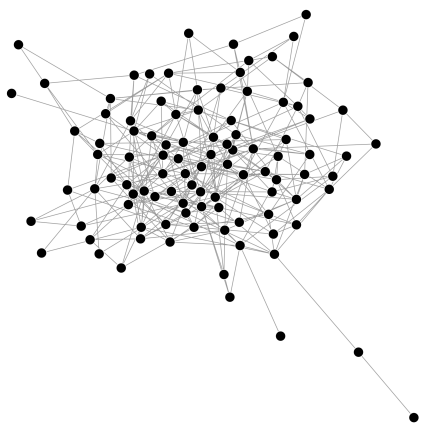
An Empirical Comparison of Mathematical Network Models

We end this section by demonstrating how the network characteristics described in Section 1.1.2 can distinguish the previous mathematical network models. Figure 1.1 visualizes an (a) Erdős-Rényi, (b) 6-regular, (c) preferential attachment, and (d) small-world network with the Fruchterman-Reingold layout (Fruchterman & Reingold, 1991). To focus on structural differences, we fixed the number of nodes to $n = 100$ and chose the model parameters so that each had an average degree of roughly six. Although it is clear from the visualization that the networks are structurally different, it is difficult to pinpoint how they differ. For this, we use the network statistics described in Section 1.1.2.

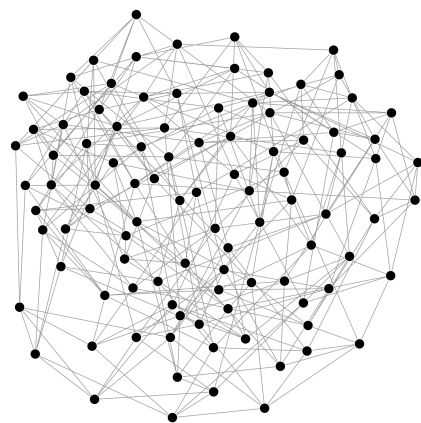
Table 1.1 summarizes the median of five network statistics calculated from 100 simulations of an Erdős-Rényi, 6-regular, preferential attachment, and small-world network with the same settings as visualized networks but with $n = 10,000$ nodes. We also included a static version of a real-world contact network ($n = 236$) analyzed as a dynamic network in Chapter 4. The 6-regular and Erdős-Rényi models serve as baselines with small degree heterogeneity and little higher-order structure. The PA model has the most heavy-tailed degree distribution with a median maximum degree of 211. The small-world network is the only network model that contains a large amount of transitivity. Note that the distance between nodes is quite small for all networks. All of these models contain minimal degree assortativity, which is highly prevalent in the real primary school network. Ensuring that we capture the relevant structures found in the network under investigation is crucial for network modeling. In this regard, statistical network models provide a powerful tool, which we review in the next section.

	Max. Degree	Assortativity	Transitivity	Avg. Path Length	Diameter
Erdős-Rényi	17(1.1)	0.000(0.005)	0.006(0.001)	5.35(0.01)	10(0.4)
6-Regular	6(0)	NA	0.0042(0.0001)	5.61(0.0006)	8(0)
PA	211(35)	-0.006(0.004)	0.003(0.0002)	4.46(0.02)	7(0.2)
Small-World	11(0.6)	-0.001(0.006)	0.308(0.003)	7.05(0.03)	11(0.5)
Primary School	36	0.21	0.47	2.94	7

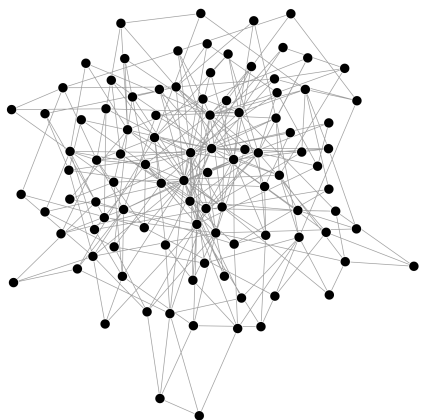
Table 1.1: Summary statistics for four mathematical network models (Erdős-Rényi, 6-regular, preferential attachment, and small-world) as well as the real-world primary school network described in Chapter 4. To focus on topological differences, we fixed the number of nodes $n = 10,000$, the average degree to roughly 6, and the number of edges to the same order of magnitude ($\sim 30,000$) for all simulated networks. For the simulations, the reported statistics are the median value over 100 different samples. Values in parentheses are the standard deviation over these 100 samples. Note that the assortativity of a 6-regular network is undefined since it has zero degree variance.



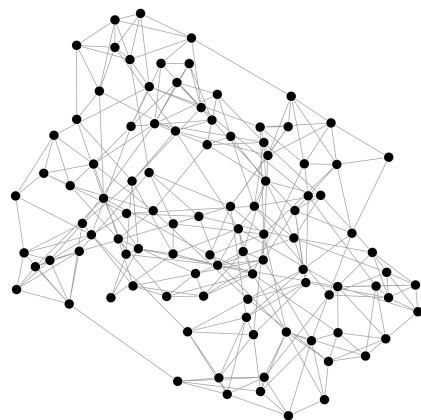
(a) Erdős-Rényi



(b) 6-regular



(c) Preferential attachment



(d) Small-world

Figure 1.1: Visualization of an (a) Erdős-Rényi, (b) 6-regular, (c) preferential attachment, and (d) small-world network. Each network has $n = 100$ nodes and an average degree of roughly six. The visualizations use the Fruchterman-Reingold layout (Fruchterman & Reingold, 1991).

1.2.2 Statistical Network Models

Statistical network models posit that the adjacency matrix, $\mathbf{Y}$, and its elements are random variables with complex interdependencies. We differentiate statistical network models by the statistical dependencies they place on the adjacency matrix. As their name implies, statistical network models are suited for typical tasks of statistical analysis such as significance testing, model comparison, and uncertainty quantification. Furthermore, they are often highly flexible, which allows them to better capture the higher-order network structures described in Section 1.1.2. However, unlike mathematical network models, statistical network models are often analytically intractable. In the following sections, we provide an overview of the common types of statistical network models.

ERGMs

Exponential random graph models (ERGMs) or p^* models (Frank & Strauss, 1986; Holland & Leinhardt, 1981; Robins et al., 2007; Wasserman & Pattison, 1996) posit that the adjacency matrix is drawn from a canonical exponential family, i.e.,

$$\mathbb{P}(\mathbf{Y} = \mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) = k(\boldsymbol{\theta}, \mathbf{X})^{-1} \exp(\boldsymbol{\theta}^T g(\mathbf{y}, \mathbf{X})), \quad (1.5)$$

where $\mathbf{X}$ is a collection of exogenous covariates, $g(\mathbf{y}, \mathbf{X})$ is a p -dimensional vector of statistics, $\boldsymbol{\theta}$ is a p -dimensional vector of coefficients, and $k(\boldsymbol{\theta}, \mathbf{X})$ is the normalizing constant.

The main benefit of ERGMs are their ability to test the prevalence of specific characteristics in a network. These tests require choosing network statistics $g(\mathbf{y}, \mathbf{X})$ that plausibly describe the network's structure. For example, to test the presence of transitivity in a social network, one often includes some form of triangle counts. Other commonly used statistics include the number of edges, the degree sequence, and the number of k -stars. More recently, Hunter and Handcock (2006) found that geometrically weighted degree, edgewise shared partner, and dyad-wise shared partner statistics adequately fit real-world networks. See Snijders et al. (2006) or Hunter et al. (2008) for a comprehensive overview of commonly used network statistics.

The coefficients, $\boldsymbol{\theta}$, have a simple interpretation as a log-odds. To see this, one defines the so called 'change statistics':

$$\delta_g(y_{ij}) = g(\mathbf{y}_{ij}^+, \mathbf{X}) - g(\mathbf{y}_{ij}^-, \mathbf{X}), \quad (1.6)$$

where $\mathbf{y}_{ij}^+$ and $\mathbf{y}_{ij}^-$ represent the network realized by fixing $y_{ij} = 1$ or $y_{ij} = 0$, respectively, while keeping the rest of the observed network unchanged. In other words, $\delta_g(y_{ij})$ is the change in the network statistics when we flip $y_{ij} = 0$ to $y_{ij} = 1$ while keeping the rest of the observed network and covariates fixed. It can be shown that

$$\text{logit} [\mathbb{P}(Y_{ij} = 1 \mid \mathbf{Y}_{ij}^c = \mathbf{y}_{ij}^c)] = \boldsymbol{\theta}^T \delta_g(y_{ij}), \quad (1.7)$$

where $\mathbf{Y}_{ij}^c$ represents the rest of the network other than the single random variable Y_{ij} . This expression shows that each coefficient in $\boldsymbol{\theta}$ can be interpreted as its statistic's contribution, per unit increase in the statistic, to the log-odds of an individual tie conditioned on the rest of the network. For additional exposition on the interpretation of ERGM coefficients, see Goodreau et al. (2008).

Despite their interpretability, ERGMs can have trouble fitting observed networks due to model degeneracy (Handcock, 2003). Model degeneracy refers to the phenomenon where an inferred distribution places most of its probability mass on a small subset of the probability space. In particular, ERGM estimates often place most of their mass on either empty or complete networks. Such degeneracy can lead to poor estimates or

issues of convergence of the estimating algorithms.

Since their introduction, various extensions of ERGMs have been proposed. Krivitsky and Morris (2017) extended the ERGM to networks sampled under an egocentric design. Extensions to dynamic networks include the temporal ERGM (TERGM) (Hanneke et al., 2010) and the separable temporal ERGM (STERGM) (Krivitsky & Handcock, 2013).

Stochastic Block Models

Stochastic block models (SBMs) (Holland et al., 1983; Nowicki & Snijders, 2001) are among the most well-studied network models in the statistics literature. The SBM posits that each actor belongs to one of K communities. Conditioned on these community memberships, the entries of the adjacency matrix are independent Bernoulli random variables:

$$Y_{ij} \sim \text{Bernoulli}(B_{pq}), \quad (1.8)$$

where actor i is a member of community p , actor j is a member of community q , and B is a $K \times K$ matrix with entries indicating the probability of forming an edge between communities p and q . Note that the community memberships are unknown, so they must be inferred from the data. It is possible to analyze the SBM as a mixture of Erdős-Rényi graphs. Therefore, many of the properties of the random networks sampled from an SBM are analytically expressible in terms of the underlying model parameters (Daudin et al., 2008). This analytical tractability makes the SBM a compelling alternative to standard mathematical network models.

Various extensions to the SBM have been proposed. The mixed membership stochastic block model (MMSBM) (Airoldi et al., 2008) and the overlapping stochastic block model (OSBM) (Latouche et al., 2011) allow vertices to be a member of more than one community. The class of degree corrected stochastic block models aims to produce SBMs with more heterogeneous degree distributions (Karrer & Newman, 2011). Finally, the SBM was extended to dynamic networks by Matias and Miele (2017).

Latent Space Models

Latent space models (LSMs) are an alternative to ERGMs designed to capture the high levels of transitivity, reciprocity, and assortativity found in real-world networks. They do not suffer from the same model-degeneracy of ERGMs while still providing a highly flexible statistical network model. Furthermore, LSMs contain certain SBMs as special cases. Introduced by P. D. Hoff et al. (2002), LSMs embed a network's actors into a d -dimensional latent metric space. The two most popular choices of metric space are the Euclidean space (the distance model) or a hypersphere (the projection model). The likelihood of the model assumes that closeness in the latent space increases the probability that two actors form an edge in the observed network. For this reason, one interprets the proximity of two actors in the latent space as an indication that they have similar latent characteristics. This property of the LSM mimics the assortative nature of observed characteristics in real-world networks.

Formally, the latent space model assumes edges form independently when conditioned on the actor's latent positions $\{\mathbf{Z}_i\}_{i=1}^n$ and the exogenous dyadwise covariates $\{\mathbf{X}_{ij}\}$:

$$\mathbb{P}(\mathbf{Y} = \mathbf{y} \mid \{\mathbf{X}_{ij}\}, \{\mathbf{Z}_i\}_{i=1}^n, \alpha, \beta) = \prod_{i < j} \mathbb{P}(Y_{ij} = y_{ij} \mid \mathbf{X}_{ij}, \mathbf{Z}_i, \mathbf{Z}_j, \alpha, \beta), \quad (1.9)$$

where

$$\text{logit} [\mathbb{P}(Y_{ij} = 1 \mid \mathbf{X}_{ij}, \mathbf{Z}_i, \mathbf{Z}_j, \alpha, \beta)] = \alpha + \beta^T \mathbf{X}_{ij} - \|\mathbf{Z}_i - \mathbf{Z}_j\|_2. \quad (1.10)$$

In the previous expression, α is a scalar intercept and β is a vector of coefficients. From this, one can see that the latent positions act as multiplicative random effects in a dyadwise logistic regression model. Note that Equation (1.10) uses the distance model formulation. The equivalent expression for the projection model replaces the negative Euclidean distance with the projection operator $\mathbf{Z}_i^T \mathbf{Z}_j / \|\mathbf{Z}_j\|_2$. The projection model is suited to modeling reciprocity in directed networks because of the asymmetry in the metric. Also, all models exhibit transitivity since the latent metric space incorporates the triangle inequality.

There have been numerous extensions to the original latent space model. The latent position clustering model (LPCM) of Handcock et al. (2007) added community structure to the latent space through a Gaussian mixture model. Individual additive random effects were incorporated to explicitly model degree heterogeneity in P. D. Hoff (2005) and Krivitsky et al. (2009). Young and Scheinerman (2007) introduced the random dot product graph, which removes the logit-link function and utilizes the dot-product, i.e., $\mathbb{P}(Y_{ij} = 1 \mid \mathbf{Z}_i, \mathbf{Z}_j) = \mathbf{Z}_i^T \mathbf{Z}_j$. Many asymptotic results and applications exist for random dot product graphs, see Athreya et al. (2018) for a thorough survey. Sarkar and Moore (2005) and Sewell and Chen (2015b) proposed extensions of the LSM to dynamic networks.

1.3 Overview of Contributions

Chapter 2 begins by considering the problem of quantifying local variable importance in nonparametric regression. As mentioned previously, many modern data science problems in personalized medicine, policy-making, and bioinformatics are concerned with interpreting the results of seemingly black-box estimators. For example, personalized medicine aims to prescribe personalized treatments based on statistical models trained on patients' individual characteristics (Laber & Zhao, 2015). If these treatment rules are to inform clinical practice and scientific understanding, it is crucial that these predictions are explainable. To address this need, we develop a new locally adaptive kernel smoothing method called *dimension reduction forests* (DRFs). As a start, we consider random forests (Breiman, 2001), a popular class of nonparametric estimators composed of an ensemble of randomized decision trees (Breiman et al., 1984). However, despite random forest's popularity, they only provide variable importance in a global sense. To develop a locally adaptive method, we pair the random forest kernel (Scornet, 2016) with local sufficient dimension reduction (K.-C. Li, 1991) to develop the proposed estimator. We empirically show that DRFs are adaptive to a regression function's local structures, inherit the accuracy of traditional random forests, and remain very interpretable. In particular, dimension reduction forests' adaptivity to local low-dimensional spaces allows DRFs to estimate a type of local variable importance we call *local principal directions* and to achieve significant accuracy gains over competing methods.

The remainder of the dissertation develops methodology for statistical network analysis. In particular, Chapter 3 introduces a Bayesian nonparametric method to infer discrete changes in community structure in dynamic networks. Specifically, this chapter considers the problem of dynamic community detection in which the number of communities is unknown and allowed to change over time. This work was inspired by the time-varying heterogeneous community structure found in international alliance networks during the Cold War. These networks are characterized by a varying number of regional alliances with a few super-powers (e.g., the US and the USSR) influencing multiple blocs. Such heterogeneity within the community structure is ignored by popular stochastic block models but is naturally described by the latent space approach to

network modeling (Handcock et al., 2007; P. D. Hoff et al., 2002). The model developed in this chapter is based on the dynamic latent space clustering model proposed by Sewell and Chen (2017). However, the previous model assumes the number of communities is known and fixed over time. Such assumptions are unrealistic for many time-varying relational data sets and do not capture essential phenomena such as community merging and splitting. As a solution, I develop the *hierarchical Dirichlet process latent position cluster model* (HDP-LPCM) that addresses the following two open problems: (1) it allows the number of communities to change over time, and (2) it provides accurate uncertainty quantification over an unknown number of communities. Methodological advances included a new nonparametric prior based on the hierarchical Dirichlet process, an efficient block Gibbs sampler that uses the forward-backward algorithm to speed up mixing of the community labels, and a new posterior summarization method that empirically improves community detection performance.

Chapter 4 tackles the challenging problem of modeling dynamic multilayer networks. Dynamic multilayer networks are defined as several types of dyadic relations, called layers, observed over a sequence of time points. These networks are a prevalent form of relational data with applications in epidemiology, sociology, biology, and other fields. For instance, social networks contain multiple correlated relations that jointly evolve over time: friendship, vicinity, coworker-ship, partnership, and others. Existing methods include co-evolving stochastic actor oriented models (Snijders et al., 2013), network multilinear autoregressive models for real-valued networks (P. D. Hoff, 2015), and a Bayesian nonparametric latent space model proposed in Durante et al. (2017). However, these models do not capture the mixture of assortative and disassortative relations found in multilayer networks and their inference is computationally expensive. In this chapter, I develop a new Bayesian latent space model for dynamic multilayer networks. The key feature of our model is its ability to identify common time-varying structures shared by all layers while also accounting for layer-wise variation and degree heterogeneity. We established the identifiability of the model’s parameters and developed a novel variational inference approach to estimate the model’s posterior, which maintains the latent variable’s temporal dependencies for more accurate uncertainty quantification and scales to networks previously intractable to dynamic latent space models. We applied the model to two problems: discerning regional conflicts in a data set of international relations and quantifying infectious disease spread throughout a primary school. Highlights of the analyses include detecting regional conflicts in the international relations networks and detecting times during the school day with increased infectious disease spread.

Finally, Chapter 5 proposes a principled Bayesian approach to dimension selection in static and dynamic network eigenmodels. Determining the latent space’s dimension, d , is a persistent problem in practical applications of LSMs, including those in the previous chapters. An appropriate choice is crucial for balancing the bias-variance trade-off and ensuring the model captures all of a network’s essential properties. Previous approaches include selection through information criteria (Spiegelhalter et al., 2002; Watanabe, 2010), data splitting strategies such as cross-validation (P. D. Hoff, 2008; L. Li et al., 2020), or through priors that reduce the model’s sensitivity to the choice of d (Durante & Dunson, 2014; Guha & Rodriguez, 2021; Guhaniyogi & Rodriguez, 2020). However, these approaches are either too computationally expensive or lack theoretical guarantees. In this chapter, I present a new Bayesian LSM for dynamic networks that solves these issues. The model is accompanied by a new ordered spike-and-slab prior that provides posterior inference over the latent space dimension and satisfies several appealing theoretical properties. In particular, the prior places a soft-ordering constraint on the model parameters that overcomes identifiability issues in the model’s likelihood. Furthermore, we demonstrate that this prior results in the static model’s posterior concentrating on low-dimensional latent spaces near the truth. These models are accompanied by a novel parameter

expansion scheme that allows for efficient Markov chain Monte Carlo estimation using a Metropolis-within-Gibbs sampler with Hamiltonian Monte Carlo proposals. We demonstrate our approach’s versatility and consistent dimension selection on simulated and real networks.

Chapter 2

Dimension Reduction Forests

2.1 Introduction

Random forests (Breiman, 2001) have repeatedly proven themselves as an effective supervised learning method. Random forests’ competitive predictive accuracy, even when the problem is non-linear, high-dimensional, or involves complex interaction effects has resulted in their wide-scale application.¹ Also, unlike many black-box methods, random forests provide an interpretable global variable importance measure for each predictor variable, which is of a broad interest in scientific problems in bioinformatics (Diaz-Urriarte & de Andrés, 2006), ecology (Prasad et al., 2006), and personalized medicine (Laber & Zhao, 2015).

The success of random forests has sparked a desire to improve upon the original algorithm’s shortcomings. A growing effort attempts to understand the statistical properties of random forests (Athey et al., 2019; Cui et al., in press; Mentch & Hooker, 2016) such as the consistency and asymptotic normality of their predictions. Also, the core random forest methodology has been extended to numerous frameworks, including quantile regression (Meinshausen, 2006), survival analysis (Ishwaran et al., 2008), and causal inference (Athey et al., 2019). Furthermore, the tree-growing process has been modified to allow for additional randomness (Geurts et al., 2006), linear combination splits (Breiman, 2001; Menze et al., 2011), and non-uniform feature selection (Amaratunga et al., 2008). Lastly, the out-of-bag global variable importance measure has been extended to better account for correlated features through conditional independence tests (Strobl et al., 2008). For a comprehensive overview of random forest research developments, we refer the reader to the review paper by Biau and Scornet (2016).

In this chapter, we develop *dimension reduction forests* (DRFs): a new method for nonparametric regression that also quantifies local variable importance by using methods from sufficient dimension reduction (SDR) (B. Li, 2018; K.-C. Li, 1991). Fields such as personalized medicine, policy-making, and bioinformatics are increasingly relying on these local assessments. For example, the ability to tailor medical treatment to patients based on their genetic makeup can drastically decrease mortality. Currently, random forests have limited value in these applications because they lack a natural measure of local variable importance. To overcome this deficiency, we take the perspective of random forests as adaptive kernel methods. We pair random forests with sufficient dimension reduction to estimate a nonparametric kernel that adapts to the regression function’s local contours. We then leverage this adaptivity to estimate a type of local variable

¹This chapter is based on research published as: Loyal, J. D., Zhu, R., Cui, Y., and Zhang, X. (in press). Dimension reduction forests: local variable importance using structured random forests. *Journal of Computational and Graphical Statistics*.

importance we call *local principal directions*. The result is a powerful model-free predictive method that is more accurate than naively combining random forests with global SDR methods. Before introducing our approach, we motivate how its local adaptivity overcomes certain deficiencies in traditional random forests.

2.2 Motivation

Throughout this chapter, we consider a general regression problem in which a continuous response $Y \in \mathbb{R}$ is predicted as a function of p covariates $\mathbf{X} = (X_1, \dots, X_p)^T \in \mathbb{R}^p$.

2.2.1 Adapting the Random Forest Kernel to Local Structure

A random forest (RF) is an ensemble of M randomized decision trees. The random forest kernel (Scornet, 2016) between two points $\mathbf{x}_0, \mathbf{x}_1 \in \mathbb{R}^p$ is

$$K_{\text{RF}}(\mathbf{x}_0, \mathbf{x}_1) = \frac{1}{M} \sum_{m=1}^M \sum_{u=1}^{\ell_m} \mathbb{1}_{\{\mathbf{x}_0 \in A_u^m\}} \mathbb{1}_{\{\mathbf{x}_1 \in A_u^m\}}, \quad (2.1)$$

where ℓ_m is the number of leaf nodes in the m th tree and $\{A_1^m, \dots, A_{\ell_m}^m\}$ contains the m th tree's leaf nodes with each $A_u^m \subseteq \mathbb{R}^p$. This kernel equals the empirical probability that $\mathbf{x}_0$ and $\mathbf{x}_1$ share a leaf node within the forest. Our goal is to extract a local variable importance measure from this kernel; however, this measure is only meaningful if the kernel reflects the regression function's local structure. The RF kernel does not exhibit such adaptivity.

To be more concrete, consider the following non-linear regression model of a univariate response on a bivariate predictor: We draw $\mathbf{X}_1, \dots, \mathbf{X}_n$ independently from the uniform distribution on $[-3, 3]^2$, with response

$$Y = 20 \max \left\{ e^{-2(X_1 - X_2)^2}, 2e^{-0.5(X_1^2 + X_2^2)}, e^{-(X_1 + X_2)^2} \right\} + \varepsilon, \quad \varepsilon \stackrel{\text{iid}}{\sim} N(0, 1), \quad (2.2)$$

and our goal is to extract local information about the regression (mean) function $m(\mathbf{x}_0) = \mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}_0]$. Figure 2.1 displays the contour lines of this function, which depend on three subspaces throughout the function's domain: $\mathcal{S}_1 = \text{span}(\mathbf{e}_1 - \mathbf{e}_2)$ (bottom-left and top-right), $\mathcal{S}_2 = \text{span}(\mathbf{e}_1 + \mathbf{e}_2)$ (top-left and bottom-right), and $\mathcal{S}_3 = \text{span}(\{\mathbf{e}_1, \mathbf{e}_2\})$ (center), where $\mathbf{e}_1$ and $\mathbf{e}_2$ are the standard basis. An ideal kernel should reflect the local structure by extending its weights along either the orthogonal complement of the one-dimensional subspaces, $\mathcal{S}_1$ and $\mathcal{S}_2$, or the contour lines within the two-dimensional subspace, $\mathcal{S}_3$. The random forest kernel does not exhibit this behavior.

Figure 2.1 displays the random forest kernel estimated on data from the regression function in Equation (2.2). The RF kernel cannot reflect the local structure because the leaf nodes, $\{A_u^m\}_{u=1}^{\ell_m}$, form an axis-aligned partition of the input space. Indeed, the RF kernel exhibits an axis-aligned plus shape that does not exploit the local dimension reduction structure. As a first encouraging result, our proposed dimension reduction forest's induced kernel, displayed in Figure 2.1, warps around the contour lines, improving its performance and accurately reflecting the regression function's local structure. We develop the dimension reduction forest kernel in Section 2.3.

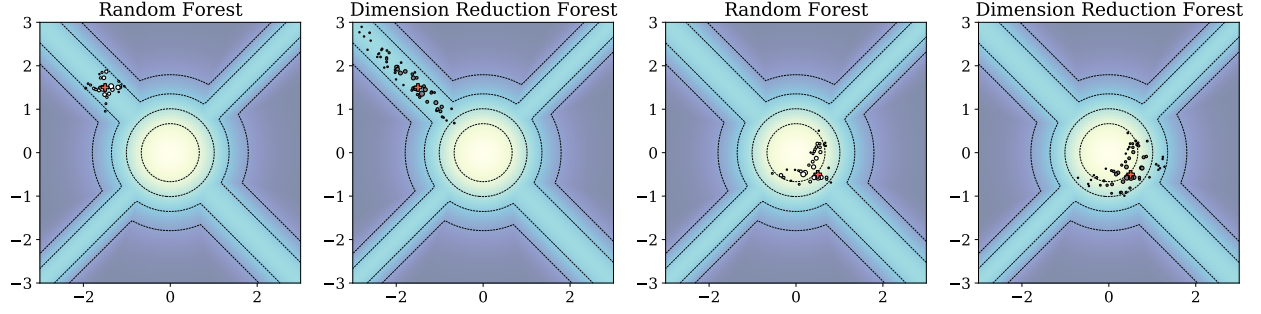


Figure 2.1: Kernels generated from a random forest and a dimension reduction forest on the problem defined in Equation (2.2). The target prediction point $\mathbf{x}_0$ is displayed with a plus. The size of each circle is proportional to the kernel's weight. The contours of the true regression function are shown as black dashed curves.

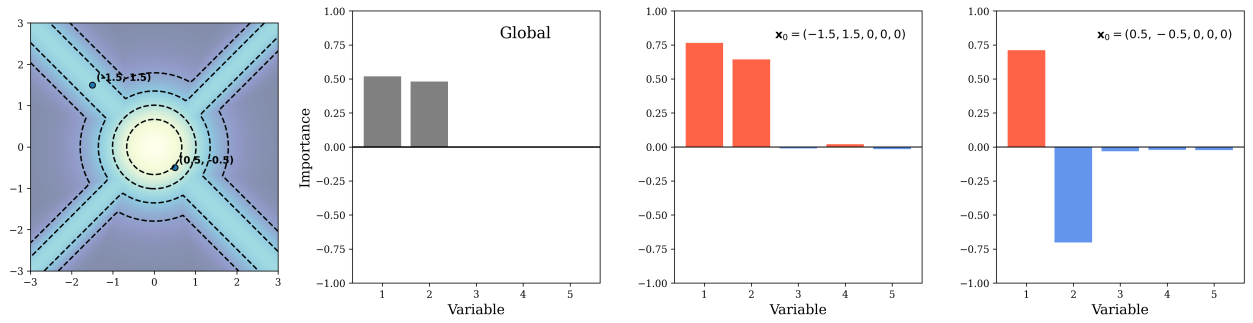


Figure 2.2: Variable importance measures produced by a dimension reduction forest.

2.2.2 Local Variable Importance

Random forests' popularity is partly due to their ability to produce interpretable global variable importance measures; however, extracting local variable importance is an active research area. As previously mentioned, estimation and inference at the local level are becoming increasingly important, with personalized medicine applications such as inferring individualized therapeutic treatments. Our primary reason for constructing a kernel that can reflect local structure is to uncover such local insights.

As an illustration, we again consider the problem outlined in Equation (2.2); however, now we draw $\mathbf{X}_1, \dots, \mathbf{X}_n$ independently from a $U[-3, 3]^5$ so that there are three additional uninformative covariates. Random forests' global importance measure can indicate that X_1 and X_2 are informative; however, it cannot reveal the feature's heterogeneous contributions throughout the input space.

Figure 2.2 displays the two levels of variable importance provided by a dimension reduction forest: permutation-based global variable importance and our proposed local variable importance measured at two query points. Matching a standard random forest, the global permutation-based importance assigned by a DRF to a feature is the decrease in mean squared error when the values of that feature are randomly permuted in the out-of-bag samples (Breiman, 2001). Absent from traditional random forests, DRF's local variable importance can be found in Algorithm 2.2 of Section 2.3. The DRF's global importance identifies the two significant predictors and the DRF's local importance determines the variables' heterogeneous contribution. At $\mathbf{x}_0 = (-1.5, 1.5, 0, 0, 0)^T$, the dimension reduction forest correctly indicates that simultaneously increasing or decreasing both X_1 and X_2 influences changes in the regression function. In contrast, at $\mathbf{x}_0 = (0.5, -0.5, 0, 0, 0)^T$, the DRF indicates that simultaneously increasing X_1 while decreasing X_2 , or vice versa, is influential. Unlike the global importance measure, the DRF's local importance suggests two drastically different interventions depending on the query point.

2.3 Dimension Reduction Forests

Now, we formally introduce dimension reduction trees (DRTs) and dimension reduction forests (DRFs). We describe the DRF algorithm in Section 2.3.1, connect the DRF splitting rule to an approximation of a locally adaptive kernel in Section 2.3.2, and outline our method for extracting local variable importances from the induced DRF kernel in Section 2.3.3.

2.3.1 The Splitting Rule for Dimension Reduction Forests

Dimension reduction forests are random forests composed of dimension reduction trees that use sufficient dimension reduction (SDR) techniques (B. Li, 2018; K.-C. Li, 1991) to approximate a locally adaptive kernel. Dimension reduction trees are built using a modified classification and regression tree (CART) algorithm (Breiman et al., 1984) with linear combination splitting rules. Specifically, a regression tree $T(\mathbf{x}_0) = \sum_{u=1}^{\ell} \mu_u \mathbb{1}_{\{\mathbf{x}_0 \in A_u\}}$, with ℓ leaf nodes $A_u \subseteq \mathbb{R}^p$ and leaf means $\mu_u \in \mathbb{R}$, is recursively built using a training data set $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)\}$ as follows. At any current leaf node A , we search for a splitting rule $\mathbb{1}_{\{\beta_j^T \mathbf{x} \leq c\}}$ for $\beta_j \in \mathbb{R}^p$ and $j \in \{1, \dots, d\}$, for some $d \leq p$. We set the possible values of the threshold c to all unique values of $\beta_j^T \mathbf{X}_i$ within node A . We then refine the current leaf node into two child nodes: $A_L = A \cap \{\beta_j^T \mathbf{x} \leq c\}$ and $A_R = A \cap \{\beta_j^T \mathbf{x} > c\}$ based on a certain splitting criterion. In the regression

Given a training data set $\mathcal{D}_n = \{(\mathbf{X}_i, Y_i)\}_{i=1}^n$, do the following:

1. Draw M bootstrap samples each with sample size n from $\mathcal{D}_n$.
2. For the m th bootstrap sample, where $m \in \{1, \dots, M\}$, fit one DRT model $\hat{T}_m$ using the following rules:
 - (a) *Optional variable screening*: At an internal node A , select the top m_{try} features that minimize Equation (2.3) based on their corresponding axis-aligned splitting rules. Otherwise, set $m_{try} = p$.
 - (b) Estimate the leading sufficient dimension reduction direction $\hat{\beta} \in \mathbb{R}^p$ using SIR ($\hat{\beta}_{\text{SIR}}$) and SAVE ($\hat{\beta}_{\text{SAVE}}$) with the features selected in step (a) and the samples in A as input.
 - (c) Project all observations onto the reduced space by calculating $[\hat{\beta}_{\text{SIR}}, \hat{\beta}_{\text{SAVE}}]^T \mathbf{X}_i$ for each sample i in A . Construct a univariate splitting rule $\mathbb{1}_{\{\hat{\beta}^T \mathbf{x} \leq c\}}$ by minimizing Equation (2.3) over the two features in the reduced data set.
 - (d) Recursively apply steps (a)–(c) on each child node until the node’s sample size is smaller than a pre-specified value $n_{\min}$.
 - (e) For each leaf node, set $\mu_u = \bar{Y}_{A_u} = |\{i : \mathbf{X}_i \in A_u\}|^{-1} \sum_{\{i : \mathbf{X}_i \in A_u\}} Y_i$.
3. Average the M trees to obtain the final model, $\hat{f}(\mathbf{x}) = M^{-1} \sum_{m=1}^M \hat{T}_m(\mathbf{x})$, with associated kernel $K_{\text{DRF}}(\mathbf{x}_0, \mathbf{x}_1)$ defined in Equation (2.1).

Algorithm 2.1: Dimension reduction forests.

context, the best split minimizes the sum of squared errors in each child node:

$$\arg \min_{(A_L, A_R) \in \mathcal{C}} \left\{ \sum_{\{i : \mathbf{X}_i \in A_L\}} (Y_i - \bar{Y}_{A_L})^2 + \sum_{\{i : \mathbf{X}_i \in A_R\}} (Y_i - \bar{Y}_{A_R})^2 \right\}, \quad (2.3)$$

where $\mathcal{C}$ is the set of all partitions formed by adding a rule of the form $\mathbb{1}_{\{\beta_j^T \mathbf{x} \leq c\}}$, and $\bar{Y}_{A_L}$ and $\bar{Y}_{A_R}$ are the mean value of Y inside A_L and A_R , respectively. When we restrict the directions to the standard basis vectors, $\beta_j = \mathbf{e}_j$ for $j \in \{1, \dots, p\}$, the algorithm reduces to the axis-aligned splits used in the original CART algorithm. Algorithm 2.1 outlines the procedure for building a dimension reduction forest.

What differentiates DRFs from other random forest variants are splitting directions, β , estimated through inverse regression, a type of SDR. Briefly, sufficient dimension reduction aims to find a matrix $B \in \mathbb{R}^{p \times d}$, where $d \leq p$, such that the conditional distribution of Y given $\mathbf{X}$ is the same as Y given $B^T \mathbf{X}$. In regression, this implies that

$$Y = g(B^T \mathbf{X}, \varepsilon) = g(\beta_1^T \mathbf{X}, \dots, \beta_d^T \mathbf{X}, \varepsilon) \quad (2.4)$$

for an unspecified link function $g(\cdot)$ and random noise ε . This model is equivalent to assuming $Y \perp \mathbf{X} \mid B^T \mathbf{X}$. The column space of B , denoted by $\text{span}(B)$, is called a dimension reduction (DR) subspace. Note that a DR subspace of dimension p always exists because $Y \perp \mathbf{X} \mid \mathbf{X}$ always holds. The intersection of all DR subspaces, assuming itself is also a DR subspace, is called the central subspace and denoted by $S_{Y|\mathbf{X}}$. The existence of $S_{Y|\mathbf{X}}$ is often guaranteed by mild conditions. For example, $S_{Y|\mathbf{X}}$ always exists when the covariates’ support is a convex set (Proposition 6.4 in Cook (1998)); then by construction it is unique and is the smallest DR subspace. For ease of presentation, we assume the central subspace exists throughout this chapter. However, the proposed method is naturally applicable with respect to any DR subspace, even when the central subspace does not exist.

Since SDR does not constrain the form of $g(\cdot)$, it is a powerful tool for designing flexible linear combination

splitting rules. This flexibility differentiates our approach from existing oblique forests (Menze et al., 2011; Rainforth & Wood, 2015), which impose restrictive linear constraints on the regression surface. Note that DRTs do not necessarily impose any global dimension reduction, such as in Equation (2.4). They only assume meaningful local dimension reduction structures develop as one focuses on more local regions throughout the tree building process, i.e., $Y \perp \mathbf{X} \mid B_A^T \mathbf{X}$ for $\mathbf{X} \in A \subseteq \mathbb{R}^p$ where $B_A \in \mathbb{R}^{p \times d}$. Furthermore, we posit that using DRTs within the RF kernel improves its local statistical efficiency by aligning its weights with the local Hessian. We elaborate on this point in the next section.

To estimate the sufficient dimension reduction directions, $\{\beta_1, \dots, \beta_d\}$, we use SIR (K.-C. Li, 1991) and SAVE (Cook & Weisberg, 1991) since the former is efficient for small sample sizes and the latter is more exhaustive. Both methods solve a generalized eigenvalue problem to estimate B , which we overview in Section 2.8.1 for completeness. Note that DRTs select the leading eigenvector instead of searching over the full DR subspace. Such an exhaustive search would be redundant because the eigenvalues associated with each direction already measure their local importance. Furthermore, to avoid choosing between SIR and SAVE, we search over the leading directions estimated by both methods when determining a split. This results in a splitting rule that combines the advantages of both SIR and SAVE and performs better than just applying a single method in each node. Also, a DRF’s computational complexity scales at a comparable rate to traditional random forests and has empirical run times roughly three times slower than an RF over various values of n and p . See Section 2.8.2 for details.

Three problems arise when using traditional SDR methods within dimension reduction forests. First, SIR (SAVE) require at least p samples for inference; however, it is often advisable to set $n_{min} < p$. To accommodate this setting, the DRT searches over axis-aligned splits when the internal node’s sample size is less than p . Although this may affect the forest’s ability to adapt to local dimension reduction structures, we found the degradation negligible in practice. Second, the use of dense linear combination splits limits DRF’s applicability in medium to high-dimensional settings. For this reason, we introduce an optional variable screening step in Algorithm 2.1 that selects covariates based on the minimization of Equation (2.3). Finally, both SIR and SAVE require strong assumptions on the covariates’ joint distribution for valid estimation. In Section 2.4, we elicit conditions under which these assumptions only need to hold at the root node, extending our method’s applicability to any setting valid for SIR or SAVE. Also, DRFs can incorporate categorical variables by excluding them from the SDR estimation and considering them as separate splitting variables. Section 2.8.3 elaborates on this approach.

2.3.2 A Connection to Locally Adaptive Kernel Regression

Here, we further motivate the DRF splitting rule by connecting it to an approximation of a locally adaptive kernel. Specifically, we demonstrate how a kernel estimator can improve its statistical performance in the presence of local dimension reduction structure by adapting to the regression function’s local Hessian. Then we argue that the DRF’s induced kernel may inherit this statistical efficiency because the DRF’s recursive splitting procedure mimics this oracle approach. The latter argument is heuristic and only serves an illustrative purpose; however, the empirical studies in Section 2.5 support the conclusions.

Consider the multivariate kernel $K(\mathbf{X}_i, \mathbf{x}_0) = \phi\left(\sqrt{(\mathbf{X}_i - \mathbf{x}_0)^T H_{\mathbf{x}_0} (\mathbf{X}_i - \mathbf{x}_0)}\right)$, where $\phi(\cdot)$ is a univariate kernel and $H_{\mathbf{x}_0}$ is a positive definite $p \times p$ bandwidth matrix that depends on the prediction point $\mathbf{x}_0$. Under certain regularity conditions, Ruppert and Wand (1994) showed that the bias and variance of the Nadaraya-Watson estimator, $\hat{m}(\mathbf{x}_0) = (\sum_{i=1}^n K(\mathbf{X}_i, \mathbf{x}_0))^{-1} \sum_{i=1}^n K(\mathbf{X}_i, \mathbf{x}_0) Y_i$, conditioned on $\mathbf{X}_1, \dots, \mathbf{X}_n$

are

$$\begin{aligned}\text{Bias}(\hat{m}(\mathbf{x}_0)) &= \mu_2(\phi) \frac{m'(\mathbf{x}_0)^T H_{\mathbf{x}_0} H_{\mathbf{x}_0}^T f'_{\mathbf{X}}(\mathbf{x}_0)}{f_{\mathbf{X}}(\mathbf{x}_0)} + \frac{1}{2} \mu_2(\phi) \text{tr}(H_{\mathbf{x}_0} m''(\mathbf{x}_0)) + o_p(\text{tr}(H_{\mathbf{x}_0})), \\ \text{Var}(\hat{m}(\mathbf{x}_0)) &= \frac{1}{n \det(H_{\mathbf{x}_0})} \|\phi\|_2^2 \frac{\sigma^2(\mathbf{x}_0)}{f_{\mathbf{X}}(\mathbf{x}_0)} \{1 + o_p(1)\},\end{aligned}$$

where $\mu_2(\phi) = \int u^2 \phi(u) du$, $\|\phi\|_2^2 = \int \phi(u)^2 du$, $\sigma^2(\mathbf{x}_0) = \text{Var}(Y \mid \mathbf{X} = \mathbf{x}_0)$, the covariates' density is $f_{\mathbf{X}}(\cdot)$, and $m'(\mathbf{x}_0)$ and $m''(\mathbf{x}_0)$ are the gradient and Hessian of $m(\cdot)$ evaluated at $\mathbf{x}_0$, respectively. The bias's first term, also known as the design bias, is often eliminated by applying a local linear correction (J. Fan & Gijbels, 1996; Friedberg et al., 2021). Therefore, we focus on minimizing the second term while also controlling the variance.

In the presence of local dimension reduction structure, $m(\mathbf{x}_0) = m(B_{\mathbf{x}_0}^T \mathbf{x}_0)$ for some $B_{\mathbf{x}_0} \in \mathbb{R}^{p \times d}$, where $d < p$. In this case, $\text{span}(m''(\mathbf{x}_0)) = \text{span}(m''(B_{\mathbf{x}_0}^T \mathbf{x}_0)) = \text{span}(B_{\mathbf{x}_0} m''(\mathbf{u}_0) B_{\mathbf{x}_0}^T) \subseteq \text{span}(B_{\mathbf{x}_0})$, where $\mathbf{u}_0 = B_{\mathbf{x}_0}^T \mathbf{x}_0$ and the Hessian in the third expression is taken with respect to $\mathbf{u}_0$. This observation implies that the $\text{rank}(m''(\mathbf{x}_0)) = d_0 \leq d$, where we assume $d_0 \neq 0$ for simplicity. According to the bias-variance decomposition given above, one can decrease a kernel estimator's variance without incurring any additional bias by increasing the bandwidth along the $(p - d_0)$ -dimensional orthogonal complement of $\text{span}(m''(\mathbf{x}_0))$, which we denote by $\text{span}(m''(\mathbf{x}_0))^\perp$. Indeed, this can be accomplished by choosing a specific eigendecomposition of $H_{\mathbf{x}_0} = U_{\mathbf{x}_0} \Lambda_{\mathbf{x}_0} U_{\mathbf{x}_0}^T$, where $U_{\mathbf{x}_0}^T U_{\mathbf{x}_0} = I_p$ and $\Lambda_{\mathbf{x}_0}$ is a diagonal matrix. Ignoring the design bias, if one selects $\text{span}(U_{\mathbf{x}_0}) = \text{span}(m''(\mathbf{x}_0))$, the estimator's bias and variance in the new basis are

$$\begin{aligned}\text{Bias}(\hat{m}(\mathbf{x}_0)) &\propto \text{tr} \left\{ \begin{pmatrix} \Lambda_{\mathbf{x}_0}^{(1)} & 0 \\ 0 & \Lambda_{\mathbf{x}_0}^{(2)} \end{pmatrix} \begin{pmatrix} \Gamma_{d_0 \times d_0} & 0 \\ 0 & 0_{(p-d_0) \times (p-d_0)} \end{pmatrix} \right\} = \text{tr}(\Lambda_{\mathbf{x}_0}^{(1)} \Gamma_{d_0 \times d_0}), \\ \text{Var}(\hat{m}(\mathbf{x}_0)) &\propto \det(\Lambda_{\mathbf{x}_0}^{(1)})^{-1} \det(\Lambda_{\mathbf{x}_0}^{(2)})^{-1},\end{aligned}$$

where $\propto$ denotes proportionality between two values, $\Gamma_{d_0 \times d_0}$ is a diagonal matrix of the non-zero eigenvalues of $m''(\mathbf{x}_0)$ and $\Lambda_{\mathbf{x}_0}^{(1)} \in \mathbb{R}^{d_0 \times d_0}$ and $\Lambda_{\mathbf{x}_0}^{(2)} \in \mathbb{R}^{(p-d_0) \times (p-d_0)}$ are also diagonal matrices. The previous two expressions demonstrate that increasing $\Lambda_{\mathbf{x}_0}^{(2)}$, the bandwidth along $\text{span}(m''(\mathbf{x}_0))^\perp$, reduces the variance without altering the asymptotic bias.

Of course, specifying an appropriate rotation matrix, $U_{\mathbf{x}_0}$, and bandwidth matrix, $\Lambda_{\mathbf{x}_0}$, for each sample point $\mathbf{x}_0$ is computationally infeasible. Instead, the dimension reduction forest kernel approximates this procedure by recursively using the leading SDR direction to estimate an appropriate rotation and the splitting criterion in Equation (2.3) to estimate an optimal bandwidth along this direction. We use SDR for the splitting direction because the span of the local Hessian lies in $S_{Y|\mathbf{X}}$. Indeed, $\text{span}(m''(\mathbf{x}_0)) \subseteq \text{span}(B_{\mathbf{x}_0}) \subseteq S_{Y|\mathbf{X}}$.

2.3.3 Local Principal Directions

We propose a heuristic approach to extracting local variable importance from DRFs. Figure 2.1 demonstrates that the DRF kernel concentrates its bandwidth around the regression function's contour lines. The previous section gives intuition for this phenomenon, where we argued that the DRF kernel potentially adapts to the subspace spanned by the local Hessian. Note that when the local gradient is non-zero, we also expect this subspace to align with the gradient because $\text{span}(m'(B_{\mathbf{x}_0}^T \mathbf{x})) = \text{span}(B_{\mathbf{x}_0} m'(\mathbf{u})) \subseteq \text{span}(B_{\mathbf{x}_0})$. In general, given an estimated kernel function, $K_{\text{DRF}}(\mathbf{x}_0, \cdot)$, the subspace of minimum variance about the query point $\mathbf{x}_0$

potentially aligns with the leading eigenvector of $B_{\mathbf{x}_0}$ where $\text{span}(B_{\mathbf{x}_0})$ is a local DR subspace. Formally, we define this direction as the *local principal direction* (LPD) at $\mathbf{x}_0$ or $\text{LPD}(\mathbf{x}_0)$. We estimate the LPD at $\mathbf{x}_0$ using the smallest principal component of the covariates' covariance matrix after centering the covariates around $\mathbf{x}_0$ and weighting them by the dimension reduction forest kernel. Algorithm 2.2 presents the procedure for estimating LPDs. Only the span of each LPD is identifiable, which we interpret as the one-dimensional subspace that most influences $\hat{m}(\mathbf{x}_0)$.

Given a dimension reduction forest $\hat{f}(\mathbf{x})$ estimated with training data set $\mathcal{D}_n = \{(\mathbf{X}_i, Y_i)\}_{i=1}^n$ and a query point $\mathbf{x}_0$, do the following:

1. Calculate the normalized sample weights $w_i = K_{\text{DRF}}(\mathbf{x}_0, \mathbf{X}_i) / (\sum_{k=1}^n K_{\text{DRF}}(\mathbf{x}_0, \mathbf{X}_k))$, $1 \leq i \leq n$, where the RF kernel is defined in Equation (2.1).
2. Center the variables at the query point: $\tilde{\mathbf{X}}_i = \mathbf{X}_i - \mathbf{x}_0$ for $1 \leq i \leq n$.
3. Calculate the weighted sample mean $\boldsymbol{\mu} = \sum_{i=1}^n w_i \tilde{\mathbf{X}}_i$.
4. Output the eigenvector of $\sum_{i=1}^n w_i (\tilde{\mathbf{X}}_i - \boldsymbol{\mu})(\tilde{\mathbf{X}}_i - \boldsymbol{\mu})^T$ with the smallest eigenvalue.

Algorithm 2.2: Local principal direction.

In step 4 of Algorithm 2.2, we estimate the LPD at $\mathbf{x}_0$ by solving a non-sparse eigenvalue problem; however, it is often practically important to seek a k -sparse approximation to the LPD at $\mathbf{x}_0$, i.e., $\|\text{LPD}(\mathbf{x}_0)\|_0 = k$. The truncated power method (Yuan & Zhang, 2013) can efficiently solve this k -sparse eigenvalue problem. We will not consider k -sparse LPDs in this chapter, but the `drforest` Python package includes an efficient implementation.

2.4 Theoretical Results

Here, we develop results that give theoretical insights into how a binary splitting mechanism affects SDR estimation in a dimension reduction tree. The main results show that if the conditions sufficient to estimate SIR/SAVE hold in a node and we make a binary split of the form $\mathbb{1}_{\{\beta^T \mathbf{x} \leq c\}}$ where β lies in a local central subspace, then these conditions do not need to hold in the child nodes. We provide the proofs in Section 2.8.4.

First, we introduce some notation and assumptions. Let $A \subseteq \mathbb{R}^p$ denote the subset of the domain occupied by an internal node A in a DRT. We denote the local central subspace of data contained in A by $S_{Y|\mathbf{X}, \mathbf{X} \in A}$. Also, let $B_A \in \mathbb{R}^{p \times d}$ denote a matrix such that $\text{span}(B_A) = S_{Y|\mathbf{X}, \mathbf{X} \in A}$, i.e., $Y \perp \mathbf{X} \mid B_A^T \mathbf{X}$ for data in A . Known results from the SDR literature demonstrate that the following two assumptions are sufficient for SIR (SAVE) to recover a subspace of $S_{Y|\mathbf{X}, \mathbf{X} \in A}$ when applied to data in an internal node A .

Assumption A2.1 (Local Linearity Condition). $\mathbb{E}[\mathbf{X} \mid B_A^T \mathbf{X}, \mathbf{X} \in A]$ is a linear function of $B_A^T \mathbf{X}$.

Assumption A2.2 (Local Constant Variance Condition). $\text{Var}(\mathbf{X} \mid B_A^T \mathbf{X}, \mathbf{X} \in A)$ is a nonrandom matrix.

When $A = \mathbb{R}^p$, Assumption A2.1 and Assumption A2.2 reduce to the usual linearity and constant variance conditions commonly made in the SDR literature. However, the DRT algorithm also requires the recovery of directions in $S_{Y|\mathbf{X}, \mathbf{X} \in A}$ within A 's child nodes where the local linearity and constant variance conditions are unlikely to remain valid. Remarkably, the following results address this requirement by showing that as long as Assumptions A2.1 and Assumption A2.2 hold in the parent node A , population-level SIR (SAVE) applied to data in A 's children still recovers directions within the local central subspace, $S_{Y|\mathbf{X}, \mathbf{X} \in A}$.

Theorem 2.1. Assume that model (2.4) is satisfied in the parent node A , i.e., $Y \perp \mathbf{X} \mid B_A^T \mathbf{X}$ for some B_A if $\mathbf{X} \in A$. Further assume Assumption A2.1 holds within the parent node. If we implement a splitting rule of the form $\mathbb{1}_{\{\beta^T \mathbf{x} \leq c\}}$, where $\beta \in \text{span}(B_A)$, then within each child node $A_L = A \cap \{\beta^T \mathbf{x} \leq c\}$ and $A_R = A \setminus A_L$ we have that

$$\Sigma_A^{-1} \mathbb{E}[\mathbf{X} \mid Y, \mathbf{X} \in A_L] \in \text{span}(B_A) \quad \text{and} \quad \Sigma_A^{-1} \mathbb{E}[\mathbf{X} \mid Y, \mathbf{X} \in A_R] \in \text{span}(B_A),$$

e.g., the population-level SIR estimator remains valid in the child nodes.

Theorem 2.2. Assume that model (2.4) is satisfied in the parent node A , i.e., $Y \perp \mathbf{X} \mid B_A^T \mathbf{X}$ for some B_A if $\mathbf{X} \in A$. Further assume Assumption A2.1 and Assumption A2.2 hold within the parent node. If we implement a splitting rule of the form $\mathbb{1}_{\{\beta^T \mathbf{x} \leq c\}}$, where $\beta \in \text{span}(B_A)$, then within each child node $A_L = A \cap \{\beta^T \mathbf{x} \leq c\}$ and $A_R = A \setminus A_L$ we have that

$$\Sigma_A - \text{Var}(\mathbf{X} \mid Y, \mathbf{X} \in A_L) \in \Sigma_A \text{span}(B_A) \quad \text{and} \quad \Sigma_A - \text{Var}(\mathbf{X} \mid Y, \mathbf{X} \in A_R) \in \Sigma_A \text{span}(B_A),$$

e.g., the population-level SAVE estimator remains valid in the child nodes.

These theorems combined with known results from the dimension reduction literature, Corollary 3.1 and Proposition 5.1 in B. Li (2018), imply that population-level SIR and SAVE applied in the two child nodes recover directions within $S_{Y|\mathbf{X}, \mathbf{X} \in A}$ even when the child nodes violate the linearity and constant variance conditions.

Theorem 2.1 and Theorem 2.2 still hold when A_L and A_R are replaced by any ancestral node $A_C \subseteq A$ constructed by repeatedly applying splitting rules of the form $\mathbb{1}_{\{\beta^T \mathbf{x} \leq c\}}$, where each $\beta \in \text{span}(B_A)$. The consequence of this observation is important: When the linearity and constant variance conditions hold for the covariates' joint distribution at the tree's root node, where $A = \mathbb{R}^p$, then we preserve SDR estimation further down the tree as long as the splitting directions remain within $S_{Y|\mathbf{X}}$. Corollary 2.1 summarizes this result.

Corollary 2.1. Assume that $Y \perp \mathbf{X} \mid B^T \mathbf{X}$ and Assumption A2.1 and Assumption A2.2 hold in the root node, where $A = \mathbb{R}^p$, then

$$\Sigma^{-1} \mathbb{E}[\mathbf{X} \mid Y, \mathbf{X} \in A_C] \in \text{span}(B) \quad \text{and} \quad \Sigma - \text{Var}(\mathbf{X} \mid Y, \mathbf{X} \in A_C) \in \Sigma \text{span}(B),$$

where $A_C \subset \mathbb{R}^p$ is any child node constructed by repeatedly applying splitting rules of the form $\mathbb{1}_{\{\beta^T \mathbf{x} \leq c\}}$ or $\mathbb{1}_{\{\beta^T \mathbf{x} > c\}}$ for $\beta \in \text{span}(B)$.

This corollary suggests that a DRT's non-root nodes can violate the linearity and constant variance conditions as long as the splitting directions are asymptotically within $\text{span}(B)$. Although we do not establish the splitting directions' consistency, this result suggests why DRT's appear robust to violations of the linearity and constant variance conditions.

2.5 Simulation Studies

We provide several simulation studies to evaluate dimension reduction forest's predictive performance and analyze our method's accuracy in generating local principal directions.

2.5.1 Predictive Performance

We compared DRFs with two major competitors: traditional random forests as implemented in the `scikit-learn` Python package and Nadaraya-Watson kernel estimators (NW Kernel). We also included two variations on the traditional random forest (SIR + RF and SAVE + RF) and the Nadaraya-Watson kernel estimator (SIR + NW Kernel and SAVE + NW Kernel). As their names suggest, these two variants first extract global SDR predictors using SIR or SAVE, and then train a random forest or an Nadaraya-Watson kernel estimator using only these features. This approach of using global SDR predictors within a kernel estimator is common practice, see Adraghi and Cook (2009). We included these methods to demonstrate the benefit of the recursive dimension reduction performed by DRFs.

We evaluated predictive performance using 50 repeated train-test splits on seven known regression functions. Each split contained $n_{train} = 2000$ training samples and $n_{test} = 1000$ testing samples. We recorded each method’s mean squared error (MSE) on the test set. The simulations were repeated 50 times with different random seeds. Since machine learning methods are sensitive to hyperparameter settings, it is crucial to explore a large space of values. Table 2.3 in Section 2.8.5 lists all hyperparameter settings used in our experiments. All tree ensembles used $M = 500$ trees. Section 2.8.6 confirmed that each ensemble’s test set performance converged. Also, Section 2.8.7 found that the DRF’s performance was relatively robust to the hyperparameters of SIR (SAVE). To compare each method fairly, we reported the lowest test error achieved by each method over all parameter settings. This procedure differs from the standard practice of using cross-validation to choose the hyperparameters before evaluating on an independent test set. Our approach mitigates the impact of parameter tuning by reporting each method’s best performance over all parameter settings.

For the predictive task, we considered the following seven regression functions:

- *Simulation 1:* $Y = 20 \max \left\{ e^{-2(X_1 - X_2)^2}, 2e^{-0.5(X_1^2 + X_2^2)}, e^{-(X_1 + X_2)^2} \right\} + \varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} U[-3, 3]^5$.
- *Simulation 2:* $Y = 20 \max \left\{ e^{-18X_1^2}, e^{-18X_2^2}, 1.75e^{-20(X_1 + X_2)^2}, 1.75e^{-20(X_1 - X_2)^2} \right\} + \varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} U[-1, 1]^4$.
- *Simulation 3:* $Y = (\mathbf{X}^T \beta_1)^2 + (\mathbf{X}^T \beta_2)^2 + 0.5\varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} N(\mathbf{0}_{10}, \Sigma)$.
- *Simulation 4:* $Y = \mathbf{X}^T \beta_3 (\mathbf{X}^T \beta_4)^2 + (\mathbf{X}^T \beta_5) (\mathbf{X}^T \beta_6) + 0.5\varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} N(\mathbf{0}_{10}, \Sigma)$.
- *Simulation 5:* $Y = \mathbf{X}^T \beta_7 (\mathbf{X}^T \beta_8)^2 + (\mathbf{X}^T \beta_9) (\mathbf{X}^T \beta_{10}) + 0.5\varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} N(\mathbf{0}_2, \Sigma)$.
- *Simulation 6:* $Y = \sum_{c=0}^1 [\gamma_c (\mathbf{X}^T \beta)^2 + \alpha_c] \mathbb{1}_{\{W=c\}} + 0.5\varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} N(\mathbf{0}_3, \Sigma)$,
- *Simulation 7:* $Y = \sum_{c=0}^1 [\gamma_c (\mathbf{X}^T \beta_c)^2 + \alpha_c] \mathbb{1}_{\{W=c\}} + 0.5\varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} N(\mathbf{0}_3, \Sigma)$.

Note that $\mathbf{0}_n$ denotes an n -dimensional vector of zeros. We sampled $\varepsilon_i \stackrel{\text{iid}}{\sim} N(0, 1)$, $W_i \stackrel{\text{iid}}{\sim} \text{Bernoulli}(0.5)$, and set $\Sigma_{ij} = 0.5^{|i-j|}$ for all simulations. We also include the three Friedman functions first introduced in Friedman (1991). Section 2.8.5 provides the Friedman functions’ definitions and the values of the remaining parameters.

We designed the first two simulations to test the DRF’s ability to adapt to local dimension reduction structure. Simulation 1 is the motivating example in Section 2.2. Simulation 2 contains four different local one-dimensional subspaces; two are axis-aligned, and two lie in a rotated coordinate system. The next two simulations are commonly used to test SDR methods, and Simulation 5 contains a similar function but with no dimension reduction ($d = p$). We included these functions to compare the DRF with a standard RF trained on SIR and SAVE input features. Simulations 6 and 7 assess the DRF’s performance in the presence of a single binary-valued categorical covariate. Finally, Friedman 1–3 are classic regression problems used to test the performance of random forest methods. Here, our goal is to compare DRFs with RFs on problems

	DRF	SIR + RF	SAVE + RF	NW Kernel	SIR + NW Kernel	SAVE + NW Kernel
Simulation #1	23.93 (6.08)	-1180.25 (221.77)	-62.88 (43.37)	-1116.11 (107.45)	-1120.66 (108.47)	-1119.08 (107.29)
Simulation #2	65.87 (4.36)	-500.71 (103.25)	-188.85 (105.25)	-610.03 (81.84)	-610.75 (81.71)	-610.87 (82.28)
Simulation #3	69.80 (5.26)	-157.48 (64.65)	71.29 (5.02)	-29.21 (15.02)	-194.10 (42.12)	-29.21 (15.02)
Simulation #4	28.08 (7.21)	-41.90 (19.09)	6.82 (14.16)	-19.61 (10.88)	-31.83 (12.62)	-21.94 (12.85)
Simulation #5	9.44 (6.37)	1.55 (11.13)	-1.05 (6.61)	-22.14 (11.38)	-22.27 (11.36)	-39.05 (15.03)
Simulation #6	37.14 (8.43)	25.68 (10.43)	31.70 (10.20)	-8.38 (12.28)	-8.48 (12.41)	-29.07 (7.60)
Simulation #7	50.35 (5.64)	25.60 (9.59)	31.63 (7.64)	-15.09 (12.11)	-15.79 (12.47)	-35.60 (11.90)
Friedman #1	25.86 (3.32)	-66.27 (12.08)	1.40 (10.74)	-123.98 (13.99)	-124.42 (14.01)	-124.39 (14.01)
Friedman #2	36.53 (10.10)	-613.50 (178.54)	-43.86 (34.44)	-2158.59 (264.81)	-2162.67 (265.14)	-2160.85 (264.83)
Friedman #3	0.22 (1.07)	-1.33 (1.51)	-1.29 (1.53)	-0.83 (1.27)	-0.84 (1.27)	-0.85 (1.27)

Table 2.1: Percentage (100%) improvement over the RF baseline: $1 - (\text{MSE of method}) / (\text{MSE of random forest})$. The standard deviation over the 50 runs is displayed in parentheses. Methods with the best improvement (if any) over the RF are shaded in gray.

where DRFs have no clear advantage.

Table 2.1 displays each method’s mean percentage (100%) improvement over the RF baseline and the corresponding standard deviations for the seven simulation scenarios. The best performing method for each scenario is shaded in gray. A negative value indicates degraded performance, a value of zero indicates no improvement, and a positive value indicates an improvement. Overall, the proposed DRF model performs well in all simulations.

DRFs see some of their largest gains over the competitors in Simulation 1 and Simulation 2. We expected this behavior because these regression surfaces exhibit heterogeneity in the dependent local subspace. Our intuition is that the DRF’s recursive estimation of the local subspace allows it to modify its splitting direction based on the prediction point. The fact that the SIR + RF and SAVE + RF baselines, which can only split along the global DR subspace, have a much higher MSE lends credibility to this assertion.

Simulation 3, 4 and 5 demonstrate that the DRF’s performance does not degrade when a global DR subspace adequately describes the regression surface. In Simulation 3, the DRF performs similarly to the SAVE + RF method due to the presence of a global quadratic response surface. However, the DRF outperforms all methods on Simulation 4, which contains a regression surface that is troublesome for SIR and on which SAVE sees only a modest improvement. Lastly, Simulation 5 demonstrates that the DRF still performs well when there is no dimension reduction. We conclude that the DRF is a flexible alternative to trying different SDR methods as inputs to nonparametric regressors.

Simulation 6 and Simulation 7 show that DRFs can handle a single binary-valued covariate. As in Simulation 5, the DRF performs similarly to the SAVE + RF due the presence of a global quadratic response surface. However, the DRF outperforms all methods on Simulation 7 where the subspace varies by the value of the categorical covariate.

Lastly, DRFs perform the best or equivalent to the RF baseline on Friedman 1–3. DRF’s improvement on Friedman 1 and 2 may be due to better capturing linear and quadratic terms. SAVE + RF’s relative performance on these functions supports this claim. These results demonstrate that DRFs can be a competitive alternative to traditional RFs.

2.5.2 Local Principal Directions

Next, we evaluated how the LPD estimates’ (Algorithm 2.2) accuracy scaled with (1) a decreasing signal-to-noise ratio and (2) an increase in the number of uninformative covariates. We considered the following four regression functions. *Simulation 1*: $Y = |X_1| + |X_2| + \varepsilon$, *Simulation 2*: $Y = X_1 + X_2^2 + \varepsilon$, *Simulation 3*: $Y =$

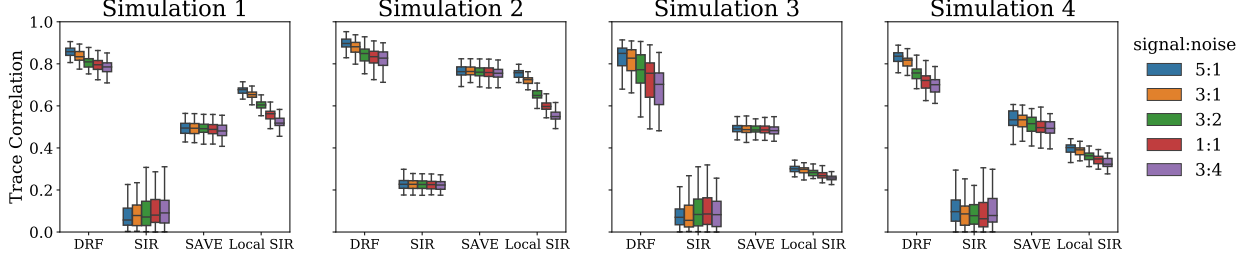


Figure 2.3: Boxplots of the trace correlations between the local gradients and the LPDs estimated by the four methods outlined in Section 2.5.2 for different signal-to-noise ratios. For each method, the boxplots are sorted in descending order by signal-to-noise ratio, which matches the order in the legend. Higher values of the trace correlation are better.

$5 \max \left\{ e^{-0.25X_1^2}, e^{-0.25X_2^2} \right\} + \varepsilon$, and *Simulation 4*: $Y = 20 \max \left\{ e^{-2(X_1-X_2)^2}, 2e^{-0.5(X_1^2+X_2^2)}, e^{-(X_1+X_2)^2} \right\} + \varepsilon$. For each simulation, we drew $\mathbf{X}_i \stackrel{\text{iid}}{\sim} U[-3, 3]^p$ and $\varepsilon_i \stackrel{\text{iid}}{\sim} N(0, \sigma^2)$.

We chose these functions because they exhibit a variety of local dimension reduction structures. Simulation 1 contains the simplest structure with only four local directions that vary by quadrant in the X_1 - X_2 plane. Simulation 2 contains a simple analytic gradient, $(1, X_2)^T$, that varies along a single direction. This simulation tested whether the LPD estimates capture smoothly varying local structures. Simulation 3 is composed of a cross of simple one-dimensional axis-aligned subspaces. Simulation 4 is the function in the motivating example, which includes a highly variable local structure.

To evaluate the LPD estimates' accuracy, we generated $n = 2000$ samples from each regression function and selected $n_{\text{test}} = 100$ points uniformly at random to test the LPDs. We recorded the trace correlation between the estimated LPDs and the normalized gradients at the randomly chosen test points. We repeated each simulation 50 times with different random seeds. To evaluate the effect of a degrading signal-to-noise ratio, we performed the experiments with $p = 10$ covariates and set σ^2 so that the signal-to-noise ratio was 5:1, 3:1, 3:2, 1:1, and 3:4. We define the signal-to-noise ratio as $\text{Var}(\mathbb{E}[Y | \mathbf{X}]) / \sigma^2$.

We compared the LPDs generated by the following four methods: DRF, SAVE, SIR, and Local SIR. LPD estimation is sensitive to the DRF's minimum leaf node size. We searched over $n_{\text{min}} \in \{3, 10, 25, 50, 100\}$ at each prediction point and reported the best direction in terms of trace correlation to mitigate this effect. Furthermore, we set $m_{\text{try}} = 5$ for all simulations. The SIR (SAVE) procedure always selects the first principal SDR direction estimated with the standard SIR (SAVE) algorithm with 10 slices. We included these benchmarks to ensure that local structure informs the LPD algorithm. Local SIR is related to the KNN-SIR method proposed in Q. Wang (2009). We estimated the leading SIR direction restricted to each prediction point's $k \in \{\max(10, p), 25, 50, 100\}$ nearest neighbors. We recorded the best trace correlation obtained by these four directions.

Figure 2.3 displays boxplots of the four method's trace correlations at $p = 10$ while the signal-to-noise ratio varied. For most scenarios, the DRF procedure performs best with slowly degrading performance as the signal-to-noise ratio decreases. In particular, as the signal-to-noise ratio decrease, the DRF estimates degenerate to the performance of SAVE. This behavior is especially evident in Scenario 2, where the principal SAVE direction is highly informative. In conclusion, LPDs produced by DRFs adequately describe the local dimension reduction structure, especially when the signal-to-noise ratio is greater than one.

We include the analysis of how LPD estimation performed as the number of uninformative covariates increased in Section 2.8.8. Our analysis indicated that LPDs are relatively robust to the number of uninfor-

mative covariates, unlike Local SIR, which performed poorly due to the curse of dimensionality.

2.6 Real Data Applications

In this section, we focus on the empirical performance of dimension reduction forests on real data sets. To illustrate our algorithm’s predictive value, we applied DRFs to a variety of real-world regression tasks. Furthermore, we used the LPDs produced by a DRF to understand the monthly variation of PM2.5 concentration in Beijing, China.

2.6.1 Predictive Performance on Real Data Sets

We applied the dimension reduction forest algorithm to 14 regression data sets taken from the UCI Machine Learning Repository (Dua & Graff, 2017) and OpenML (Vanschoren et al., 2013). The 14 data sets represent a variety of regression tasks on primarily real-valued covariates. The Athletes and Diabetes data sets contain continuous covariates and a single binary-valued categorical covariate, while the other data sets contain only continuous covariates. Table 2.4 in Section 2.8.5 displays each data set’s sample size and the number of features. We compared the DRF’s predictive performance to the same six methods included in Section 2.5.1. For each data set, we performed 15 different rounds of 10-fold cross-validation. We calculated the average out-of-sample MSE over the 10-folds and recorded the percentage (100%) improvement over the random forest baseline. As before, we recorded the best value over the range of hyperparameter settings displayed in Table 2.3 of Section 2.8.5. Table 2.2 contains the results for the 14 data sets.

	DRF	SIR + RF	SAVE + RF	NW Kernel	SIR + NW Kernel	SAVE + NW Kernel
Abalone	5.91 (0.41)	4.66 (0.51)	3.53 (0.84)	−18.79 (0.49)	−27.64 (0.52)	−66.74 (1.95)
Athletes	38.32 (5.89)	25.97 (8.68)	−89.02 (60.17)	−143.83 (23.26)	−143.83 (23.26)	−524.88 (92.77)
Auto Price	−7.07 (3.79)	−58.32 (18.19)	−189.10 (32.07)	−58.89 (10.41)	−62.45 (13.99)	−210.93 (28.86)
Bank8FM	13.22 (0.30)	6.01 (0.74)	6.76 (0.27)	−135.76 (1.05)	−135.95 (1.06)	−135.92 (1.04)
Body Fat	13.44 (3.20)	−20.83 (8.00)	−227.34 (47.44)	−294.76 (19.67)	−294.76 (19.67)	−912.44 (122.01)
CPU Small	−5.70 (0.43)	−48.51 (3.60)	−36.91 (3.18)	−57.84 (3.71)	−65.71 (7.01)	−148.87 (7.47)
Diabetes	6.70 (0.78)	3.36 (1.67)	−0.82 (2.78)	2.35 (1.04)	2.35 (1.04)	−45.61 (4.09)
Fish Catch	37.59 (5.68)	9.37 (12.89)	−110.90 (40.24)	−46.95 (15.92)	−745.03 (230.87)	−2247.21 (355.27)
Kin8nm	50.57 (0.29)	23.97 (1.13)	50.89 (0.35)	32.98 (0.39)	32.75 (0.39)	32.90 (0.40)
Liver	0.31 (0.90)	−4.10 (1.76)	−5.16 (1.44)	−13.54 (2.28)	−13.67 (2.40)	−17.16 (2.12)
Mu284	22.15 (2.18)	−8.58 (9.69)	−2.97 (10.34)	−13.78 (3.28)	−69.76 (8.78)	−165.74 (19.20)
Puma32H	6.09 (0.31)	−997.69 (19.96)	−175.46 (3.65)	−1757.70 (12.61)	−1761.38 (12.70)	−1757.70 (12.61)
Puma8NH	2.10 (0.25)	0.79 (0.33)	0.57 (0.44)	−33.84 (0.69)	−33.88 (0.69)	−33.87 (0.69)
Wisconsin	3.63 (0.89)	−1.21 (1.97)	−3.05 (2.42)	−8.36 (1.23)	−8.36 (1.23)	−9.36 (1.13)

Table 2.2: Percentage (100%) improvement over the RF baseline: $1 - (\text{MSE of method}) / (\text{MSE of random forest})$. The standard deviation over the 15 runs is displayed in parentheses. Methods with the best improvement (if any) over the RF are shaded in gray.

Overall, the DRF performs better or equivalent to the RF baseline on most data sets. Specifically, the DRF outperforms the RF on 11 out of 14 data sets. The largest improvements occur either at small sample sizes or where SIR (SAVE) features already result in an improvement. Furthermore, the DRF saw an improvement where SIR (SAVE) features hindered the RF’s performance, e.g., on Puma32H and Wisconsin. The DRF method performed worse than the RF only when the SDR features performed poorly. In conclusion, DRFs are a simple alternative to complicated SDR feature engineering with the added benefit that they can improve performance due to their ability to adapt to local structures.

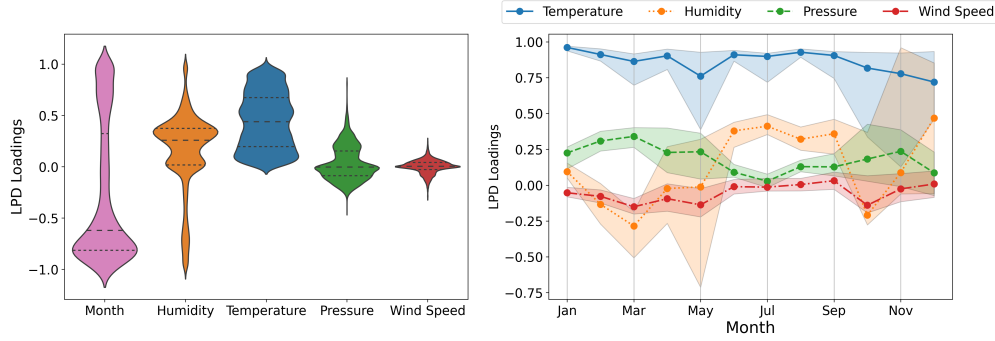


Figure 2.4: Violin plot of each LPDs’ marginal distribution with the temperature loading’s sign forced positive (left). Dashed lines indicate the median and dotted lines are the interquartile range. The distribution of the LPDs loadings as a function of month (right). Lines indicate the median and shaded regions denote the interquartile range.

2.6.2 PM2.5 Concentration in Beijing, China

Finally, we demonstrate how to use the LPDs generated by the DRF algorithm to understand the seasonal variation of PM2.5 (particulate matter) concentration ($\mu\text{g}/\text{m}^3$) in the Chaoyang district of Beijing, China. The raw data was collected by the Urban Air project (Urban Computing Team, Microsoft Research) and consists of meteorological and air quality data collected hourly by 437 air quality stations located in 43 Chinese cities. The data was collected over one year, from May 1st, 2014 to April 30th, 2015. We limited the analysis to observations recorded in Beijing’s Chaoyang district because it is the largest and most populous district. For more details on this data set, see Zheng et al. (2015).

The analysis’s goal is to quantify how different meteorological covariates influence the predictions of PM2.5 concentration throughout the year. We included five covariates in the analysis: **month**, **temperature** ($^{\circ}\text{C}$), (atmospheric) **pressure** (hPa), **humidity** (%), and **wind speed** (m/s). We coded the month as an integer from 1 to 12, starting with January. Furthermore, we standardized all covariates so that they have mean zero and unit variance. The final processed data set contained $n = 25,794$ observations with $p = 5$ covariates. For reference, the pairwise scatter plots and marginal histograms of the covariates used in the analysis are displayed in Figure 2.9 of Section 2.8.8.

We fit a dimension reduction forest to this data set with $M = 500$ trees, $n_{\text{slices}} = \sqrt{n}$, and no feature screening. We selected 3 as the minimum leaf node size using an 80%–20% train-test split to search over $n_{\text{min}} \in \{3, 5, 10\}$. Then, we re-fit the DRF to the full data set with $n_{\text{min}} = 3$. The model’s in-sample R^2 is 0.94, indicating a good fit to the data.

To discern how the covariates affected the prediction of PM2.5 concentration, we plotted the LPD loadings’ marginal distributions in Figure 2.4. Since the LPD estimates are invariant to sign changes, we forced the temperature loadings’ sign positive. The largest loadings are on month, humidity, and temperature. Air pressure and wind speed play a less significant role. For comparison, we included the global permutation-based importance extracted from a traditional random forest and a DRF in Figure 2.10 of Section 2.8.8. These results corroborate wind speed’s lack of global importance.

To assess how these effects varied by month, we plotted the loadings’ distributions as a function of month in Figure 2.4. During the summer months (Jun, Jul, Aug, and Sep), the regression function primarily varies along the subspace spanned by two meteorological variables: **temperature + humidity**. In contrast, the regression function varies along the subspace spanned by **temperature + pressure - humidity** from

February to April. These findings indicate that PM2.5 concentration’s cause varies by season (Z. Chen et al., 2020).

2.7 Discussion

This chapter proposed dimension reduction forests, a new nonparametric estimator that can adapt to local dimension reduction structure and measure local variable importance through a novel application of sufficient dimension reduction. We presented theory on the sufficient conditions for using SDR within the forest’s splitting rule. A simulation study demonstrated that dimension reduction forests can outperform traditional random forests in certain predictive tasks, such as when the regression function contains a local dimension reduction structure. Also, we demonstrated the accuracy of our proposed local principal direction procedure. Finally, we established our model’s effectiveness on 14 real-world regression tasks and used it to study seasonal variation in PM2.5 concentration in Beijing.

The use of sufficient dimension reduction within random forests opens up an exciting line of future work on designing more efficient adaptive kernels. For example, although SIR and SAVE are robust to violations of the linearity and constant variance conditions, these assumptions can be removed entirely using computationally expensive semiparametric (Y. Ma & Zhu, 2012) or forward regression (Xia et al., 2002) estimators. Furthermore, high-dimensional problems pose a challenge for our method, which we accounted for through an optional variable screening step. Another approach is to utilize sparse SDR estimators (Lin et al., 2018, 2019). However, the computational complexity and sample size requirements of these two approaches are prohibitive for most practical problems. Modifications to improve the scalability of the aforementioned estimators within a DRF are an area of research interest. Further research directions include establishing the asymptotic properties of DRFs and their associated local principal direction measure. Regardless, dimension reduction forests remain a flexible and interpretable nonparametric estimator.

2.8 Supplementary Material

Python-package for DRFs: Python-package `drforest` containing all data sets and scripts used to run the analyses in this chapter. This package is available at <https://github.com/joshloyal/drforest>.

2.8.1 Overview of Inverse Regression Methods

In this section, we detail the SIR and SAVE algorithms used to estimate the central subspace in a dimension reduction tree. Both methods fall under the framework of inverse regression and can be formulated as solving the following generalized eigenvalue problem:

$$\Lambda B = \lambda \Sigma B,$$

where Σ is the covariance matrix of $\mathbf{X}$, the columns of B span at least part of the central subspace, and Λ is a method specific matrix that is a function of the moments of $\mathbf{X} \mid Y$. Since these estimators reverse the usual dependence of Y on $\mathbf{X}$, they are known as inverse regression methods in the literature. SIR uses first order moment information with

$$\Lambda_{\text{SIR}} = \text{Cov}(\mathbb{E}[\mathbf{X} \mid Y]). \quad (2.5)$$

It is easy to see that SIR fails to recover B when $\mathbb{E}[\mathbf{X} \mid Y] = 0$, e.g., when $\mathbb{E}[\mathbf{X} \mid Y]$ is symmetric about $\mathbb{E}[\mathbf{X}]$. A solution proposed in SAVE is to utilize second moment information. In particular, SAVE uses the matrix

$$\Lambda_{\text{SAVE}} = \mathbb{E}[(I_p - \text{Var}(\mathbf{X} \mid Y))^2], \quad (2.6)$$

where I_p is the p -dimensional identity matrix. Algorithm 2.3 outlines the SIR algorithm, and Algorithm 2.4 details the SAVE algorithm. Note that the linearity condition is sufficient for SIR to recover a direction in $S_{Y|\mathbf{X}}$, while SAVE requires both the linearity condition and the constant variance condition.

Algorithm 2.3 Sliced Inverse Regression

```

1: procedure SOLVESIR(data set  $\mathcal{D}_n = \{\mathbf{X}_i, Y_i\}_{i=1}^n$ )
2:    $U, \hat{\mu} \leftarrow \text{CENTER}(X)$ 
3:    $Q, R \leftarrow \text{QRDECOMPOSITION}(U)$ 
4:    $Z \leftarrow \sqrt{n} Q$  ▷ Whiten the data.
5:   for slice  $S_k$  in GETSLICES( $Z, \mathbf{n\_slices}$ ) do
6:      $\hat{\mu}_{S_k} \leftarrow \frac{1}{|S_k|} \sum_{i=1}^n \mathbf{Z}_i \mathbb{1}_{\{\mathbf{Z}_i \in S_k\}}$ 
7:      $\hat{\Lambda}_{\text{SIR}} \leftarrow \sum_{k=1}^{\mathbf{n\_slices}} \frac{|S_k|}{n} (\hat{\mu}_{S_k} - \hat{\mu})(\hat{\mu}_{S_k} - \hat{\mu})^T$  ▷ See Equation (2.5).
8:     Solve  $\hat{\Lambda}_{\text{SIR}} \hat{\Gamma} = \hat{\Lambda} \hat{\Gamma}$ 
9:      $\hat{B} \leftarrow (\sqrt{n} R)^{-1} \hat{\Gamma}$  ▷ Solved via back-substitution.
10:  output Estimated directions  $\hat{B}$  and eigenvalues  $\hat{\lambda}$ .
```

CENTER calculates the empirical feature-wise means $\hat{\mu}$ and centers the data: $\mathbf{U}_i = \mathbf{X}_i - \hat{\mu}$. GETSLICES splits the rows of Z into $\mathbf{n_slices}$ contiguous segments with roughly equal sample size. Note that Z is assumed to be presorted in terms of the target Y .

Algorithm 2.4 Sliced Average Variance Estimation

```

1: procedure SOLVESAVE(data set  $\mathcal{D}_n = \{\mathbf{X}_i, Y_i\}_{i=1}^n$ )
2:    $U, \hat{\mu} \leftarrow \text{CENTER}(X)$ 
3:    $Q, R \leftarrow \text{QRDECOMPOSITION}(U)$ 
4:    $Z \leftarrow \sqrt{n} Q$  ▷ Whiten the data.
5:    $\hat{\Lambda}_{\text{SAVE}} \leftarrow 0_{p \times p}$ 
6:   for slice  $S_k$  in GETSLICES( $Z, \mathbf{n\_slices}$ ) do
7:      $\tilde{Z}_{S_k} \leftarrow \text{CENTERSLICE}(Z, S_k)$ 
8:      $\hat{\Lambda}_{\text{SAVE}} \leftarrow \hat{\Lambda}_{\text{SAVE}} + \frac{|S_k|}{n} (I_{|S_k|} - \tilde{Z}_{S_k}^T \tilde{Z}_{S_k})^2$  ▷ See Equation (2.6).
9:   Solve  $\hat{\Lambda}_{\text{SAVE}} \hat{\Gamma} = \hat{\Lambda} \hat{\Gamma}$ 
10:   $\hat{B} \leftarrow (\sqrt{n} R)^{-1} \hat{\Gamma}$  ▷ Solved via back-substitution.
11:  output Estimated directions  $\hat{B}$  and eigenvalues  $\hat{\lambda}$ .
```

CENTER calculates the empirical feature-wise means $\hat{\mu}$ and centers the data: $\mathbf{U}_i = \mathbf{X}_i - \hat{\mu}$. CENTERSLICE is the same as CENTER but restricted to data in a given slice S_k . It returns the centered data for slice S_k . GETSLICES splits the rows of Z into $\mathbf{n_slices}$ contiguous segments with roughly equal sample size. Note that Z is assumed to be presorted in terms of the target Y .

2.8.2 Computational Complexity Analysis

An advantage of random forests over other machine learning methods is their relatively fast training speed, so transferring that speed to dimension reduction forests is paramount. In Section 2.8.2, we analyze the theoretical computational complexity of DRFs. Section 2.8.2 quantifies the empirical run times of dimension reduction trees on simulated data.

Theoretical Computational Complexity

We demonstrate that DRF's theoretical computational complexity is comparable to a standard random forest when $p < n$. We only consider DRFs built without the optional screening step in step (a) of Algorithm 2.1; however, we allow the random selection of $m_{try} \leq p$ variables at each node. The following theorem quantifies the computational complexity of building a single dimension reduction tree.

Theorem 2.3. *Let $C(s)$ denote the computational complexity of forming a single node on a data set with sample size s . Assume that the trees are grown such that all possible sample sizes assigned to a child node are equally probable, i.e., uniformly distributed on $\{1, \dots, n_A\}$ where n_A is the number of samples in node A . Also, assume m_{try} variables are used to estimate each splitting direction. If there exist constants $c_1, c_2 > 0$ such that $c_1(sm_{try}^2 + s \log(s)) \leq C(s) \leq c_2(sm_{try}^2 + s \log(s))$ (for all $s \geq 1$), then the computational complexity for building a dimension reduction tree on a data set with sample size n is $\mathcal{O}(m_{try}^2 n \log(n) + m_{try} n \log^2(n))$.*

Proof. The proof is a modification of the proof of Theorem 5.6 in Louppe (2014). Let $T(n)$ denote the computational complexity for building a decision tree from a data set $\mathcal{D}_n$ with n samples. For simplicity of the proof, we assume that the decision trees can be fully grown such that each leaf node contains a single sample. In this framework, the decision tree building process is characterized by the following recurrence relation:

$$\begin{cases} T(1) = c_1, \\ T(n) = C(n) + T(n_L) + T(n_R), \end{cases}$$

where c_1 is the time complexity to make a leaf node, and n_L and n_R are the number of samples in the left and right child node. The average time complexity assumes n_L and n_R are uniformly distributed on $\{1, \dots, n-1\}$. Thus the recurrence relation in the average case is

$$\begin{cases} T(1) = c_1, \\ T(n) = C(n) + \frac{1}{n-1} \sum_{i=1}^{n-1} (T(i) + T(n-i)). \end{cases}$$

The proof proceeds in standard fashion. Let $C(n) = c_3(nm_{try}^2 + n \log(n))$. By symmetry and multiplying by $(n-1)$ the recurrence becomes

$$(n-1)T(n) = (n-1)c_3(nm_{try}^2 + n \log(n)) + 2 \sum_{i=1}^{n-1} T(i). \quad (2.7)$$

In addition, for $n \geq 3$, substituting n with $n-1$ yields

$$(n-2)T(n-1) = (n-2)c_3((n-1)m_{try}^2 + (n-1) \log(n-1)) + 2 \sum_{i=1}^{n-2} T(i). \quad (2.8)$$

Define $S(n) = \frac{T(n)}{n}$. Now subtracting Equation (2.8) from Equation (2.7) and dividing by $n(n-1)$ we have

$$\begin{aligned} S(n) &= S(n-1) + 2c_3 \frac{m_{try}^2}{n} + c_3 \frac{2}{n} \log(n-1) + c_3 \log\left(\frac{n}{n-1}\right), \\ &= c'_1 + c_3 \sum_{i=2}^n \frac{2}{i} (m_{try}^2 + \log(i-1)) + \log\left(\frac{i}{i-1}\right), \end{aligned}$$

$$\begin{aligned}
&= c'_1 + c_3 \log(n) + c_3 \sum_{i=1}^n \frac{2}{i} (m_{try}^2 + \log(i-1)), \\
&\leq c'_1 + c_3 \log(n) + 2c_3 (m_{try}^2 + \log(n)) \sum_{i=2}^n \frac{1}{i}, \\
&= c'_1 + c_3 \log(n) + 2c_3 (m_{try}^2 + \log(n)) (H_n - 1), \\
&= \mathcal{O}(H_n m_{try}^2 + H_n \log(n)),
\end{aligned}$$

where $c'_1 = c_1/n$ and H_n is the n th harmonic number. To finish the bound we note that $H_n \sim \log(n)$. As a result we see that the average case computational complexity of building this decision tree is

$$T(n) = \mathcal{O}(m_{try}^2 n \log(n) + m_{try} n \log^2(n)),$$

where we upper bounded the last term with a multiple of m_{try} in order to relate it to the normal decision tree bound. □

As we shortly show, the DRT splitting procedure satisfies the bounds in Theorem 2.3. Furthermore, since the construction of dimension reduction forests is linear in the number of trees M , we immediately have the following corollary to Theorem 2.3.

Corollary 2.2. *The computational complexity of the dimension reduction forest algorithm is $\mathcal{O}(M m_{try} n \log(n) (m_{try} + \log(n)))$.*

Proof. To apply Theorem 2.3 to our dimension reduction tree algorithm, we must demonstrate that the computational complexity of our splitting procedure satisfies the bounds listed in the theorem. For simplicity, we assume that the dimension reduction tree is built using a SIR (SAVE) splitting rule all the way down the tree and without the variable screening step. In this case, the computation time per node is due to the calculation of the SIR and SAVE directions and the search for the best split. SIR and SAVE both require a QR decomposition, an eigendecomposition of the Λ matrix, and sorting the data set with respect to the response. These operations have a combined complexity of

$$\mathcal{O}(s m_{try}^2 + m_{try}^3 + s \log(s)) = \mathcal{O}(s m_{try}^2 + s \log(s))$$

according to Golub and Loan (2012), where s is the sample size of the node. We used the fact that the first term dominates under the assumption that $m_{try} < s$, which we require due to the limitations of SIR (SAVE). Once these directions are computed, a DRT searches for the best split along the leading direction. This operation is bounded by the theoretical complexity of sorting, i.e., $\mathcal{O}(s \log(s))$. Putting this all together we have that

$$C(s) = \mathcal{O}(s m_{try}^2 + s \log(s)) \tag{2.9}$$

for a dimension reduction tree.

Equation (2.9) allows us to apply Theorem 2.3 to determine the computational complexity of a single DRT. The full dimension reduction forest algorithm is linear in the number of trees M , so that the computational complexity of the DRF algorithm is

$$\mathcal{O}(M m_{try} n \log(n) (m_{try} + \log(n))).$$

□

The time complexity of a standard random forest under similar assumptions (Louppe, 2014) is $\mathcal{O}(Mm_{try}n\log(n)^2)$. If we choose $m_{try} \leq \log(n)$, then the time complexity of a DRF is bounded by the standard random forest’s complexity. Note that the worst-case performance is when $m_{try} = p$. In this case, the DRF’s performance is worse than a standard RF by a factor of $p/\log(n)$, which is small for many applications where $p \ll n$.

Software Implementation

We implemented the dimension reduction forest algorithm in the `drforest` Python package. The core algorithm is written in C++ and uses the efficient `Armadillo` linear algebra library for matrix operations. Furthermore, we parallelized the tree growing and prediction process across multiple threads. Our use of multithreading decreases the run time linearly in the number of available threads. For ease of use, we wrapped the code in a `scikit-learn` style estimator with the `Cython` package. The result is an efficient open-source package compatible with Python’s machine learning ecosystem.

Empirical Run Times

Now, we consider empirical run times to explicitly evaluate the speed of DRFs compared to RFs. Since the tree growing process can be performed in parallel, we focused on the run time of growing a single decision tree. To ensure a fair comparison, we used our code-base for calculating the dimension reduction trees and axis-aligned decision trees.

As a benchmark, we simulated data from Simulation #1 with different sample sizes and number of covariates. We fully grew all trees until the leaf node size $n_{min} = 1$ and each tree considered all features at each split. We evaluated the run times on 10 different sample sizes equally spaced from $n = 150$ to 10,000 and 10 different feature sizes equally spaced from $p = 5$ to 50. The machine used for this comparison was a 2021 MacBook Pro with an Apple M1 Pro processor and a memory of 32 GB. The reported run times were calculated with Python’s `timeit` package as the median over 30 repetitions.

Figure 2.5 shows the median run times as a function of the sample size and the number of features. As expected, the run time of a dimension reduction tree is slower than the corresponding axis-aligned decision tree. We find that DRTs are roughly 2 to 4 times slower with the largest difference being at small sample sizes where the run times of both methods is already less than a tenth of a second. Furthermore, the run time for a DRT is about 0.25 seconds at a sample size of $n = 10,000$, which translates to training a DRF with $M = 500$ trees in less than 15 seconds using 10 threads. As predicted by the theoretical computational complexity analysis, the gap in run time decreases as the sample size increases because the time complexity of sorting begins to dominate the calculation. Interestingly, the run time as a function of the number of covariates is roughly constant for the majority of the $p = 5$ to 50 range. This may change as p increases or in a high-dimensional setting; however, we did not design DRFs for these scenarios. Overall, we find that the DRF’s training time is sufficiently fast for many applied problems of interest.

2.8.3 Incorporating Categorical Covariates

A prominent advantage of random forests is their ability to handle a mixture of continuous and categorical covariates. Here, we propose a modification to the dimension reduction forest algorithm that retains this advantage. Specifically, we propose estimating the SIR and SAVE directions using only the continuous

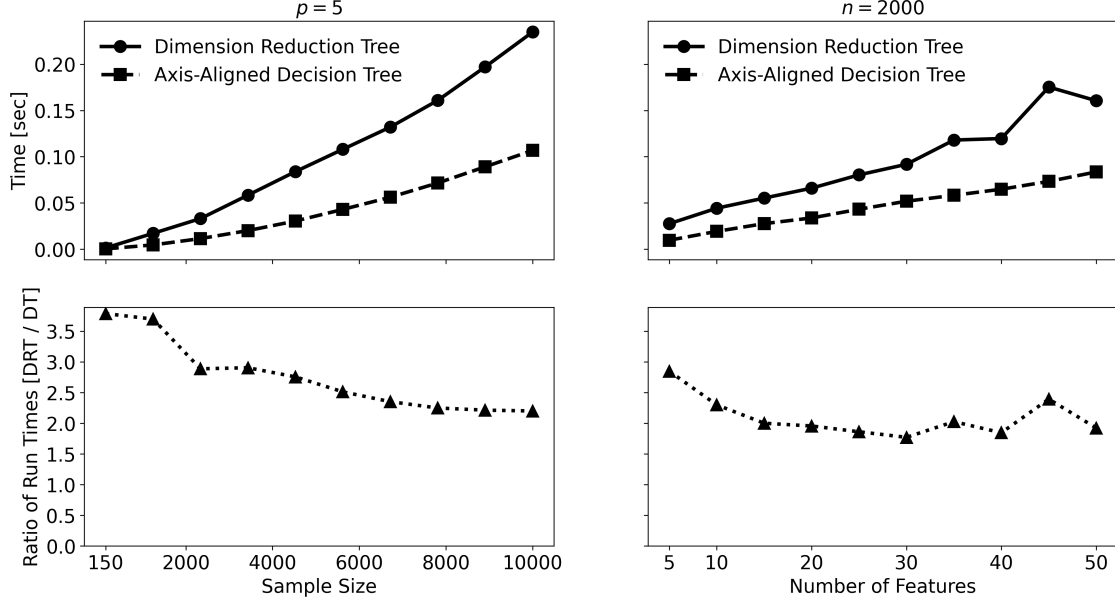


Figure 2.5: (Top) Median empirical run times for a dimension reduction tree (solid curves) and a axis-aligned decision tree (dashed curves) as the sample size (left) or number of features (right) increases. (Bottom) Ratio of the median empirical run time of a dimension reduction tree to a axis-aligned decision tree as a function of sample size (left) or number of features (right).

covariates within the node while considering the categorical covariates as separate splitting variables. The simulation studies and real data analysis in the main text found that this algorithm works well in the presence of a single binary-valued covariate, which is a common occurrence in real data.

Formally, let $\mathbf{W} = (W_1, \dots, W_q)$ be a collection of q categorical covariates. Also, let $\mathbf{Z}_i = (\mathbf{X}_i, \mathbf{W}_i)$ denote the continuous and categorical features, respectively. In the presence of categorical features, we propose defining the set of all partitions $\mathcal{C}$ considered at a CART split (Equation 2.3) as partitions formed by adding a rule of the form $\mathbb{1}_{\{\beta_j^T \mathbf{x} \leq c\}}$ or $\mathbb{1}_{\{w_j \in \mathcal{S}_{W_j}\}}$ where $\mathcal{S}_{W_j}$ is a subset of the set of possible values of W_j . The dimension reduction forest algorithm with this modified splitting rule is outlined in Algorithm 2.5.

In the presence of a single categorical variable, there are explicit connections with the splitting rule proposed in this section and partial dimension reduction (Chiaromonte et al., 2002). In particular, the theoretical results in Section 2.4 could be extended to this setting. In addition, as mentioned in the discussion, semiparametric methods (Y. Ma & Zhu, 2012) can incorporate categorical covariates directly into the SDR estimator; however, these methods are currently too computationally expensive to include in a DRT. As such, we consider fully incorporating categorical features into the DRT’s linear combination split as an area for future research.

2.8.4 Proofs of Theorem 2.1 and Theorem 2.2

Theorem 2.1 verifies that recursive application of SIR within a dimension reduction tree continues to estimate a part of the local central subspace. First for some preliminaries. One can show that Assumption A2.1 implies that

$$\mathbb{E}[\mathbf{X} - \mathbb{E}[\mathbf{X} \mid \mathbf{X} \in A] \mid B_A^T \mathbf{X}, \mathbf{X} \in A] = P_{B_A}(\Sigma_A)^T(\mathbf{X} - \mathbb{E}[\mathbf{X} \mid \mathbf{X} \in A]),$$

Given a training data set $\mathcal{D}_n = \{(\mathbf{Z}_i, Y_i)\}_{i=1}^n = \{(\mathbf{X}_i, \mathbf{W}_i, Y_i)\}_{i=1}^n$, do the following:

1. Draw M bootstrap samples each with sample size n from $\mathcal{D}_n$.
2. For the m th bootstrap sample, where $m \in \{1, \dots, M\}$, fit one DRT model $\hat{T}_m$ using the following rules:
 - (a) *Optional variable screening*: At an internal node A , select the top m_{try} features that minimize Equation (2.3) based on their corresponding axis-aligned splitting rules. Otherwise, set $m_{try} = p$.
 - (b) Estimate the leading sufficient dimension reduction direction $\hat{\beta} \in \mathbb{R}^p$ using SIR ($\hat{\beta}_{\text{SIR}}$) and SAVE ($\hat{\beta}_{\text{SAVE}}$) with the continuous features selected in step (a) and the samples in A as input.
 - (c) Project all observations onto the reduced space by calculating $[\hat{\beta}_{\text{SIR}}, \hat{\beta}_{\text{SAVE}}]^T \mathbf{X}_i$ for each sample i in A . Construct a univariate splitting rule $\mathbb{1}_{\{\hat{\beta}^T \mathbf{X} \leq c\}}$ or $\mathbb{1}_{\{w_j \in S_{W_j}\}}$ by minimizing Equation (2.3) over the two features in the reduced data set and any selected categorical covariates.
 - (d) Recursively apply steps (a)–(c) on each child node until the node’s sample size is smaller than a pre-specified value $n_{\min}$.
 - (e) For each leaf node, set $\mu_u = \bar{Y}_{A_u} = |\{i : \mathbf{Z}_i \in A_u\}|^{-1} \sum_{\{i : \mathbf{Z}_i \in A_u\}} Y_i$.
3. Average the M trees to obtain the final model, $\hat{f}(\mathbf{z}) = M^{-1} \sum_{m=1}^M \hat{T}_m(\mathbf{z})$, with associated kernel $K_{\text{DRF}}(\mathbf{z}_0, \mathbf{z}_1)$ defined in Equation (2.1).

Algorithm 2.5: Dimension reduction forests with categorical covariates.

where $P_{B_A}(\Sigma_A) = B_A(B_A^T \Sigma_A B_A)^{-1} B_A^T \Sigma_A$ is the projection matrix onto the central subspace $S_{Y|\mathbf{X}, \mathbf{X} \in A}$ (Lemma 1.1 in B. Li (2018)). Also, if Assumption A2.1 and Assumption A2.2 both hold, then one can show that $\text{Var}(\mathbf{X} | B_A^T \mathbf{X}, \mathbf{X} \in A) = \Sigma_A Q$, where $Q = I_p - P_{B_A}(\Sigma_A)$ (Corollary 5.1 in B. Li (2018)). We are now ready to prove Theorem 2.1 and Theorem 2.2.

Proof of Theorem 2.1. Without loss of ambiguity, all expectations in this proof are conditioned on $\mathbf{X}$ being contained in the parent node A . The proof is little more than an application of the law of total expectation. We only carry out the proof for the left child node A_L . The proof for the right child node A_R is exactly the same. Without loss of generality, assume $\mathbb{E}[\mathbf{X} | \mathbf{X} \in A] = 0$. We have that

$$\begin{aligned}
\mathbb{E}[\mathbf{X} | Y, \mathbf{X} \in A_L] &= \mathbb{E}[\mathbf{X} | Y, \beta^T \mathbf{X} \leq c], \\
&= \mathbb{E}[\mathbb{E}[\mathbf{X} | Y, \beta^T \mathbf{X} \leq c, B_A^T \mathbf{X}] | Y, \beta^T \mathbf{X} \leq c], \\
&= \mathbb{E}[\mathbb{E}[\mathbf{X} | B_A^T \mathbf{X}] | Y, \beta^T \mathbf{X} \leq c], \\
&= P_{B_A}(\Sigma_A)^T \mathbb{E}[\mathbf{X} | Y, \beta^T \mathbf{X} \leq c], \\
&= \Sigma_A P_{B_A}(\Sigma_A) \Sigma_A^{-1} \mathbb{E}[\mathbf{X} | Y, \beta^T \mathbf{X} \leq c], \\
&= \Sigma_A P_{B_A}(\Sigma_A) \Sigma_A^{-1} \mathbb{E}[\mathbf{X} | Y, \mathbf{X} \in A_L].
\end{aligned}$$

Multiplying both sides by Σ_A^{-1} completes the proof. The third line is due to the fact that $Y \perp \mathbf{X} | B_A^T \mathbf{X}$ (the postulated model within the node) and that $\beta^T \mathbf{X} \perp \mathbf{X} | B_A^T \mathbf{X}$ since $\beta \in \text{span}(B_A)$. We also used the easily verifiable identity $P_{B_A}(\Sigma_A)^T = \Sigma_A^{-1} P_{B_A}(\Sigma_A) \Sigma_A^{-1}$. $\square$

Theorem 2.2 is the corresponding theorem for SAVE. It says that recursive application of SAVE within a dimension reduction tree continues to estimate a part of the local central subspace.

Proof of Theorem 2.2. Without loss of ambiguity, all expectations in this proof are conditioned on $\mathbf{X}$ being

contained in the parent node A . Once again, we only carry out the proof for the left child node A_L . The proof for the right child node A_R is exactly the same. Without loss of generality, we assume $\mathbb{E}[\mathbf{X} \mid \mathbf{X} \in A] = 0$. Similar to the proof of Theorem 2.1, we just apply the law of total variation:

$$\begin{aligned}
\text{Var}(\mathbf{X} \mid Y, \mathbf{X} \in A_L) &= \text{Var}(\mathbf{X} \mid Y, \beta^T \mathbf{X} \leq c), \\
&= \mathbb{E}[\text{Var}(\mathbf{X} \mid Y, \beta^T \mathbf{X} \leq c, B_A^T \mathbf{X}) \mid Y, \beta^T \mathbf{X} \leq c] + \\
&\quad \text{Var}(\mathbb{E}[\mathbf{X} \mid Y, \beta^T \mathbf{X} \leq c, B_A^T \mathbf{X}] \mid Y, \beta^T \mathbf{X} \leq c), \\
&= \mathbb{E}[\text{Var}(\mathbf{X} \mid B_A^T \mathbf{X}) \mid Y, \beta^T \mathbf{X} \leq c] + \text{Var}(\mathbb{E}[\mathbf{X} \mid B_A^T \mathbf{X}] \mid Y, \beta^T \mathbf{X} \leq c), \\
&= \Sigma_A (I_p - P_{B_A}(\Sigma_A)) + P_{B_A}(\Sigma_A)^T \text{Var}(\mathbf{X} \mid Y, \beta^T \mathbf{X} \leq c) P_{B_A}(\Sigma_A), \\
&= \Sigma_A (I_p - P_{B_A}(\Sigma_A)) + P_{B_A}(\Sigma_A)^T \text{Var}(\mathbf{X} \mid Y, \mathbf{X} \in A_L) P_{B_A}(\Sigma_A).
\end{aligned}$$

Subtracting Σ_A from both sides and applying the identity $\Sigma_A P_{B_A}(\Sigma_A) = P_{B_A}(\Sigma_A)^T \Sigma_A P_{B_A}(\Sigma_A)$, we have

$$\text{Var}(\mathbf{X} \mid Y, \mathbf{X} \in A_L) - \Sigma_A = \Sigma_A P_{B_A}(\Sigma_A) \Sigma_A^{-1} (\text{Var}(\mathbf{X} \mid Y, \mathbf{X} \in A_L) - \Sigma_A) P_{B_A}(\Sigma_A),$$

where the right hand side is in $\Sigma_A \text{span}(B_A)$ as desired. $\square$

2.8.5 Simulation Study and Data Set Details

In this section, we elaborate on various details of the simulation studies and real data analysis presented in the main text. Table 2.3 contains the hyperparameter settings used in the simulation studies of Section 2.5. Also, the three Friedman functions and the full definitions of Simulations 1 – 7 used to test predictive performance of DRFs are

- *Friedman 1*: $Y = 10 \sin(\pi X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + \varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} U[0, 1]^{10}$.
- *Friedman 2*: $Y = \left(X_1^2 + \left[X_2 X_3 - \frac{1}{X_2 X_4} \right]^2 \right)^{1/2} + \varepsilon$, where $\mathbf{X}_i$ are uniform over the hyper-rectangle $[0, 100] \times [20, 280] \times [0, 1] \times [1, 11]$.
- *Friedman 3*: $Y = \tan^{-1} \left(\frac{X_2 X_3 - \frac{1}{X_2 X_4}}{X_1} \right) + \varepsilon$, where $\mathbf{X}_i$ are uniform over the hyper-rectangle $[0, 100] \times [20, 280] \times [0, 1] \times [1, 11]$.
- *Simulation 1*: $Y = 20 \max \left\{ e^{-2(X_1 - X_2)^2}, 2e^{-0.5(X_1^2 + X_2^2)}, e^{-(X_1 + X_2)^2} \right\} + \varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} U[-3, 3]^5$.
- *Simulation 2*: $Y = 20 \max \left\{ e^{-18X_1^2}, e^{-18X_2^2}, 1.75e^{-20(X_1 + X_2)^2}, 1.75e^{-20(X_1 - X_2)^2} \right\} + \varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} U[-1, 1]^4$.
- *Simulation 3*: $Y = (\mathbf{X}^T \beta_1)^2 + (\mathbf{X}^T \beta_2)^2 + 0.5\varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} N(\mathbf{0}_{12}, \Sigma)$, $\beta_1 = (1, 1, 1, 1, 1, 1, \mathbf{0}_6^T)^T / \sqrt{6}$ and $\beta_2 = (1, -1, 1, -1, 1, -1, \mathbf{0}_6^T)^T / \sqrt{6}$.
- *Simulation 4*: $Y = \mathbf{X}^T \beta_3 (\mathbf{X}^T \beta_4)^2 + (\mathbf{X}^T \beta_5)(\mathbf{X}^T \beta_6) + 0.5\varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} N(\mathbf{0}_{10}, \Sigma)$, $\beta_3 = (1, 2, 3, 4, \mathbf{0}_6^T)^T / \sqrt{30}$, $\beta_4 = (-2, 1, -4, 3, 1, 2, \mathbf{0}_4^T)^T / \sqrt{35}$, $\beta_5 = (\mathbf{0}_4^T, 2, -1, 2, 1, 2, 1)^T / \sqrt{15}$, and $\beta_6 = (\mathbf{0}_6^T, -1, -1, 1, 1)^T / 2$.
- *Simulation 5*: $Y = \mathbf{X}^T \beta_7 (\mathbf{X}^T \beta_8)^2 + (\mathbf{X}^T \beta_9)(\mathbf{X}^T \beta_{10}) + 0.5\varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} N(\mathbf{0}_2, \Sigma)$, $\beta_7 = (1, 3)^T / \sqrt{10}$, $\beta_8 = (-2, 1)^T / \sqrt{5}$, $\beta_9 = (3, -4)^T / \sqrt{25}$, and $\beta_{10} = (-1, -1)^T / \sqrt{2}$.

- *Simulation 6*: $Y = \sum_{c=0}^1 [\gamma_c(\mathbf{X}^T \beta)^2 + \alpha_c] \mathbb{1}_{\{W=c\}} + 0.5\varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} N(\mathbf{0}_3, \Sigma)$, $W_i \stackrel{\text{iid}}{\sim} \text{Bernoulli}(0.5)$, $\beta = (1, -2, 1)^T / \sqrt{6}$, $\alpha_0 = 1$, $\alpha_1 = 2$, $\gamma_0 = 2$, and $\gamma_1 = -2$.
- *Simulation 7*: $Y = \sum_{c=0}^1 [\gamma_c(\mathbf{X}^T \beta_c)^2 + \alpha_c] \mathbb{1}_{\{W=c\}} + 0.5\varepsilon$, where $\mathbf{X}_i \stackrel{\text{iid}}{\sim} N(\mathbf{0}_3, \Sigma)$, $W_i \stackrel{\text{iid}}{\sim} \text{Bernoulli}(0.5)$, $\beta_0 = (1, -2, 1)^T / \sqrt{6}$, $\beta_1 = (-1, -1, 2)^T / \sqrt{6}$, $\alpha_0 = 1$, $\alpha_1 = 2$, $\gamma_0 = 2$, and $\gamma_1 = -2$.

We sampled $\varepsilon_i \stackrel{\text{iid}}{\sim} N(0, 1)$ and set $\Sigma_{ij} = 0.5^{|i-j|}$ for all simulations. Lastly, Table 2.4 displays the sample size and number of covariates of each data set used in the real data analysis of Section 2.6.

NW Kernel	We use a Gaussian kernel as the kernel function. The data is standardized along each dimension. The bandwidth is then set to the recommendation in Silverman (1986): $(4/(p+2))^{1/(p+4)} n^{-1/(p+4)}$.
SIR + NW Kernel	Estimates a global dimension reduction subspace using SIR with $d = p$ and the number of slices set to 10. Fits an NW Kernel to the projection of the data on to this subspace with the same settings as NW Kernel.
SAVE + NW Kernel	Estimates a global dimension reduction subspace using SAVE with $d = p$ and the number of slices set to 10. Fits an NW Kernel to the projection of the data on to this subspace with the same settings as NW Kernel.
RF	A total of 12 parameter settings. $M = 500$, $m_{try} = 2, 4, 6, 1/3, \sqrt{p}, p$ and $n_{min} = 1, 5$.
SIR + RF	Estimates a global dimension reduction subspace using SIR with $d = p$ and the number of slices set to 10. Fits an RF model to the projection of the data onto this subspace with the same settings as RF.
SAVE + RF	Estimates a global dimension reduction subspace using SAVE with $d = p$ and the number of slices set to 10. Fits an RF model to the projection of the data onto this subspace with the same settings as RF.
DRF	A total of 12 parameter settings. $M = 500$, $m_{try} = 2, 4, 6, 1/3, \sqrt{p}, p$ and $n_{min} = 1, 5$. The number of slices used to estimate SIR and SAVE is set to 10.

Table 2.3: Hyperparameter settings.

2.8.6 Sensitivity to the Number of Trees

Here, we investigate the performance of the dimension reduction forest as the number of trees increases. The goal is to justify our choice of $M = 500$ trees in the predictive performance benchmarks. Using the individual tree predictions from Section 2.5.1, we computed the MSE on the test set for different ensemble sizes. Figure 2.6 and Figure 2.7 summarizes the results. As expected, it is always beneficial to use more trees, but there are drastically diminishing returns after a point. The MSE on the test set begins to saturate at various values — all less than 100 trees — for both DRFs and RFs. Furthermore, the advantage of DRFs over RFs is maintained across all ensemble sizes for the majority of these data sets. Overall, we conclude that our use of $M = 500$ is more than enough trees for the performance of both DRFs and RFs to stabilize. Also, Figure 2.7 shows that on the order of 10 dimension reduction trees are needed to match the test error of 500 axis-aligned decision trees on the simulated data sets. As such, under a restricted computational budget, one may be able to use a smaller DRF that matches the accuracy of a larger RF grown in the same amount of time.

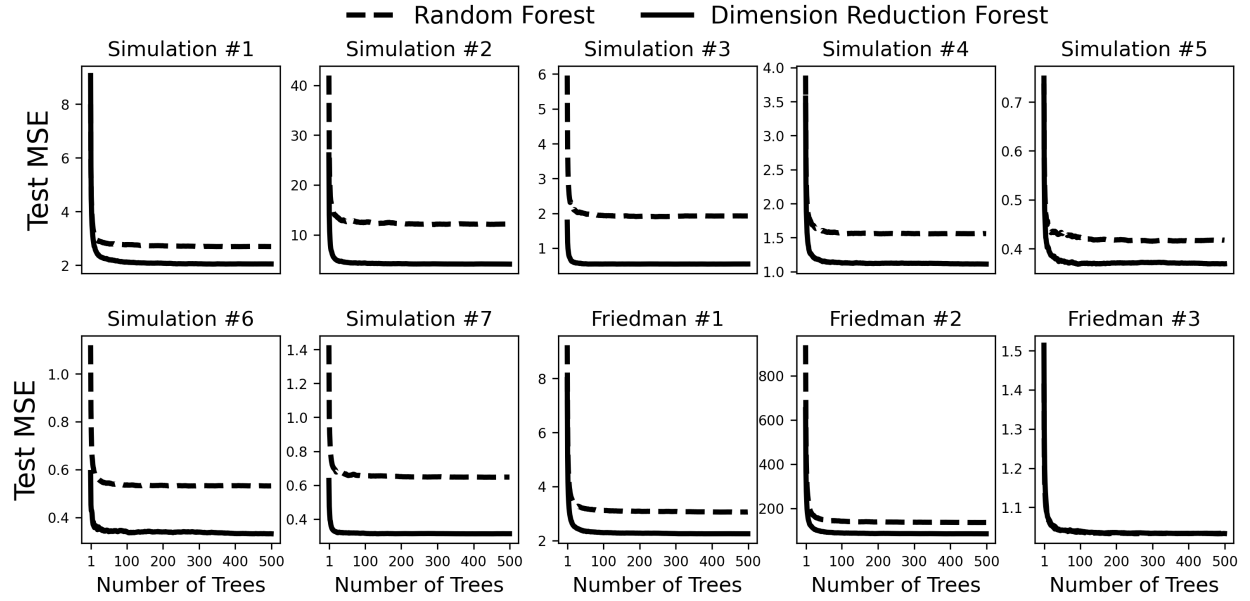


Figure 2.6: Median test MSE as a function of the number of trees for a dimension reduction forest (black curves) and a random forest (dashed curves) on the simulated data sets.

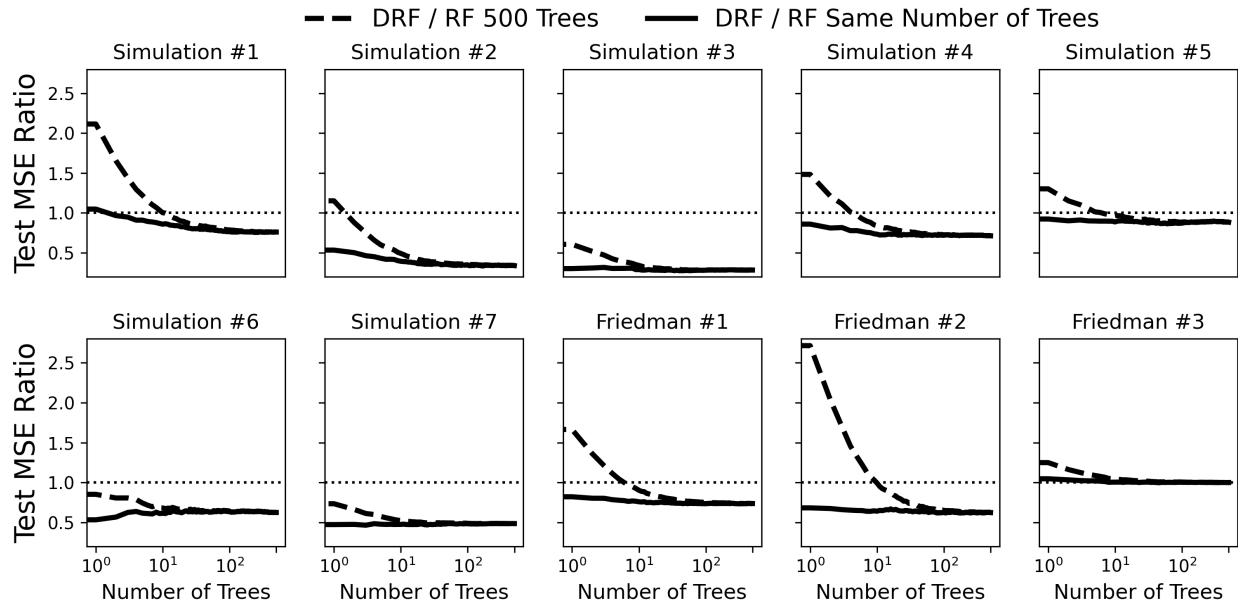


Figure 2.7: Ratio of the median test MSE between a DRF and an RF of the same size (black curves) and a DRF with a varying number of trees compared to an RF with 500 trees (dashed curves) as a function of the number of trees on a logarithmic scale.

Data Set Name	Number of Samples	Number of Features
Abalone	4177	8
Athletes	202	9
Body Fat	245	14
CPU Small	8192	12
Diabetes	442	10
Fish Catch	158	7
Kin8nm	8192	8
Auto Price	159	15
Liver	345	5
Mu284	284	9
Puma32H	8192	32
Puma8NH	8192	8
Wisconsin	194	32
Bank8FM	8192	8

Table 2.4: Metadata for data sets analyzed in the real data analysis.

2.8.7 Sensitivity to the Number Slices

Here, we investigate the sensitivity of the DRF algorithm to the number of slices used to estimate SIR and SAVE. Note that it has been empirically and theoretically demonstrated that SIR is insensitive to the number of slices and the way of making slices (K.-C. Li, 1991; Zhu & Fang, 1996; Zhu & Ng, 1995); however, this robustness is not shared by SAVE. As such, we investigate DRF’s sensitivity to the number of slices chosen for estimation. We used the same simulation setup outlined in Section 2.5.1, but we varied the number of slices used by the DRF such that $n_{slice} \in \{5, 10, 15, 20\}$. Recall that the number of slices used in the original simulation study was $n_{slice} = 10$.

The average test MSE of the four DRFs on the simulated data sets is displayed in Table 2.5. For some data sets, we observe some variation in the performance of the DRF as the number of slices changes. This is particularly apparent on the Friedman #2 data set where the DRF’s performance decreases as n_{slices} increases. However, the change in performance is on the order of the estimates standard deviation. Overall, our experience leads us to conclude that while DRFs are somewhat sensitive to the number of slices, properly tuning the hyperparameters shared by a standard random forest results in a much more substantial performance improvement.

2.8.8 Additional Results and Figures

In this section, we include the results of a simulation study designed to assess LPD estimation under an increasing number of uninformative covariates and additional figures from the analysis of air pollution in Beijing, China.

Additional Simulation Studies

Here, we include a simulation study designed to measure how LPD estimation performed under an increasing number of uninformative covariates. We used the same setup as described in Section 2.5.2; however, here we set fixed σ^2 so that the signal-to-noise ratio was 3:1 and varied $p = 2, 5, 10, 15$, and 20. Figure 2.8 displays the boxplots of the trace correlations achieved by the four methods at $p = 2, 5, 10, 15, 20$, respectively. For

	DRF ($n_{slices} = 5$)	DRF ($n_{slices} = 10$)	DRF ($n_{slices} = 15$)	DRF ($n_{slices} = 20$)
Simulation #1	2.07 (0.14)	2.06 (0.12)	2.07 (0.13)	2.07 (0.13)
Simulation #2	4.23 (0.51)	4.13 (0.46)	4.21 (0.49)	4.28 (0.49)
Simulation #3	0.57 (0.18)	0.60 (0.18)	0.62 (0.19)	0.63 (0.19)
Simulation #4	1.04 (0.21)	1.16 (0.22)	1.23 (0.23)	1.28 (0.24)
Simulation #5	0.39 (0.06)	0.38 (0.06)	0.38 (0.06)	0.38 (0.07)
Simulation #6	0.34 (0.06)	0.34 (0.06)	0.34 (0.06)	0.34 (0.06)
Simulation #7	0.33 (0.07)	0.33 (0.08)	0.33 (0.08)	0.33 (0.08)
Friedman #1	2.23 (0.13)	2.26 (0.13)	2.28 (0.14)	2.29 (0.13)
Friedman #2	78.63 (9.80)	85.63 (10.32)	87.91 (10.78)	88.33 (10.58)
Friedman #3	1.04 (0.05)	1.03 (0.05)	1.03 (0.05)	1.03 (0.05)

Table 2.5: Simulation results testing DRF’s sensitivity to the number slices used to estimate SIR and SAVE. The table’s entries contain the average test MSE on each simulated data set. The standard deviation over the 50 runs is displayed in parentheses.

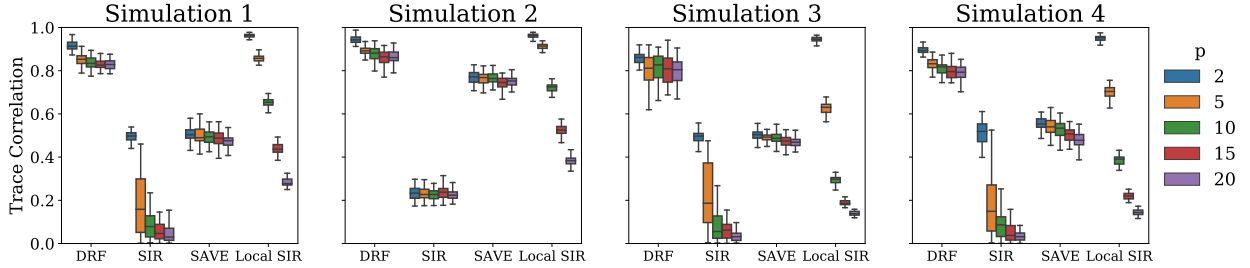


Figure 2.8: Boxplots of the trace correlations between the local gradients and the LPDs estimated by the four methods outlined in Section 2.5.2 for different numbers of covariates. For each method, the boxplots are sorted in increasing order by the number of covariates, which matches the order in the legend. The true regression functions only utilized the first two covariates. Higher values of the trace correlation are better.

all but the $p = 2$ cases, the DRF method performs best. As expected, Local SIR performs best when only informative covariates are present; although, the DRF estimates are still competitive. Furthermore, as p increases, Local SIR quickly drops to the worst-performing method due to the curse of dimensionality. In contrast, the DRF estimates remain accurate.

Additional Figures

Figure 2.9 displays the pairwise scatter plots of the meteorological features included in Section 2.6.2. The points are color coded by the response variable’s value. Figure 2.10 contains the global permutation-based variable importance calculated using a standard random forest and a dimension reduction forest trained on the data set.

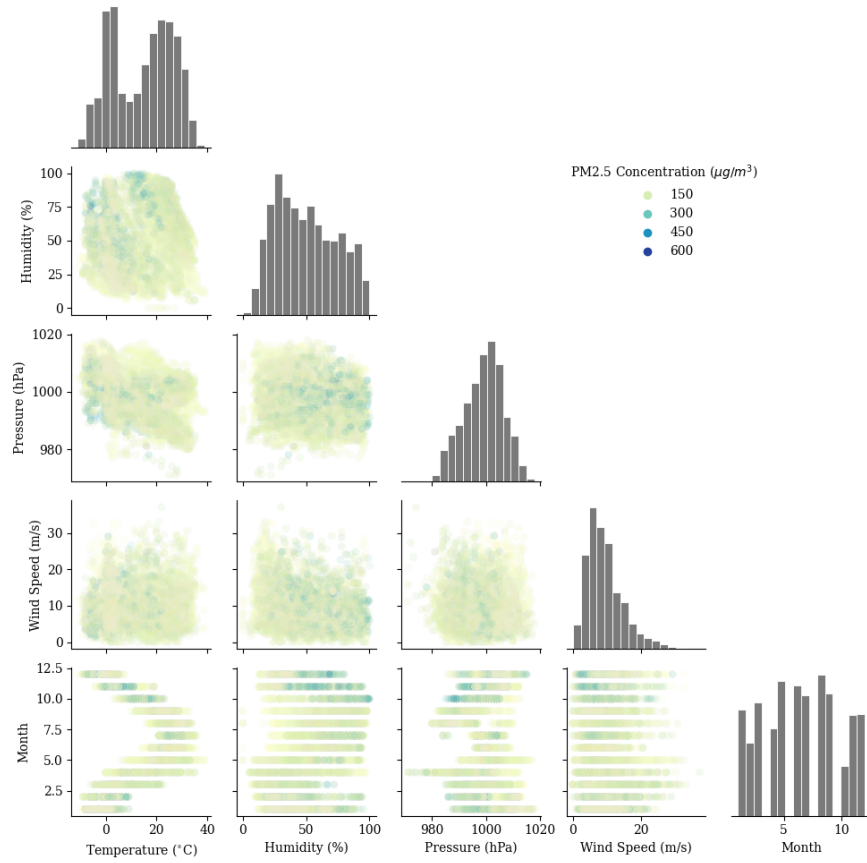


Figure 2.9: Pairwise scatter plots of meteorological variables.

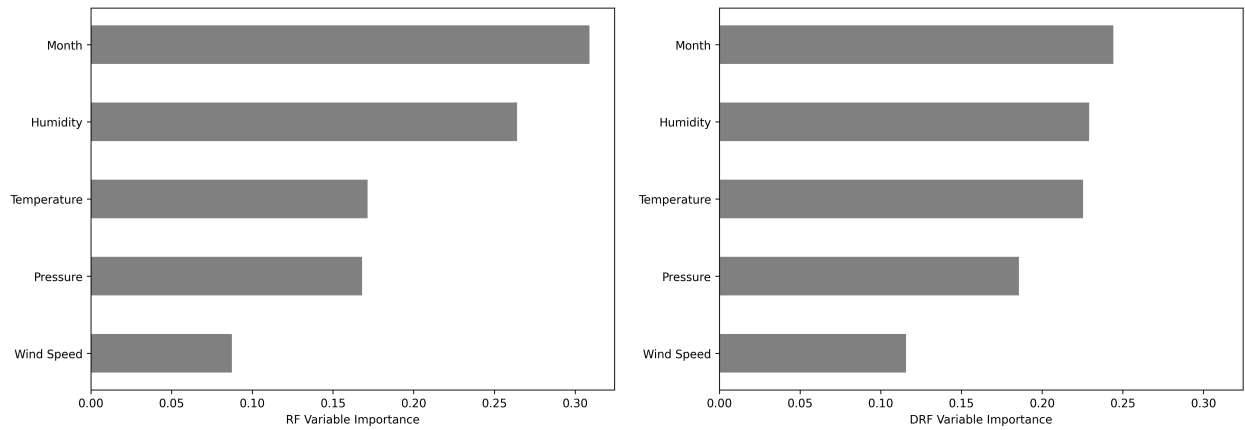


Figure 2.10: Global permutation-based variable importance from a traditional random forest (left) and a dimension reduction forest (right) trained on the PM2.5 concentration data set.

Chapter 3

Modeling Evolving Community Structure in Dynamic Networks

3.1 Introduction

Many naturally occurring networks contain discrete changes in community structure.¹ When high school students move across the country for college, old friendship groups often dissolve, which leads to the formation of new collegiate friendship groups. After World War II, the Eastern and Western Blocs emerged and dominated the network of global alliances. However, after the fall of the Soviet Union, these blocs reshuffled into new political alliances. When exposed to external stimuli, regions of the brain activate before becoming dormant. By identifying these community-level phase changes, we can gain valuable insight into the rich processes that generate dynamic (longitudinal or time-varying) networks.

In this chapter, we address this problem of inferring discrete changes in a network’s community structure. See Figure 3.1 for a concrete example. In this case, a dynamic network’s community structure contains two groups during the first three time points, splits into six groups at the fourth time point, and then merges into four groups during the seventh time point. These additions, deletions, splits, and mergers of groups in dynamic networks is our primary object of interest. For brevity, we refer to these group-level dynamics as a network’s *evolving community structure*. Note that this does not refer to individual actors (or nodes) moving between existing communities. While we will model such actor-level dynamics, our primary goal is to infer changes in community structure that occur on a larger macro scale.

We adopt the latent space approach to network modeling to infer an evolving community structure in dynamic networks. Initially developed in P. D. Hoff et al. (2002), latent space models (LSMs) embed actors within a Euclidean space (the distance model) or a hypersphere (the projection model). Closeness in the latent space increases the probability that the two actors form an edge in the observed network. For this reason, one interprets the proximity of two actors in the latent space as an indication that they have similar characteristics. The LSM’s popularity stems from the intuitive meaning of the embeddings and its ability to naturally incorporate desirable sociological features such as homophily, reciprocity, and transitivity. Sarkar and Moore (2005) and Sewell and Chen (2015b) extended the distance model to undirected and directed dynamic networks, respectively.

¹This chapter is based on research published as: Loyal, J. D., and Chen, Y. (in press). A Bayesian nonparametric latent space approach to modeling evolving communities in dynamic networks. *Bayesian Analysis*.

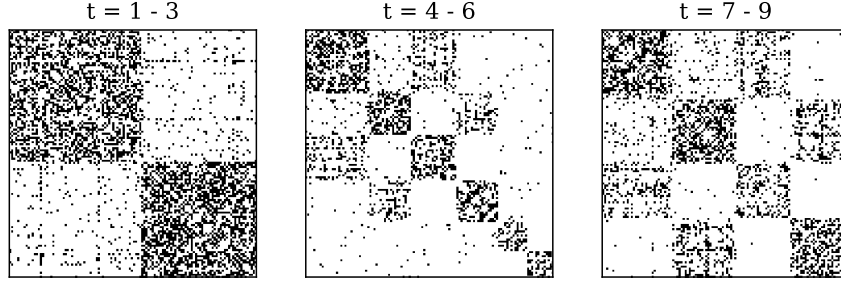


Figure 3.1: Adjacency matrices from a dynamic sequence of networks that exhibit an evolving community structure. At $t = 4$ the groups split from two into six groups, then at $t = 7$ the groups merge into four communities. The rows and columns of each adjacency matrix have been sorted according to the ground truth labels. These networks were simulated according to the procedure described in Section 3.5.4.

The latent space model was first applied to the community detection problem in the case of a single static network by Handcock et al. (2007). Their proposed latent position cluster model (LPCM) uses a Gaussian mixture model to cluster the latent positions embedded according to the Euclidean distance model. This approach assumes that a cluster of actors in the latent space corresponds to a densely connected community in the network. Sewell and Chen (2017) extended the LPCM to dynamic networks in which the number of communities does not change over time. Their approach allows the actors to transition between these fixed communities by endowing the actor’s latent trajectories with autoregressive hidden Markov model (AR-HMM) dynamics.

The following two limitations make the existing LPCM for dynamic networks inadequate for inferring an evolving community structure: (1) the number of communities must remain fixed over time, and (2) the number of communities must be set a priori. The first shortcoming implies that the existing LPCM cannot allow the number of communities to change over time so that inference over evolving communities is not possible. The second shortcoming means that uncertainty in the community structure is not assessed in a fully Bayesian way, which is essential in most applications where multiple clusterings fit the data well. The inability of LPCMs to handle time-varying groups in dynamic networks motivates our investigation into a suitable Bayesian methodology.

In this chapter, we propose a Bayesian nonparametric model that addresses the shortcomings outlined in (1) and (2). As in Sewell and Chen (2017), we model the actor’s trajectories through the latent space as a collection of independent and identically distributed (iid) AR-HMMs with Gaussian emissions. However, unlike their approach, we remove the restriction that the hidden label Markov chain is homogeneous. Instead, we propose a model with time-inhomogenous transition distributions. This modification provides us with the flexibility to model the additions, deletions, splits, and mergers of communities. Furthermore, we place a hierarchical Dirichlet process (HDP) prior on these time-inhomogenous transition distributions. This nonparametric prior produces fully Bayesian inference over the number of communities. Since our approach combines the advantages of the HDP with the LPCM for modeling dynamic networks, we refer to our model as the hierarchical Dirichlet process latent position cluster model (HDP-LPCM). To the best of our knowledge, there is no other latent space approach that accomplishes our modeling criteria.

The rest of this chapter is organized as follows. Section 3.2 reviews other Bayesian nonparametric approaches for modeling evolving communities. In Section 3.3, we describe the hierarchical Dirichlet process latent position cluster model (HDP-LPCM), our proposed Bayesian nonparametric method for modeling

evolving communities in dynamic networks. In Section 3.4, we outline our Metropolis-Hastings within Gibbs estimation procedure and our methodology for posterior summarization. We demonstrate the empirical performance of our proposed method through various simulation studies in Section 3.5. In Section 3.6, we use our model to analyze two real-world dynamic networks: a network of military alliances during the Cold War (Gibler, 2009b) and a narrative network constructed from the Game of Thrones television series (Beveridge & Chemers, 2018). Section 3.7 contains a discussion. Lastly, the supplementary material in Section 3.8 contains in-depth derivations of the sampling algorithm, implementation details, further simulation studies and real data analysis, and additional figures.

3.2 Related Methods

Although the latent space methodology has not previously incorporated an evolving community structure, the statistical modeling of varying communities has a long history in the broader network literature. For brevity, we focus on Bayesian nonparametric approaches because they are most related to our work. We restrict the discussion to the three most popular statistical network models: stochastic block models (SBMs), mixed-membership stochastic block models (MMSBMs), and latent feature models.

For stochastic block models, Kemp et al. (2006) introduced the infinite relational model (IRM), which uses a Dirichlet process to infer the number of blocks in the traditional stochastic block model. Ishiguro et al. (2010) extended this work to longitudinal networks in their dynamic infinite relational model (dIRM). Similar to our work, the dIRM assumes the block memberships form a time-inhomogeneous Markov chain. For multiplex networks, Paez et al. (2019) proposed the Hierarchical SBM (HSBM), which uses an HDP to pool community structure across layers. Although the authors apply the HSBM to dynamic networks, the model fails to capture essential autocorrelations in an actor’s community assignments by assuming they are independent across time.

In the mixed-membership literature, X. Fan et al. (2015) introduced the dynamic infinite mixed-membership stochastic block model. Their model also utilizes the hierarchical Dirichlet process framework to extend the original mixed-membership model (Airoldi et al., 2008; Fu et al., 2009; Ho et al., 2011) to the dynamic setting. In contrast to our method, they do not assume an HMM structure and only use the HDP to re-sample the mixed-membership vectors at every time step.

Finally, in the latent feature modeling literature, M. Kim and Leskovec (2013) introduced the nonparametric multi-group membership model. They utilize a distance-dependent Indian buffet process (dd-IBP) that assigns each actor a latent binary feature vector. This nonparametric prior can infer an evolving number of binary features.

While some of these competing approaches have addressed shortcomings (1) and (2), LPCMs have many appealing advantages that make our extension worthwhile. Foremost is the LPCM’s ability to capture network structures on multiple scales. Unlike SBMs, where the same within and between block probabilities are shared by all actors in a group, LPCMs allow for heterogeneous connectivity patterns due to an actor’s local position. Furthermore, unlike other methods such as MMSBMs, the latent space provides an interpretable visualization of the entire network. These visualizations allow domain experts to make essential assessments and critiques of the statistical methodology.

3.3 The HDP Latent Position Cluster Model

We consider binary relational data between n individuals recorded over T time periods. These relations are collected in a sequence of $n \times n$ binary adjacency matrix Y_t , $t = 1, \dots, T$, where the entries Y_{ijt} indicate the presence ($Y_{ijt} = 1$) or absence ($Y_{ijt} = 0$) of an edge between individuals i and j at time t . For clarity, we only consider undirected random graphs without self-loops, so that Y_t is a symmetric matrix with zeros on the diagonal. The extension of our model to directed networks or networks with weighted edges is straightforward and discussed in Section 3.7.

To each of the n individuals, we associate a latent (unobserved) position that may vary through time in a p -dimensional Euclidean latent space. We represent the latent position of individual i at time t with the vector-valued random variable $\mathbf{X}_t^i \in \mathbb{R}^p$. In addition, we collect a snapshot of all individual latent positions at time t in the $n \times p$ matrix $\mathcal{X}_t = (\mathbf{X}_t^1, \dots, \mathbf{X}_t^n)'$. Like the traditional LPCM, we assign each actor to a latent group at each time point. Note that their assignment may change over time. However, unlike the LPCM, we assume that the number of groups changes over time to accommodate the network's evolving community structure. We use G to denote the total number of groups in the network over all observational periods. We refer to a group that contains at least one actor at time t as an *active group at time t* . We use $G_t \subseteq \{1, \dots, G\}$ to indicate the set of all active groups at time t . Note that under the assumption of an evolving community structure, G_t is a random set and may grow or shrink over time. We encode the latent group membership of individual i at time t with the integer-valued random variable $Z_t^i \in G_t$. The collection of group assignments for all individuals at time t is summarized by the n -dimensional vector $\mathcal{Z}_t = (Z_t^1, \dots, Z_t^n)'$.

Common to the latent space literature, we assume that the latent community labels only play a role in the distribution of the latent positions. Formally, we assume that the formation of edges in the dynamic networks $Y_{1:T}$ is conditionally independent of the actor labels $\mathcal{Z}_{1:T}$ given the latent positions $\mathcal{X}_{1:T}$, i.e., $Y_{1:T} \perp \mathcal{Z}_{1:T} \mid \mathcal{X}_{1:T}$. Note that throughout the remainder of this work, we will use the notation $A_{1:K}$ to refer to the sequence $(A_1, A_2, \dots, A_K)$. This allows us to decompose the joint probability distribution as follows:

$$\mathbb{P}(Y_{1:T}, \mathcal{X}_{1:T}, \mathcal{Z}_{1:T}) = \mathbb{P}(Y_{1:T} \mid \mathcal{X}_{1:T}) \mathbb{P}(\mathcal{X}_{1:T}, \mathcal{Z}_{1:T}). \quad (3.1)$$

This independence assumption says that the probability of a tie is solely determined by the underlying latent positions of the actors. This decomposition is consistent with the latent space clustering idea described earlier. Specifically, the notion that an underlying cluster of actors in the latent space results in observed communities. We believe such a generative model is natural for modeling communities in networks. For example, friend groups often form due to their members having similar interests or characteristics.

3.3.1 The Euclidean Distance Model

As in P. D. Hoff et al. (2002), we posit that the probability of an edge forming between actors increases as the Euclidean distance between actors decreases. Let $d_{ijt} = \|\mathbf{X}_t^i - \mathbf{X}_t^j\|_2$ denote the Euclidean distance between actors i and j at time t . A conventional link between the conditional probability of forming an edge and d_{ijt} is the logistic regression model:

$$\mathbb{P}(Y_{ijt} = 1 \mid \mathcal{X}_t, \psi) = \text{logit}^{-1}(\eta_{ijt}) = \frac{\exp(\eta_{ijt})}{1 + \exp(\eta_{ijt})}, \quad (3.2)$$

where η_{ijt} is a linear predictor that depends on the distances d_{ijt} and the vector $\boldsymbol{\psi}$ holds any additional parameters. A sequence of conditional independence assumptions results in the full network likelihood. First, we assume that the longitudinal networks are conditionally independent given the latent positions, i.e., $Y_t \perp Y_s \mid \mathcal{X}_{1:T}$ for all $s, t \in \{1, \dots, T\}$. Second, we posit that edges between actors form independently conditioned on their latent positions at each time point. Under these assumptions, the likelihood of the adjacency matrices factors as a product over the networks at each time point and the set of dyads $\mathcal{D}$:

$$\mathbb{P}(Y_{1:T} \mid \mathcal{X}_{1:T}, \boldsymbol{\psi}) = \prod_{t=1}^T \prod_{(i,j) \in \mathcal{D}} \mathbb{P}(Y_{ijt} = y_{ijt} \mid \mathcal{X}_t, \boldsymbol{\psi}) = \prod_{t=1}^T \prod_{(i,j) \in \mathcal{D}} \frac{\exp(y_{ijt}\eta_{ijt})}{1 + \exp(\eta_{ijt})}. \quad (3.3)$$

In this work, we focus on undirected networks without self-loops for which

$$\mathcal{D} = \{(i, j) : 1 \leq i \leq n, j < i\} \quad (3.4)$$

and the linear predictor takes the form

$$\eta_{ijt} = \beta_0 - d_{ijt} \quad (3.5)$$

so that $\boldsymbol{\psi} = \{\beta_0\}$. Various versions of this likelihood have been proposed, such as in Sarkar and Moore (2005), Krivitsky et al. (2009), and Sewell and Chen (2015b). The intercept parameter β_0 is sometimes referred to as the abundance. This is because higher values of β_0 result in a higher probability of forming edges.

3.3.2 An AR-HMM for Latent Space Dynamics

Recall that our goal is to endow the dynamics of the latent space with a probabilistic structure that allows for inference over an evolving collection of communities. To accomplish this goal, we must place an adequately flexible joint distribution over the latent space, $\mathbb{P}(\mathcal{X}_{1:T}, \mathcal{Z}_{1:T})$. To begin, we adopt the assumptions taken by Sewell and Chen (2017). Specifically, we assume that an actor's community assignments form a Markov chain, i.e.,

$$Z_t^i \mid Z_1^i, \dots, Z_{t-1}^i \stackrel{d}{=} Z_t^i \mid Z_{t-1}^i, \quad (3.6)$$

where $\stackrel{d}{=}$ denotes equality in distribution. Similarly, we assume an actor's latent position follows a Markov process with an additional dependence on the current community assignment, i.e.,

$$\mathbf{X}_t^i \mid \mathbf{X}_1^i, \dots, \mathbf{X}_{t-1}^i, Z_1^i, \dots, Z_{t-1}^i, Z_t^i \stackrel{d}{=} \mathbf{X}_t^i \mid \mathbf{X}_{t-1}^i, Z_t^i. \quad (3.7)$$

These two assumptions allow us to factor the marginal density of an individual actor's trajectory. In particular, we conclude that an actor's trajectory follows an independent autoregressive hidden Markov model (AR-HMM):

$$p(\mathbf{X}_{1:T}^i, Z_{1:T}^i) = p(Z_1^i) p(\mathbf{X}_1^i \mid Z_1^i) \prod_{t=2}^T p(Z_t^i \mid Z_{t-1}^i) p(\mathbf{X}_t^i \mid \mathbf{X}_{t-1}^i, Z_t^i), \quad (3.8)$$

where $p(Z_1^i)$ is the initial distribution over actor i 's initial community assignment, $p(Z_t^i \mid Z_{t-1}^i)$ is the transition distribution between actor i 's label assignment at time $t-1$ and time t , and $p(\mathbf{X}_t^i \mid \mathbf{X}_{t-1}^i, Z_t^i)$ is the emission distribution for actor i 's latent position at time t , which we allow to depend on the previously emitted value $\mathbf{X}_{t-1}^i$.

To properly model evolving communities, we make an important departure from previous dynamic latent space models. In particular, we expand the probabilistic model to include time-inhomogeneous Markov chains where the transition matrix $p(Z_t^i | Z_{t-1}^i)$ can vary over time. This structure contrasts with previous methods, which assume that an actor’s probability of transitioning from community i to community j is the same at all times. We argue that time-inhomogeneous transitions are essential characteristics of evolving communities. For example, the addition of a group requires the transition matrices to add a non-zero probability of transitioning into that new group. Furthermore, once the new group appears in the network, an actor’s probability of transitioning into that group may approach a different steady-state from when it initially appeared. Although models that utilize homogeneous Markov chains may infer changing group structure by including empty clusters, their inferences will undoubtedly be biased due to smoothing over sudden group structure changes.

Tractable inference over these time-inhomogeneous latent trajectories is made possible by tying together the transition distributions and emission distributions of every actor. We collect the transition probabilities into a single row-stochastic transition matrix Π_t for each time step $t = 2, \dots, T$. Each entry is defined as $(\Pi_t)_{jk} = \pi_{jk}^t = p(Z_t^i = k | Z_{t-1}^i = j)$, which is the same for all n actors. In what follows, it will be useful to isolate the j -th row of Π_t in the vector π_j^t . Furthermore, we refer to $\{\pi_j^t\}$ as the set of rows of the transition matrix at time t . For simplicity, we assume that all actors share a common initial state $Z_0^i = 0$. This allows us to view the initial distribution of the Markov chain as a special transition distribution denoted by $\pi_0^1 = (\pi_{01}^1, \dots, \pi_{0G_0}^1)$, where $\pi_{0k}^1 = p(Z_1^i = k | Z_0^i = 0) = p(Z_1^i = k)$ is equal for all actors.

Conditioned on the initial and transition probability matrices and the parameters of the emission distributions θ , the data generating process of the actors’ trajectories is characterized by a collection of iid AR-HMMs. In this work, we model the joint density over all actors’ trajectories as

$$\begin{aligned} p(\mathcal{X}_{1:T}, \mathcal{Z}_{1:T} | \Pi_{2:T}, \pi_0^1, \theta) &= \prod_{i=1}^n p(\mathbf{X}_{1:T}^i, Z_{1:T}^i | \Pi_{2:T}, \pi_0^1, \theta) \\ &= \prod_{i=1}^n \pi_{0, Z_1^i}^1 N(\mathbf{X}_1^i | \mu_{Z_1^i}, \sigma_{Z_1^i}^2 I_p) \times \prod_{t=2}^T \pi_{Z_{t-1}^i, Z_t^i}^t N(\mathbf{X}_t^i | \lambda \mu_{Z_t^i} + (1 - \lambda) \mathbf{X}_{t-1}^i, \sigma_{Z_t^i}^2 I_p), \end{aligned} \quad (3.9)$$

where I_p is the $p \times p$ identity matrix, and $N(\mathbf{X} | \mu, \Sigma)$ is the normal density with mean vector μ and covariance Σ evaluated at the point $\mathbf{X}$. Note that in this case $\theta = \{\{\mu_g, \sigma_g^2\}_{g=1}^G, \lambda\}$. Like the clustering model in Handcock et al. (2007), the communities are modeled as a multivariate normal distribution in the latent space with mean location μ_g , and spherical covariance $\sigma_g^2 I_p$. As in the longitudinal clustering approach of Sewell et al. (2016), the mean position $\mathbf{X}_t^i$ is equal to $\lambda \mu_g + (1 - \lambda) \mathbf{X}_{t-1}^i$ where $\lambda \in (0, 1)$. This is a blend between the actor’s previous position and the current assigned group location. Consequently, λ is called the blending coefficient. Note that we exclude $\lambda = 0$ so that latent positions remain clustered around the group centers in latent space. This behavior is necessary to link the latent space’s evolving cluster structure to the observed network’s evolving community structure. Furthermore, we exclude $\lambda = 1$ so that the model accounts for the usually strong autocorrelation in the latent positions, which improves forecasting performance.

In addition, we kept $\lambda \in (0, 1)$ to allow the model to capture slowly merging (or splitting) communities, which is not possible when $\lambda = 0$ or 1 . Specifically, $\lambda \in (0, 1)$ induces a biased random walk that encourages actors to travel along the line connecting their current latent position to their assigned group center μ_g . For example, consider two communities that slowly merge into one, e.g., a dynamic friendship network

between students from two middle schools that slowly exhibits more across school friendships as students begin to attend a common high school. Such dynamics are described by a latent space initially populated by two communities, corresponding to the two middle schools, and the emergence of a third community, corresponding to the high school, at their midpoint. As students are assigned to the high school community, the high school's group center acts as a sink where actors from the two middle school clusters slowly wander over to the third community's center. The size of λ controls the speed of the merge with smaller values resulting in slower merging behavior. Although the main manuscript primarily analyzes abrupt changes in community structure, where λ is near one, we include a slow merging event in Section 3.8.8.

To complete the model, we must specify a prior on the initial distribution π_0^1 , the transition probabilities $\Pi_{2:T}$, and the parameters of the emission distribution $\theta = \{\{\mu_g, \sigma_g^2\}_{g=1}^G, \lambda\}$. Recall that we account for evolving community structure by allowing the transition probabilities' dimensionality to change over time. Furthermore, we refrain from specifying G , the total number of groups, or G_t , the active groups at time t , which means that the prior must quantify their uncertainty. To satisfy these requirements, we turn to the flexibility afforded by a Bayesian nonparametric approach.

3.3.3 Background: Dirichlet Processes and the HDP-HMM

The Dirichlet process (DP) is a distribution over discrete probability measures F_0 :

$$\beta \sim \text{GEM}(\gamma), \quad \theta_k \stackrel{\text{iid}}{\sim} H, \quad F_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}, \quad (3.10)$$

where $\beta \sim \text{GEM}(\gamma)$ denotes the following sticking-breaking process (Sethuraman, 1994):

$$\beta_k = w_k \prod_{i=1}^{k-1} (1 - w_i), \quad w_k \sim \text{Beta}(1, \gamma), \quad k = 1, 2, \dots, \quad (3.11)$$

and δ_{θ_k} is a point mass at θ_k . In effect, $\beta \sim \text{GEM}(\gamma)$ divides a unit-length stick into pieces with lengths given by the weights β_k where the k -th weight is a random proportion w_k of the remaining stick length $\prod_{i=1}^{k-1} (1 - w_i)$. We use the notation $F_0 \sim \text{DP}(\gamma, H)$ to indicate draws from a DP with concentration parameter γ and base measure H . Since draws from a DP are discrete with probability one, the DP cannot be used as a general nonparametric prior over continuous densities. To extend the DP to continuous density estimation, one uses F_0 as a mixing measure over some parametric class of distributions f_{θ} . This construction is known as the DP mixture model. The sampling process begins by drawing indicator variables $z_i \stackrel{\text{iid}}{\sim} \beta$, and generating observations as $y_i \sim f_{\theta_{z_i}}$. Note that we adopt the convention that if a K -dimensional vector $\mathbf{v}$ belongs to the K simplex, i.e., $\sum_{j=1}^K v_j = 1$ and each entry $v_j \geq 0$, then $z \sim \mathbf{v}$ means $z \sim \sum_{j=1}^K v_j \delta(z, j)$, where $\delta(i, j)$ is the Kronecker delta. Furthermore, the same convention applies when K is infinite, which is the case for β .

The hierarchical Dirichlet process (HDP) is a distribution over a collection of discrete probability measures, $\{F_j : j = 1, \dots, J\}$, that share a common set of atoms:

$$\begin{aligned} \beta &\sim \text{GEM}(\gamma), \\ \pi_j &\stackrel{\text{iid}}{\sim} \text{DP}(\alpha, \beta), \quad \theta_k \stackrel{\text{iid}}{\sim} H, \quad F_j = \sum_{k=1}^{\infty} \pi_{jk} \delta_{\theta_k}, \quad j = 1, \dots, J. \end{aligned} \quad (3.12)$$

The group-specific distributions of the HDP, π_j , are often used as priors over the rows of an infinite-dimensional transition matrix, i.e., $(\Pi)_{jk} = \pi_{jk}$ for $j \in \mathbb{N}$ so that number of states J is infinite. This formulation allows for the construction of the hierarchical Dirichlet process hidden Markov model (HDP-HMM) (Teh et al., 2006), which is a natural Bayesian nonparametric extension of the Bayesian HMM (Robert et al., 2000). The sampling mechanism of the HDP-HMM proceeds as follows: one samples the hidden states sequentially as $z_t \sim \pi_{z_{t-1}}$, and the observations are linked to the global parameters via $y_t \sim f_{\theta_{z_t}}$. Note that the sharing of atoms induced by the HDP prior allows the Markov chain to utilize a single global set of parameters, θ_k , at all time points.

Despite its popularity the original HDP-HMM struggles to model Markov chains with long state durations. To remedy this issue E. B. Fox et al. (2011) introduced the sticky hierarchical Dirichlet process (sticky HDP):

$$\begin{aligned} \beta &\sim \text{GEM}(\gamma), \\ \pi_j &\stackrel{\text{iid}}{\sim} \text{DP}\left(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_j}{\alpha + \kappa}\right), \quad \theta_k \stackrel{\text{iid}}{\sim} H, \quad F_j = \sum_{k=1}^{\infty} \pi_{jk} \delta_{\theta_k}, \quad j = 1, \dots, J, \end{aligned} \quad (3.13)$$

where δ_j is a vector of zeros except for a single one at the j -th index. Analogous to the HDP, the sticky HDP can be used as a prior over the transition matrices of an HMM. In this case, the extra stickiness parameter $\kappa > 0$ biases the process towards self-transitions. As a result, E. B. Fox et al. (2011) found that the corresponding sticky HDP-HMM better models the longer state durations found in real-world applications. For this reason, we use the sticky HDP as a prior over the transition matrices in Equation (3.9).

3.3.4 An HDP Prior for Time-Inhomogeneous Markov Chains

Now, we present our extension to the latent position cluster model that can infer an evolving community structure in dynamic networks. To accomplish this goal, we place the following Bayesian nonparametric prior on the transition matrices in Equation (3.9):

$$\begin{aligned} \beta &\sim \text{GEM}(\gamma), \\ \pi_0^1 &\stackrel{\text{iid}}{\sim} \text{DP}(\alpha_0, \beta), \\ \pi_g^t &\stackrel{\text{iid}}{\sim} \text{DP}\left(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_g}{\alpha + \kappa}\right), \quad (\mu_g, \sigma_g^2) \stackrel{\text{iid}}{\sim} H_{\mu} \times H_{\sigma^2}, \quad g = 1, 2, \dots, \quad t = 2, \dots, T. \end{aligned} \quad (3.14)$$

In this work, we take H_{μ} to be a multivariate normal distribution and H_{σ^2} to be an inverse gamma distribution, i.e., $\mu_g \stackrel{\text{iid}}{\sim} N(\mu_0, \tau^2 I_p)$ and $\sigma_g^2 \stackrel{\text{iid}}{\sim} \Gamma^{-1}(a/2, b/2)$. For the initial distribution, we set $\kappa = 0$ and allocate it a separate concentration parameter α_0 to distinguish it from the other transition distributions. Note that, unlike the traditional HDP-HMM, our model re-samples the rows of the transition matrix, π_g^t , at each time step. This difference allows us to infer time-inhomogeneous Markov chains. As we previously argued, this time-inhomogeneity is crucial for modeling evolving communities. In short, our model extends the dynamic Euclidean distance model proposed in Sewell and Chen (2017) by incorporating time-inhomogeneous transition matrices that are inferred through a nonparametric HDP prior. For this reason, we refer to our model as the hierarchical Dirichlet process latent position cluster model (HDP-LPCM). A graphical representation of the HDP-LPCM is depicted in Figure 3.2.

Although our prior borrows heavily from the sticky HDP-HMM formulation, there are three key differences between the state-space model utilized in the HDP-LPCM and the traditional sticky HDP-HMM. First,

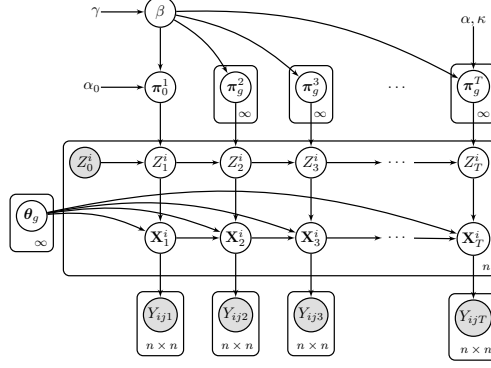


Figure 3.2: The graphical model for the HDP-LPCM. For clarity, we only display the dependence on α and κ for the last transition matrix, although it is present in all transition matrices from $t = 2, \dots, T$.

the emitted values $\mathbf{X}_t^i$ have an autoregressive dependence. This modification was originally explored outside the network setting by E. Fox et al. (2011) to model switching linear dynamical systems with Gaussian observations. However, in our case, the observations are a sequence of binary adjacency matrices. The second difference is that the HDP-HMM contains a single state sequence $Z_{1:T}$, while we infer multiple state sequences $Z_{1:T}^i$ for $i = 1, \dots, n$ from a single observed sequence of networks. The final departure from the HDP-HMM is the re-sampling of the rows of the transition matrix $\{\pi_g^t\}$ at every time step. This difference introduces time-inhomogeneity into the Markov chain, which is crucial for modeling the additions, deletions, splits, and mergers of groups found in evolving communities.

In addition to properly modeling an evolving community structure, the other important accomplishment of our model is its ability to infer the number of communities from the data in a fully Bayesian way. A typical parametric approach (specifying G_t at each time step) would require comparing a combinatorial amount of models. This task is computationally infeasible for even a small number of groups and time steps. Through a nonparametric prior, we naturally incorporate model selection into our inference procedure, which avoids the computationally expensive model selection step found in many LPCMs (Handcock et al., 2007; Sewell & Chen, 2017).

3.3.5 The Weak-Limit Approximation

For inference, we utilize the weak-limit approximation to the HDP proposed in E. B. Fox et al. (2011). The approximation replaces the infinite dimensional DPs with finite L -dimensional Dirichlet distributions as follows:

$$\begin{aligned} \beta &\sim \text{Dirichlet}(\gamma/L, \dots, \gamma/L), \\ \pi_0^1 &\stackrel{\text{iid}}{\sim} \text{Dirichlet}(\alpha_0\beta_1, \dots, \alpha_0\beta_L), \quad \pi_g^t \stackrel{\text{iid}}{\sim} \text{Dirichlet}(\alpha\beta_1, \dots, \alpha\beta_g + \kappa, \dots, \alpha\beta_L), \\ \mu_g &\stackrel{\text{iid}}{\sim} N(\mathbf{0}, \tau^2 I_p), \quad \sigma_g^2 \stackrel{\text{iid}}{\sim} \Gamma^{-1}(a/2, b/2), \quad g = 1, \dots, L, \quad t = 2, \dots, T. \end{aligned} \quad (3.15)$$

Practically, the weak limit approximation transforms the infinite transition matrices into finite $L \times L$ matrices. The parameter L gives us control over the approximation with the accuracy increasing as L goes to infinity (Ishwaran & Zarepour, 2000). In practice, one initially sets L larger than some a priori upper limit on the number of communities and verifies this choice by checking that the posterior puts negligible mass on models with L groups.

3.4 Estimation

We take a Bayesian approach to estimation. In what follows, we describe a Markov chain Monte Carlo (MCMC) method to sample from the HDP-LPCM’s posterior. We implement a Metropolis-Hastings within Gibbs MCMC scheme with the goal of identifying an evolving community structure consistent with the inferred posterior distribution.

3.4.1 Blocked Metropolis-Hastings within Gibbs Sampler

As outlined in Section 3.3, the joint distribution over all the variables (Equation (3.1)) factors as

$$\underbrace{p(Y_{1:T} \mid \mathcal{X}_{1:T}, \beta_0)}_{\text{Equation (3.3)}} \cdot \overbrace{p(\mathcal{X}_{1:T}, \mathcal{Z}_{1:T} \mid \Pi_{2:T}, \boldsymbol{\pi}_0^1, \boldsymbol{\mu}_{1:L}, \sigma_{1:L}^2, \lambda)}^{\text{Equation (3.9)}} \cdot \underbrace{p(\Pi_{2:T}, \boldsymbol{\pi}_0^1 \mid \boldsymbol{\beta}) \cdot p(\boldsymbol{\mu}_{1:L}, \sigma_{1:L}^2) \cdot p(\boldsymbol{\beta})}_{\text{Equation (3.15)}}. \quad (3.16)$$

To complete the model we assign the following priors:

$$\beta_0 \sim N(\mu_{\beta_0}, \sigma_{\beta_0}^2), \quad \lambda \sim N_{(0,1)}(\mu_\lambda, \sigma_\lambda^2), \quad (3.17)$$

where $N_{(0,1)}(\mu, \sigma^2)$ indicates the normal distribution with mean μ and variance σ^2 truncated to the range $(0, 1)$.

We realize these samples by following a Gibbs sampling algorithm in which we iteratively sample from the appropriate conditional distributions of $\mathcal{X}_{1:T}$, $\mathcal{Z}_{1:T}$, $\boldsymbol{\pi}_0^1$, $\Pi_{2:T}$, $\boldsymbol{\beta}$, $\boldsymbol{\mu}_{1:L}$, $\sigma_{1:L}^2$, λ , and β_0 . Due to our choice of priors, most conditional distributions are conjugate so that the Gibbs updates are derived in standard fashion. See Section 3.8.1 for the details on these conditional distributions. Metropolis-Hastings (MH) steps are necessary for the latent positions and the intercept. In both cases, we use a normal random walk proposal. In all experiments, we tune the proposal step sizes using an initial tuning phase so that the proposed moves have a 25% - 40% acceptance rate. The only samplers that need special care are the block sampler for the actor labels $\mathcal{Z}_{1:T}$, the sampler for the global prior $\boldsymbol{\beta}$, as well as the additional hyperparameter sampling schemes. We fully derive these samplers in the supplementary material. In addition, we provide details of our parameter initialization scheme in Section 3.8.7.

The full Metropolis-Hastings within Gibbs sampler is outlined in Algorithm 3.1. When describing the samplers, we use dot notation to indicate summations over an index, e.g., for a random variable w_{ab} , $w_{.b} = \sum_a w_{ab}$, $w_{a.} = \sum_b w_{ab}$ and $w_{..} = \sum_a \sum_b w_{ab}$. In addition, we use n_{kjt} to denote the number of actors that transitioned from group k to group j at time t and n_{kt} to denote the number of actors in group k at time t .

3.4.2 Posterior Summarization

The selection of a single point estimate that adequately summarizes the full posterior is a challenging and open problem in Bayesian nonparametric statistics. A simple solution uses the posterior mode. However, in Bayesian nonparametric models, the MCMC chain often only visits a single partition once, which makes frequency estimates unreliable. Another approach is to choose the partition corresponding to the MAP (maximum a posteriori) estimate. A downside of the MAP approach is that it does not marginalize over the uncertainty in the partitions. As a result, MAP clustering estimates tend to over-fit the data. An ideal summarization methodology should take into account the clustering uncertainty suggested by the posterior samples.

Given the previous initial distribution $(\pi_0^1)^{(\ell-1)}$, a set of state-specific transition probabilities $\Pi_{2:T}^{(\ell-1)}$, the global transition distribution $\beta^{(\ell-1)}$, group parameters $(\mu_{1:L}, \sigma_{1:L}^2)^{(\ell-1)}$, node labels $\mathcal{Z}_{1:T}^{(\ell-1)}$, latent positions $\mathcal{X}_{1:T}^{(\ell-1)}$, the likelihood specific parameters $\beta_0^{(\ell-1)}$, and blending coefficient $\lambda^{(\ell-1)}$, update the current parameters as follows:

1. Initialize current parameters to the values of the $(\ell - 1)$ th iteration.
2. Update latent positions $\mathcal{X}_{1:T}$ via MH with a normal random walk proposal.
3. Update β_0 via MH with a normal random walk proposal.
4. Update node labels $\mathcal{Z}_{1:T}$ using the forward-backward algorithm detailed in Algorithm 3.2 in Section 3.8.2.
5. Sample the auxiliary variables $\mathbf{m}, \bar{\mathbf{m}}, \mathbf{w}$ as in Algorithm 3.3 in Section 3.8.4.
6. Update the global transition distribution by sampling

$$\beta \sim \text{Dirichlet}(\gamma/L + \bar{m}_{\cdot 1}, \dots, \gamma/L + \bar{m}_{\cdot L}).$$

7. For each $k \in \{1, \dots, L\}$, sample a new initial distribution based on the initial assignments:

$$\pi_0^1 \sim \text{Dirichlet}(\alpha_0 \beta_1 + n_{011}, \dots, \alpha_0 \beta_L + n_{0L1}).$$

8. For each $k \in \{1, \dots, L\}$ and $t \in \{2, \dots, T\}$, sample a new transition distribution based on the sample assignments:

$$\pi_k^t \sim \text{Dirichlet}(\alpha \beta_1 + n_{k1t}, \dots, \alpha \beta_k + \kappa + n_{kkt}, \dots, \alpha \beta_L + n_{kLt}).$$

9. Update cluster parameters. For each $k \in \{1, \dots, L\}$:

$$\begin{aligned} \mu_k &\sim N(\bar{\mu}_k, \bar{\sigma}_k^2 I_p), \\ \sigma_k^2 &\sim \Gamma^{-1}((n_k \cdot p + a)/2, \bar{b}/2), \end{aligned}$$

where $\bar{\mu}_k, \bar{\sigma}_k^2$ are defined in Equation (3.20) and $\bar{b}$ is defined in Equation (3.21) in Section 3.8.1.

10. Update blending coefficient λ :

$$\lambda \sim N_{(0,1)}(\bar{\mu}_\lambda, \bar{\sigma}_\lambda^2),$$

where $\bar{\mu}_\lambda, \bar{\sigma}_\lambda^2$ are defined in Equation (3.22) in Section 3.8.1.

11. Update remaining hyperparameters $\{\tau^2, b, \gamma, \alpha_0, \alpha, \kappa, \rho\}$ as in Algorithm 3.5 in Section 3.8.6.

Algorithm 3.1: Blocked Metropolis-Hastings within Gibbs sampler for the HDP-LPCM.

In this work, we take a decision theoretic approach to posterior summarization. Specifically, we select the partition that minimizes the posterior expectation of an appropriately chosen loss function over dynamic clusterings. In the case of static clustering, two popular choices of loss functions are Binder’s loss (Binder, 1978) and the variation of information (Meilă, 2007). Their use for posterior summarization was advocated by Lau and Green (2012) and Wade and Ghahramani (2018), respectively. Note that minimizing Binder’s loss is equivalent to maximizing the Rand index, another popular measure of clustering performance. Both loss functions have the desirable property that they are metrics over the space of clusterings. Furthermore, the optimization of the expected losses only depends on the posterior co-occurrence probabilities, so avoids complications due to label switching.

To extend this approach to dynamic clusterings, we minimize the posterior expected time-averaged variation of information (VI) which averages the static VI’s at each time step as follows:

$$\arg \min_{\hat{\mathcal{Z}}_{1:T}} \mathbb{E} \left[\frac{1}{T} \sum_{t=1}^T \text{VI}(\mathcal{Z}_t, \hat{\mathcal{Z}}_t) \mid Y_{1:T} \right],$$

where $\text{VI}(\mathcal{Z}_t, \hat{\mathcal{Z}}_t) = 2H(\mathcal{Z}_t, \hat{\mathcal{Z}}_t) - H(\mathcal{Z}_t) - H(\hat{\mathcal{Z}}_t)$ and $H(\mathcal{Z}_t, \hat{\mathcal{Z}}_t)$, $H(\mathcal{Z}_t)$, and $H(\hat{\mathcal{Z}}_t)$ are entropies calculated using the joint and marginal distributions of the labels. Similar to Wade and Ghahramani (2018), we instead minimize the following calculable lower-bound:

$$\arg \min_{\hat{\mathcal{Z}}_{1:T}} \left\{ \sum_{t=1}^T \sum_{i=1}^n \log \left(\sum_{j=1}^n \mathbb{1}_{\{\hat{\mathcal{Z}}_t^j = \hat{\mathcal{Z}}_t^i\}} \right) - 2 \sum_{t=1}^T \sum_{i=1}^n \log \left(\sum_{j=1}^n p(Z_t^j = Z_t^i \mid Y_{1:T}) \mathbb{1}_{\{\hat{\mathcal{Z}}_t^j = \hat{\mathcal{Z}}_t^i\}} \right) \right\}.$$

Notice that this expression only depends on the posterior co-occurrence probabilities, which can be estimated from the posterior samples. We perform this optimization by searching over all posterior samples from the MCMC chain (after an appropriate burn-in). Possible ties are broken by selecting the sample with the highest likelihood (Equation (3.3)). This minimization does not require the partitions to be visited by the Markov chain; however, by restricting the search to the sampled partitions, we have access to the associated latent space for later visualization of the network.

3.4.3 Non-Identifiability of the Latent Positions

Since latent position models depend on the distance between actors, it is clear that they are invariant to translations, rotations, and reflections of the latent space. Any posterior inference that utilizes these positions must correct for such a non-identifiability. We take the approach of Procrustes matching commonly employed in the literature. This involves post-processing the samples by rotating and translating them to match a reference layout. For a detailed description of this procedure, see the original work by P. D. Hoff et al. (2002). We use the sample chosen by the procedure in Section 3.4.2 as the reference layout.

3.5 Simulation Study

We designed a simulation study to assess the HDP-LPCM’s performance on synthetic networks with known community dynamics that mimic those found in real-world networks. We considered two scenarios: a sequence of networks with (1) a single static number of groups active over all time steps and (2) a number of groups that changes over time because the groups have either merged or split. We labeled these scenar-

ios the time-homogeneous and time-inhomogeneous simulations, respectively. Additionally, we provide a simulation study that investigated how group separation and stability affect the HDP-LPCM’s performance in Section 3.8.9 and a sensitivity analysis that showed that the estimation method is unaffected by our hyperparameter choices Section 3.8.10.

3.5.1 Measuring Performance

For each simulation, we evaluated the following three tasks: (1) assigning nodes to the correct community at each time step while penalizing label switching across time steps, (2) selecting the correct number of communities at each time step, and (3) predicting in-sample, missing, and one-step ahead dyads. We emphasize that the primary goals of the HDP-LPCM are the first two tasks which involve clustering and community number selection.

To evaluate the clustering performance outlined in task (1), we used two metrics: the adjusted Rand index (ARI) and the variation of information (VI) introduced in Section 3.4.2. The ARI takes values in $[0, 1]$ and measures the amount of agreement between two cluster assignments. Values close to one indicate near-identical assignments, while values near zero are roughly expected when making assignments at random. On the other hand, the VI is a true metric that takes values in $[0, 2\log_2(K^*)]$, where K^* is the maximum number of communities, e.g., $K^* = L$ for the HDP-LPCM. Values near zero indicate that the two cluster assignments are near identical. Unlike the ARI, the VI satisfies the triangle inequality and is symmetric in its arguments. We included the VI because we used it for posterior summarization. In the dynamic setting, we can compute at least two versions of these metrics. The first version computes the metric at each time step and averages the results. We refer to these as the average ARI and average VI. The second version considers the labels for all time steps together and computes a single metric called the global ARI and global VI. Achieving a good value of the global metrics is more challenging since it requires consistent labeling within each time step and across time steps. As such, we are primarily concerned with good global performance.

We measured predictive performance in three ways. First, to determine the goodness-of-fit of the models to the training data, we calculated the area under the receiver operating characteristic curve (AUC) based on the model’s predictions of in-sample dyads. Next, we measured out-of-sample performance by evaluating the model’s ability to impute missing dyads and make one-step ahead predictions. To do this, in each simulation, we held out the last network and removed 10% of the dyads from each proceeding network. Then, we obtained edge predictions for the missing dyads and one-step ahead predictive probabilities and calculated the AUC values based on the dyads involved in each task.

3.5.2 Competing Methods

We compared our proposed HDP-LPCM to three competing methods: two stochastic block models (SBMs) and a parametric LPCM. The first method used a single SBM fit at each time step using the variational EM algorithm with the number of communities selected by maximizing the integrated classification likelihood (ICL) criterion. To apply this method to dynamic networks, we needed to account for label switching between time steps. To do so, we combined the output sequentially using the label switching algorithm proposed in Papastamoulis and Iliopoulos (2010). The second method is the dynamic SBM introduced in Matias and Miele (2017) with the number of clusters chosen by maximizing the ICL criterion. This model assumes a fixed number of communities, so we only applied it to the homogeneous simulation. We used the `blockmodels` and `dynsbm` R packages to estimate the SBM and dynamic SBM, respectively. However, since these packages

do not provide one-step ahead predictions or perform missing edge imputation, we did not remove any dyads from the simulated networks and only calculated the AUC based on predicting in-sample dyads for the block models.

The last competing method is the parametric LPCM proposed in Sewell and Chen (2017), which only applies to the homogeneous simulations. We replaced the directed likelihood with the undirected likelihood of this work and set the number of communities to the ground truth. Lastly, we extracted the following two estimators of the community assignments: LPCM (MAP) and LPCM (VI). As their names imply, LPCM (MAP) is the original MAP estimator proposed in Sewell and Chen (2017), and LPCM (VI) used the VI summarization procedure we introduced in Section 3.4.2. All LPCMs imputed missing edges and calculated one-step ahead predictions using the augmented Gibbs sampler and Monte Carlo marginalization scheme described in Sewell and Chen (2015b). Note that the HDP-LPCM’s forecasts assume that $\Pi_{T+1} = \Pi_T$ for simplicity.

The HDP-LPCM and the LPCMs were estimated using a Markov chain consisting of 50,000 samples. We used the initial 5,000 samples to tune the step sizes of the Metropolis-Hastings samplers. We discarded the following 10,000 samples as burn-in leaving 35,000 samples for inference. For the HDP-LPCM, we fixed the truncation level of the weak-limit approximation (Equation (3.15)) to $L = 10$ in all experiments.

3.5.3 Time-Homogeneous Simulation

This simulation contains a single set of groups and transition matrices common to all time points. Our goal is to demonstrate that the HDP-LPCM performs comparably to the parametric LPCM (Sewell & Chen, 2017) while also providing a proper assessment of uncertainty in the network’s community structure. The networks were generated according to the sampling mechanisms described in Sections 3.3.1 and 3.3.2. We fixed the total number of groups to $G = 6$ at each time point. We chose the blending coefficient $\lambda = 0.8$, the dimension of the latent space $p = 2$, and the intercept $\beta_0 = 1.0$. We set the group locations to

$$(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_6) = \begin{pmatrix} -4 & 4 & -2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -5 & 5 \end{pmatrix}. \quad (3.18)$$

We drew the group shapes σ_g^2 from a $\Gamma^{-1}(6, 2)$ distribution. These parameters induce substantial overlap between groups 1 and 3 and groups 2 and 4 in the latent space. See the second column of Figure 3.1 for an adjacency matrix drawn from this group structure. Next, the initial distribution $\boldsymbol{\pi}_0^1$ was drawn from a six-dimensional Dirichlet(10, ..., 10). The six rows of the transition matrix, $\boldsymbol{\pi}_g^t$ for $g = 1, \dots, 6$ and $t = 2, \dots, 6$, were chosen proportional to

$$\left(\frac{1}{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_g\|}, \dots, \frac{1}{\|\boldsymbol{\mu}_{g-1} - \boldsymbol{\mu}_g\|}, \max_{k \neq g} \left\{ \frac{\text{const}}{\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_g\|} \right\}, \frac{1}{\|\boldsymbol{\mu}_{g+1} - \boldsymbol{\mu}_g\|}, \dots, \frac{1}{\|\boldsymbol{\mu}_G - \boldsymbol{\mu}_g\|} \right).$$

We set the constant in the above equation equal to 20, which yields self-transition probabilities ranging from 0.83 to 0.9. The latent actor positions, $\mathcal{X}_{1:T}$, and group assignments, $\mathcal{Z}_{1:T}$, were drawn from Equation (3.9). Finally, we generated the adjacency matrices according to Equation (3.2) with the linear predictor η_{ijt} given by Equation (3.5). We simulated 50 dynamic networks with $T = 7$ time points and $n = 120$ actors each. Note that we only used the last network to evaluate one-step ahead predictions.

Table 3.1 contains the AUC values from this simulation. As a result of the prevalent community structure in the simulated networks, all the models have comparable in-sample predictive performances. Also, the HDP-

Simulation	Model	AUC (in-sample)	AUC (missing)	AUC (one-step ahead)
Homogeneous	HDP-LPCM	0.911 (0.008)	0.897 (0.011)	0.810 (0.018)
Homogeneous	LPCM (VI)	0.911 (0.008)	0.896 (0.011)	0.809 (0.018)
Homogeneous	LPCM (MAP)	0.911 (0.008)	0.896 (0.010)	0.809 (0.018)
Homogeneous	Dynamic SBM	0.889 (0.010)	—	—
Homogeneous	SBM	0.892 (0.012)	—	—
Homogeneous	Naive	—	—	0.617 (0.013)
Inhomogeneous	HDP-LPCM	0.871 (0.004)	0.856 (0.007)	0.782 (0.021)
Inhomogeneous	SBM	0.849 (0.006)	—	—
Inhomogeneous	Naive	—	—	0.603 (0.014)

Table 3.1: Median AUC values for all 50 time-homogeneous / time-inhomogeneous simulations. Standard deviations are displayed in parentheses. The Naive model corresponds to using the dyads in Y_T to forecast Y_{T+1} .

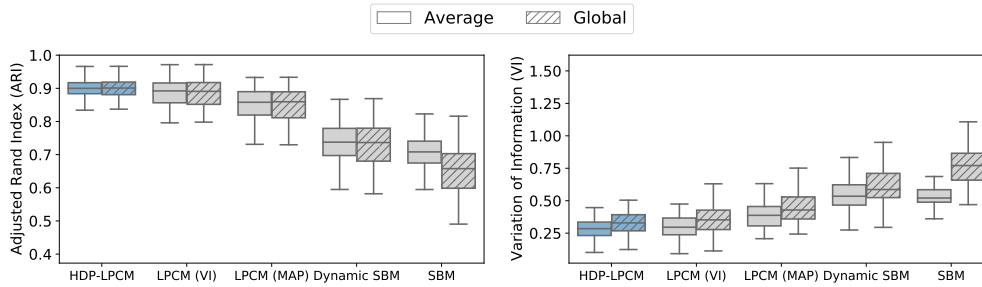


Figure 3.3: Boxplots of ARI (left) and VI (right) of the five competing methods on the time-homogeneous simulation. The non-hatched and hatched boxes denote the average and global metrics, respectively. The proposed method is highlighted in blue.

LPCM’s predictions of missing and one-step ahead dyads are on par with those of the parametric LPCMs that directly model the true data-generating process. However, the ARI and VI values in Figure 3.3 reveal that the block models are doing a poor job capturing the underlying community assignments. Furthermore, the static SBMs have a significantly worse global ARI and VI due to their inability to properly match the community labels across time steps. Lastly, the LPCMs that used the VI estimator substantially outperformed the LPCMs that used the MAP estimator, which validates our method of posterior summarization.

Next, we evaluated the ability of the HDP-LPCM to properly assign uncertainty to the dynamic network’s community structure. Figure 3.4 contains boxplots of the estimated posterior probabilities of $|G_t|$ over the 50 simulations. In most cases, $|G_t| = 6$ is correctly identified with the highest posterior probability. Next, we compared the ability of our posterior summarization method to select a partition with the correct number of communities. Figure 3.5 displays the number of communities selected by the HDP-LPCM at each time step as well as by the SBMs which are included as a baseline. The HDP-LPCM drastically outperforms the block models, especially the static SBMs which consistently underestimate the number of groups. Overall, the HDP-LPCM has equivalent predictive and clustering performance to the parametric LPCM while also accurately estimating the true number of communities which justifies its use as an efficient drop-in replacement for the parametric LPCM.

3.5.4 Time-Inhomogeneous Simulation

We designed this simulation to test our model’s ability to detect changes in group structure by allowing the number of groups to vary over time. We generated 50 networks with $T = 10$ time steps and $n = 120$

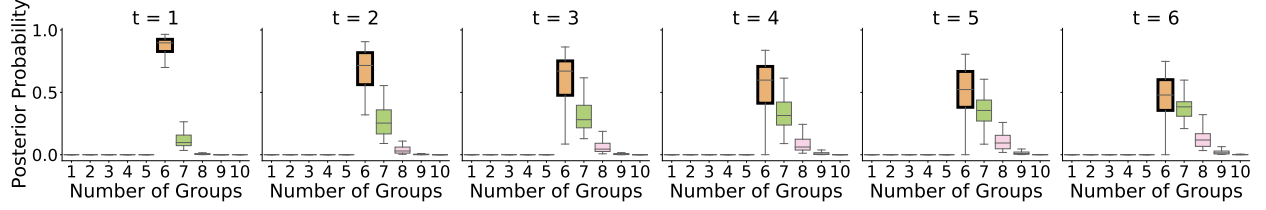


Figure 3.4: Boxplots showing the posterior distribution of $|G_t|$ over the 50 time-homogeneous simulations with bolded boxes indicating the true number of groups.

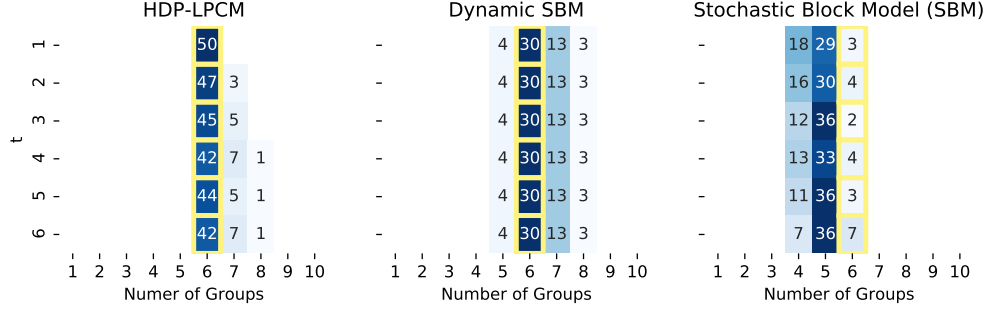


Figure 3.5: The number of groups estimated at each time step by the HDP-LPCM (left), dynamic SBM (center), and the static SBMs (right). The yellow cells contain the true number of communities, $|G_t| = 6$ at all time steps.

actors each. There are $G = 6$ groups in the networks overall. The simulations begin with two groups, these two groups split into six groups at $t = 4$, and then the six groups merge into four groups at $t = 7$. The introductory example (Figure 3.1) displays adjacency matrices generated from this procedure.

Once again, the networks were simulated according to the sampling mechanisms described in Sections 3.3.1 and 3.3.2 with the last network held out to evaluate one-step ahead forecasts and 10% of the dyads removed at each time step to assess dyad imputation. This simulation differs from the time-homogeneous simulation in how we specified the transition matrices π_g^t . We set the initial distribution $\pi_0^1 = (1/2, 1/2)^T$. We chose the rows of the transition matrix, π_g^t for $g \in G_{t-1}$ and $t = 2, \dots, 10$, proportional to

$$\left(\frac{\mathbb{1}_{\{1 \in G_t\}}}{\|\mu_1 - \mu_g\|}, \dots, \frac{\mathbb{1}_{\{g-1 \in G_t\}}}{\|\mu_{g-1} - \mu_g\|}, \right. \\ \left. \max_{k \neq g: k \in G_t} \left\{ \frac{\text{const}}{\|\mu_k - \mu_g\|} \right\} \mathbb{1}_{\{g \in G_t\}}, \frac{\mathbb{1}_{\{g+1 \in G_t\}}}{\|\mu_{g+1} - \mu_g\|}, \dots, \frac{\mathbb{1}_{\{G \in G_t\}}}{\|\mu_G - \mu_g\|} \right), \quad (3.19)$$

where G_t indicates the set of active groups at time t . At $t = 1, 2, 3$ there are two groups with $G_1 = G_2 = G_3 = \{1, 2\}$. From $t = 4, 5, 6$ these two groups split into six groups so that $G_4 = G_5 = G_6 = \{1, 2, 3, 4, 5, 6\}$. Finally, these six groups merge into four groups at $t = 7, 8, 9$ so that $G_7 = G_8 = G_9 = \{1, 2, 3, 4\}$. For all time points, we set the constant in Equation (3.19) to 20 except for $t = 4$. At $t = 4$, we set the constant equal to 1 so that the nodes were more evenly distributed among the six newly created groups. In addition, we set $\lambda = 0.9$ which resulted in stronger group cohesion and more closely matched the values found in real-world networks.

The competing methods' predictive performance on this simulation are displayed in the second half of Table 3.1. The in-sample performance is comparable between the HDP-LPCM and the SBM due to the

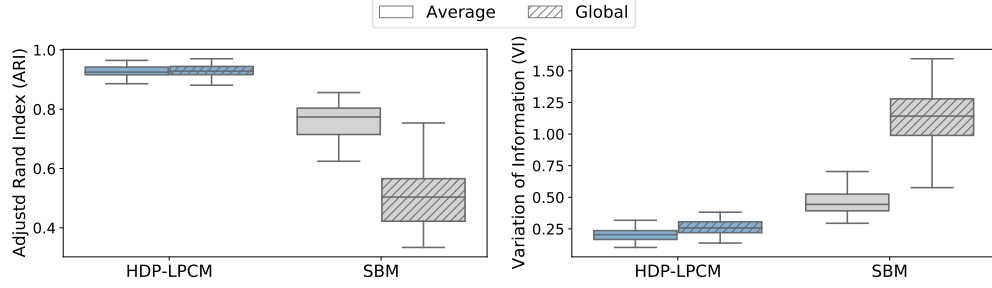


Figure 3.6: Boxplots of ARI (left) and VI (right) of the two competing methods on the time-inhomogeneous simulation.

strong group structure. Furthermore, the HDP-LPCM’s ability to predict missing and one-step ahead dyads remains good despite the evolving community structure. Once again, the advantage of the HDP-LPCM over the block models is apparent when comparing their clustering performance in Figure 3.6 where the SBM has an unsatisfactorily low global ARI.

Next, we evaluate the ability of the HDP-LPCM to quantify the changing macro-level community structure. The posterior probabilities for $|G_t|$ are displayed in Figure 3.7. For most time points the correct number of groups has the highest median posterior probability. The exception is at $t = 3$, which is right before the two groups split into six groups, where no group number has a median posterior probability that exceeds 0.5. As is common in Dirichlet process mixture models, the model tends to create a few small clusters, which can inflate the group count. This is the case here, where the extra clusters at $t = 3$ tend to be composed of less than five nodes. Once again, we assess our posterior summarization method’s ability to select a clustering with the correct number of communities. The number of groups selected by this method at each time step is displayed in Figure 3.8. For the majority of simulations, our method selects the correct number of clusters. However, the method struggles at $t = 3$ which is where the posterior is the most uncertain. However, the HDP-LPCM selects the true number of clusters, $|G_3| = 2$, the majority of the time despite often assigning $|G_3| = 3$ a higher posterior probability. The HDP-LPCM is able to recover because the VI estimator can select a clustering with the small spurious communities removed when such pruning is supported by the posterior co-occurrence matrix. In contrast, the SBMs fail to estimate the true community structure for $t = 1 - 6$ in the majority of simulations. Overall, we conclude that the HDP-LPCM is able to adequately infer evolving communities.

3.6 Real Data Application

In this section, we demonstrate the utility of our proposed HDP-LPCM on a variety of real-world dynamic networks with an evolving community structure. We include two applications: inferring changing international military alliances during the Cold War and detecting dynamic plotlines in the television series Game of Thrones. Also, in Section 3.8.11, we show that the HDP-LPCM corroborates many previous findings concerning Sampson’s monastery network (Sampson, 1969), a standard pedagogical example in the network literature.

For each analysis, we estimated the HDP-LPCM with a truncation level $L = 25$ using 50,000 iterations for tuning, 50,000 iterations for burn-in, which left a remaining 400,000 iterations for inference. The trace plots, autocorrelation functions (ACFs), and marginal posterior densities are contained in Section 3.8.12.

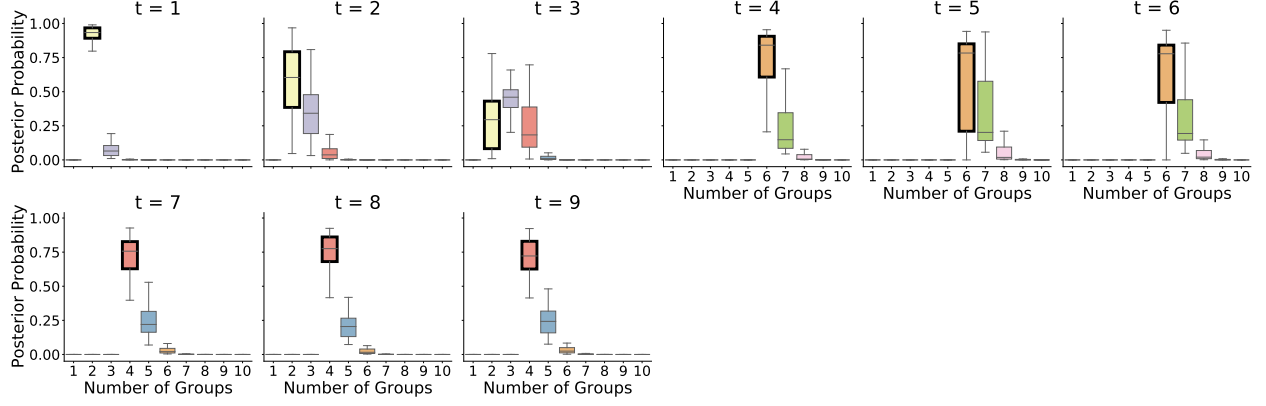


Figure 3.7: Boxplots showing the posterior distribution of $|G_t|$ over the 50 time-inhomogeneous simulations. Bolded boxes indicate the true number of communities.

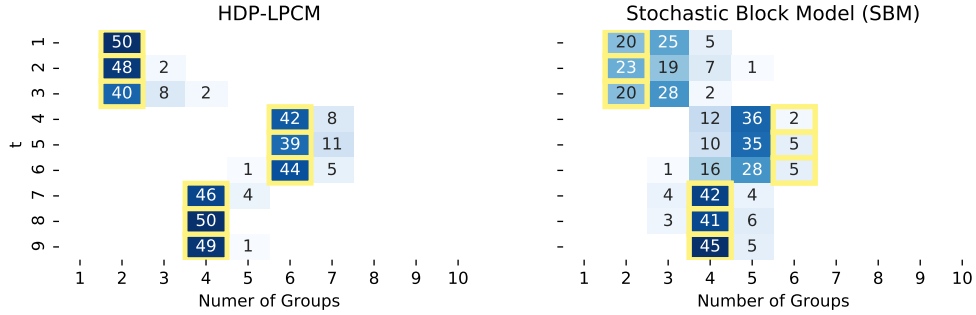


Figure 3.8: The number of groups estimated at each time step by the HDP-LPCM (left) and the static SBMs (right). The yellow cells contain the true number of communities.

Visual inspection of the trace plots indicates the algorithm converged on each dataset. Furthermore, the estimated model's in-sample AUC was 0.99 for each analysis, which indicates a good in-sample fit.

3.6.1 International Military Alliances

We begin by using the HDP-LPCM to examine changes in international military alliances during the first three decades of the Cold War (1950 - 1979). We use the Formal Alliances (v4.1) dataset curated as part of the Correlates of War Project (Gibler, 2009b). The raw dataset records all formal alliances – mutual defense pacts, non-aggression treaties, and ententes – among nations between 1816 and 2012. The goal of our analysis is to uncover the competing political blocs that defined the Cold War period in history and to determine any points in time where that alliance structure changed.

For this analysis, we use the yearly dyadic dataset, which records an undirected edge between two nations if there is a formal alliance between them during that year. To simplify the analysis, we discretized the dynamic networks into five year chunks from 1950 - 1979. We removed the nodes with a degree less than two from each network to focus on the larger political blocs found in the dataset. In addition, we required a nation to have at least one alliance during 1950 - 1979. We binarized the relations so that a connection between nations i and j at time t means they had at least one active alliance during those five years. This preprocessing resulted in $T = 6$ undirected binary networks that each contains $n = 107$ actors.

The alluvial diagram (Figure 3.9) of the partition selected with the procedure described in Section 3.4.2

shows that the HDP-LPCM estimates six communities overall with five communities active during all six time points. Note that the model uses group 2 to collect isolated nodes and small intermittent alliances. This interpretation is supported by the fact that $\hat{\sigma}_2 = 132$, which is 28 times larger than the second largest group shape $\hat{\sigma}_1 = 4.67$, and it encompasses most of the latent space. Thus we consider groups 1, 3, 4, 5, 6, and 7 as the major blocs inferred by the model. Note that these blocs strongly dictate the network's dynamics with an inferred blending coefficient $\lambda = 0.994$. Figure 3.10 depicts the estimated latent spaces for the years 1950 - 1954 and 1960 - 1964. We included the latent spaces for the remaining years in the supplementary material.

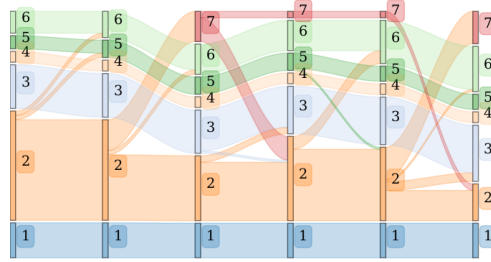


Figure 3.9: Alluvial diagram for the international military alliances network. Each line is the flux of nodes going from one group to the next at time t to $t + 1$. A line's thickness is proportional to the number of nodes and the total height represents the 107 nodes.

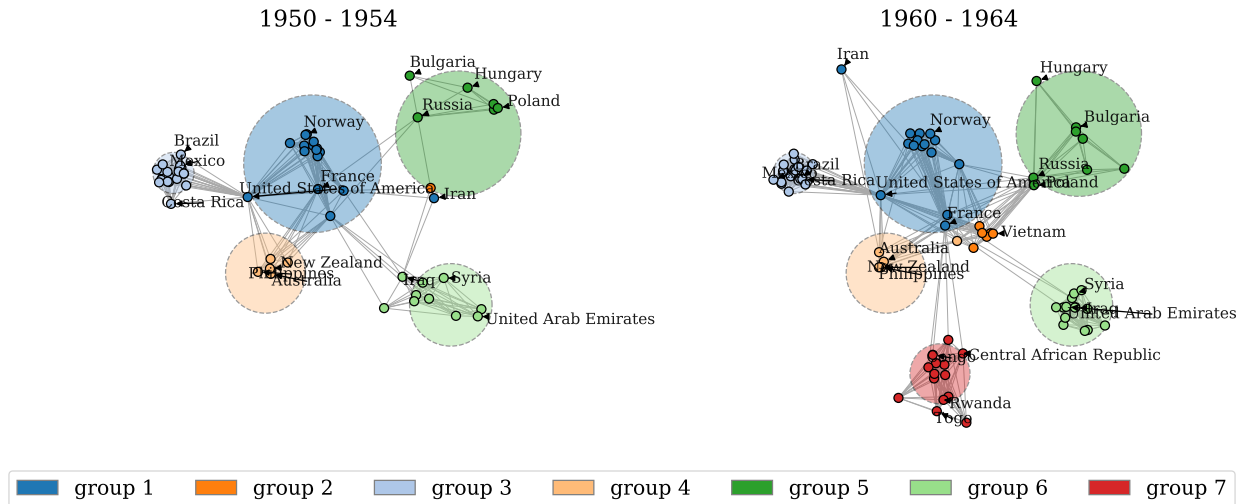


Figure 3.10: Latent space of international military alliances for the years 1950 - 1954 (left) and 1960 - 1964 (right). The group shapes σ_g are displayed as two-standard deviation ellipses centered on the group means μ_g . The names of select nations are annotated. The undirected edges are also displayed. For clarity, all unconnected nodes and group 2's two-standard deviation ellipse are removed.

The static communities (groups 1, 3, 4, 5, and 6) coincide with long term regional alliances during the Cold War. Group 1 corresponds to the Western Bloc, consisting primarily of nations that are a part of the North Atlantic Treaty Organization (NATO) and the Western European Union (WEU). The competing group 5 represents the Eastern Bloc, which consisting of the Soviet Union and its satellite states such as East Germany, Czechoslovakia, and Poland. Group 3 consists of the Latin American coalition of the Organization of American States (OAS), founded in 1948 to oppose socialism. Group 4 consists of member nations of the

Southeast Asia Treaty Organization (SEATO), which is the Asian equivalent of NATO. Group 6 is the Arab League (at the time the League of Arab States) formed in 1945 to protect the interests of Arab countries. As depicted in the alluvial diagram, an interesting finding of the HDP-LPCM is that there is little exchange of nations between these groups over this time period. This finding bolsters the claim that “once the ‘cold war’ confrontation became apparent ... many nations cast their lot with either the American or Soviet bloc” (Small & Singer, 1969).

The evolving community structure (birth of group 7) is a result of the emergence of Africa as a world power. In particular, seventeen African nations gained their independence in 1960 alone. As a result of these newly independent nations, a large number of regional alliances formed in the early 1960s (Gibler, 2009a). This is reflected in the HDP-LPCM by the emergence of group 7 at $t = 3$, which encompasses the newly independent African nations. Fewer African alliances formed over the next decade. Our model reflects this fact by only including the Union of Central African States, which consists of former French colonies, in group 7 at $t = 4$ and 5. Finally, the Economic Community of West African States (ECOWAS) formed in 1975, which resulted in the re-introduction of many western African states into group 7 at $t = 6$. The dynamic nature of the HDP-LPCM is essential in revealing the emergence of Africa as a world power and demonstrates the importance of incorporating community evolution in latent space network modeling.

3.6.2 Character Interactions in Game of Thrones

In this section, we study the networks of character interactions in the television series Game of Thrones. The goal of our analysis is to use community detection to pinpoint coherent dynamic plotlines within the series. This is an interesting case study because the series’ narrative is known for its many dynamic stories and characters who freely move between them. As such, we expect various groups of characters to form and die out across the series’ lifetime and actors to freely move between these groups.

We utilize the networks compiled by Beveridge and Chemers (2018), who parsed fan-generated scripts found on the user curation site Genius. The original dataset consists of weighted character-character interactions split up between the eight seasons of the television series. The weight of an edge equals the number of interactions during a given season. Since this is primarily a pedagogical example, we restrict the dataset to the first four seasons of the show. To remove minor characters, we only keep interactions that occur greater than or equal to 10 times each season. The final result is $T = 4$ binary undirected networks with a total of $n = 165$ actors each.

The alluvial diagram (Figure 3.11) and associated latent space (Figure 3.12) reveal a dynamic group structure. The model infers six groups overall; however, only three groups are active during the first season of the show. During the second season, group 2 splits off into groups 3, 4, 6, and 7. In addition, group 5 is created out of characters from group 6. Note that the model uses group 6 to collect inactive characters, so we exclude it from further analysis. After the change in season two, the network’s group structure remains constant for seasons three and four. The latent space of seasons three and four are included in the supplementary material. Furthermore, since the blending coefficient $\lambda = 0.974$, we conclude that the groups drive the evolution of the network. Overall, the ability of the HDP-LPCM to infer an evolving group structure is crucial for properly understanding the network’s dynamics.

The inferred groups and their dynamics are consistent with the storylines in the Game of Thrones series. During season one, there are two active groups. Group 2 consists of all characters on Westeros, while group 1 centers around Deanery Targaryen’s story on Essos. Starting at season two, group 2 contains characters revolving around Arya Stark’s story arc, House Lannister, and other characters at King’s Landing. Group

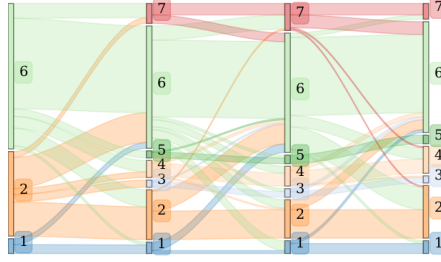


Figure 3.11: Alluvial diagram for the Game of Thrones character interaction networks.

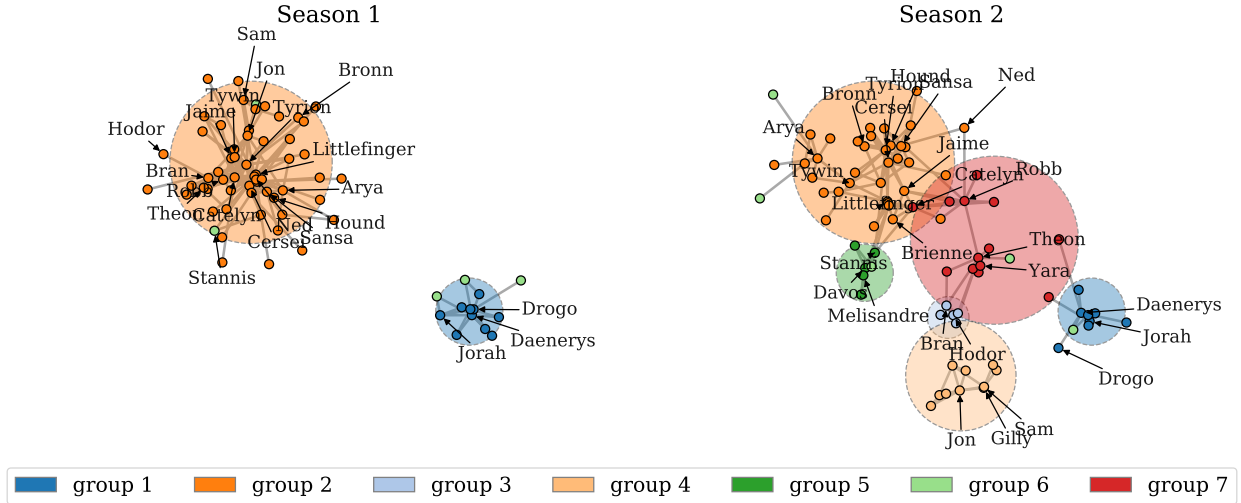


Figure 3.12: Latent space of character interactions in Game of Thrones for seasons 1 and 2. We removed all unconnected nodes and group 6’s two-standard deviation ellipse.

3 contains Bran Stark’s group of companions (Hodor, Rickon, Osha, Meera, and Jojen) as they travel north of the wall. The plotlines related to the remainder of House Stark are contained in group 7. This group is composed of both Robb Stark’s contingent as well as those related to Theon Greyjoy, who currently rules Winterfell the ancestral home of the Starks. Group 4 pertains to the story north of the wall. It most notably contains Jon Snow and Sam Tarly. Lastly, group 5 revolves around the story of Stannis Baratheon and his quest to regain the Iron throne, which was introduced in season two. In conclusion, our model recovers narratively coherent groups within the Game of Thrones character interaction networks and identifies their dynamics within the storyline.

3.7 Discussion

In this chapter, we proposed the hierarchical Dirichlet process latent position cluster model (HDP-LPCM) for dynamic networks. This is the first, to our knowledge, latent position model that can detect evolving community structures. In addition, we demonstrated that the HDP-LPCM still performs well when there is a static group structure. To accomplish our modeling goals, we used Bayesian nonparametric priors to provide simultaneous inference over the number of communities at each time point in the network and the dynamics of each actor’s latent position. Furthermore, our MCMC inference procedure has computational advantages over existing approaches. In particular, we avoided the BIC approximations of Handcock et al.

(2007) and the computationally expensive process of estimating a large number of models.

In this work, we focused solely on community detection for binary undirected networks; however, our method can easily be extended to other network types. For example, both directed and weighted networks can be accommodated through the minor modifications of the likelihood presented in Sewell and Chen (2016). Due to the HDP-LPCM’s conditional dependence structure, one only needs to modify the Metropolis-Hastings steps (i.e., steps 2 and 3) in Algorithm 3.1 to use the new network likelihood.

Common to most latent distance models, our proposed MCMC estimation method is time-intensive for large networks due to the quadratic scaling of the procedure. Specifically, the proposed sampler produced 3.3 and 4 samples per second on a 2015 MacBook Pro with a 2.2 GHz Intel Core i7 processor for the networks in Section 3.6.1 and Section 3.6.2, respectively. To alleviate this issue, one could use the case-control likelihood of Raftery et al. (2012). This method approximates the full likelihood by sub-sampling the unconnected edges, which allows for inference that scales linearly with the number of nodes. To further decrease the model’s runtime, one could create a variational Bayes or stochastic variational Bayes (Hoffman et al., 2013) algorithm. However, the lack of conjugacy in the distance model means such an algorithm is non-trivial to implement. For this reason, we leave these extensions to future work.

Although our model can infer the number of communities, it does not provide inference for the latent space’s dimension, p . In the numerical studies, we fixed $p = 2$, which allowed for visualization and was adequate for the networks considered in this work. Furthermore, we believe such visualizations allow for human-in-the-loop model scrutiny, which is a primary strength of the latent position model over other approaches such as latent feature models (Miller et al., 2009). Of course, selecting an appropriate dimension of the latent space is crucial if one is focused on tasks such as link prediction. A straightforward approach is to use a model selection criteria such as BIC or the Watanabe-Akaike information criterion (WAIC) (Watanabe, 2010). Another possibility is to put a prior over the dimensionality such as in Durante and Dunson (2014). Regardless, the selection of p remains an interesting and open question for future research.

Finally, while the HDP-LPCM’s assumption of static group locations matches certain real-world networks, one can devise situations more naturally described by moving group centers. For example, increased polarization between two groups resulting from the groups moving further apart over time. We chose to keep the groups fixed for two reasons: the simplification still captures the merging, splitting, creation, and deletion phenomena of interest, and it removes the non-identifiability issue of distinguishing between the creation of a group and the movement of a group to a new location. Devising a model that efficiently infers dynamic group centers is an area of future research interest.

Despite these possible extensions, we believe the HDP-LPCM has a broad range of applicability. As demonstrated on the real-world networks in this chapter, applications include the detection of changing alliances structures as well as inference regarding narrative plotlines. We believe our methodology can be built upon to provide new tools for understanding the sequential evolution of communities in dynamic networks.

3.8 Supplementary Material

Python-package for HDP-LPCMs: Python-package `dynetlsm` contains code to sample from the HDP-LPCM and all datasets and scripts used to produce the results in this chapter. This package is available at <https://github.com/joshloyal/dynetlsm>.

3.8.1 Conditional Distributions Used in the Gibbs Sampler

In this section, we present the conditional distributions used in the Metropolis-Hastings within Gibbs sampler for the HDP-LPCM. In what follows, let $I_{kt} = \{i : Z_t^i = k\}$ be the set of nodes in group k at time t . Furthermore, let $n_{kt} = |I_{kt}|$ be the number of actors in group k at time t , and let $n_{k\cdot} = \sum_{t=1}^T n_{kt}$. We denote conditioning on everything but the parameters of interest as $\cdot$. The full conditionals are

$$\begin{aligned} \boldsymbol{\mu}_k \mid \cdot &\sim N(\bar{\boldsymbol{\mu}}_k, \bar{\sigma}_k^2 I_p) \quad \text{where} \\ \bar{\sigma}_k^2 &= \left[\frac{1}{\sigma_k^2} \left(n_{k1} + \lambda^2 \sum_{t=2}^T n_{kt} \right) + \frac{1}{\tau^2} \right]^{-1}, \\ \bar{\boldsymbol{\mu}}_k &= \bar{\sigma}_k^2 \left(\frac{1}{\sigma_k^2} \sum_{i \in I_{k1}} \mathbf{X}_1^i + \frac{\lambda}{\sigma_k^2} \sum_{t=2}^T \sum_{i \in I_{kt}} (\mathbf{X}_t^i - (1-\lambda)\mathbf{X}_{t-1}^i) + \frac{1}{\tau^2} \boldsymbol{\mu}_0 \right), \end{aligned} \quad (3.20)$$

$$\begin{aligned} \sigma_k^2 \mid \cdot &\sim \Gamma^{-1}((n_{k\cdot}p + a)/2, \bar{b}/2) \quad \text{where} \\ \bar{b} &= b + \sum_{i \in I_{k1}} \|\mathbf{X}_1^i - \boldsymbol{\mu}_k\|_2^2 + \sum_{t=2}^T \sum_{i \in I_{kt}} \|\mathbf{X}_t^i - (1-\lambda)\mathbf{X}_{t-1}^i - \lambda\boldsymbol{\mu}_k\|_2^2, \end{aligned} \quad (3.21)$$

$$\begin{aligned} \lambda \mid \cdot &\sim N_{(0,1)}(\bar{\mu}_\lambda, \bar{\sigma}_\lambda^2) \quad \text{where} \\ \bar{\mu}_\lambda &= \frac{\mu_\lambda + \sigma_\lambda^2 \sum_{i=1}^n \sum_{t \geq 2} \frac{1}{\sigma_{Z_t^i}^2} (\mathbf{X}_t^i - \mathbf{X}_{t-1}^i)^T (\boldsymbol{\mu}_{Z_t^i} - \mathbf{X}_{t-1}^i)}{1 + \sigma_\lambda^2 \sum_{i=1}^n \sum_{t \geq 2} \frac{1}{\sigma_{Z_t^i}^2} \|\boldsymbol{\mu}_{Z_t^i} - \mathbf{X}_{t-1}^i\|_2^2}, \\ \bar{\sigma}_\lambda^2 &= \left(\frac{1}{\sigma_\lambda^2} + \sum_{i=1}^n \sum_{t \geq 2} \frac{1}{\sigma_{Z_t^i}^2} \|\boldsymbol{\mu}_{Z_t^i} - \mathbf{X}_{t-1}^i\|_2^2 \right)^{-1}, \end{aligned} \quad (3.22)$$

$$\tau^2 \mid \cdot \sim \Gamma^{-1}((a_\tau + L)/2, (b_\tau + \sum_{g=1}^L \|\boldsymbol{\mu}_g\|_2^2)/2), \quad (3.23)$$

$$b \mid \cdot \sim \Gamma((c + L a)/2, 2(d + \sum_{g=1}^L \sigma_g^{-2})^{-1}). \quad (3.24)$$

The log of the full conditional distributions used in the MH updates of the latent positions $\mathcal{X}_{1:T}$ are

(a) for $t = 1$:

$$\begin{aligned} \log(p(\mathbf{X}_1^i \mid \cdot)) &\propto \log(p(Y_1 \mid \mathcal{X}_1, \beta_0)) - \frac{1}{2\sigma_{Z_1^i}^2} \|\mathbf{X}_1^i - \boldsymbol{\mu}_{Z_1^i}\|_2^2 \\ &\quad - \frac{1}{2\sigma_{Z_2^i}^2} \|\mathbf{X}_2^i - \lambda\boldsymbol{\mu}_{Z_2^i} - (1-\lambda)\mathbf{X}_1^i\|_2^2, \end{aligned} \quad (3.25)$$

(b) for $1 < t < T$:

$$\begin{aligned} \log(p(\mathbf{X}_1^i | \cdot)) &\propto \log(p(Y_t | \mathcal{X}_t, \beta_0)) \\ &\quad - \frac{1}{2\sigma_{Z_t^i}^2} \|\mathbf{X}_t^i - \lambda\boldsymbol{\mu}_{Z_t^i} - (1-\lambda)\mathbf{X}_{t-1}^i\|_2^2 \\ &\quad - \frac{1}{2\sigma_{Z_{t+1}^i}^2} \|\mathbf{X}_{t+1}^i - \lambda\boldsymbol{\mu}_{Z_{t+1}^i} - (1-\lambda)\mathbf{X}_t^i\|_2^2, \end{aligned} \quad (3.26)$$

(c) for $t = T$:

$$\begin{aligned} \log(p(\mathbf{X}_T^i | \cdot)) &\propto \log(p(Y_T | \mathcal{X}_T, \beta_0)) \\ &\quad - \frac{1}{2\sigma_{Z_T^i}^2} \|\mathbf{X}_T^i - \lambda\boldsymbol{\mu}_{Z_T^i} - (1-\lambda)\mathbf{X}_{T-1}^i\|_2^2. \end{aligned} \quad (3.27)$$

Finally the log of the full conditional for the intercept parameter used in the MH step is

$$\log(p(\beta_0 | \cdot)) \propto \log(p(Y_{1:T} | \mathcal{X}_{1:T}, \beta_0)) - \frac{1}{2\sigma_{\beta_0}^2} \|\beta_0 - \mu_{\beta_0}\|_2^2. \quad (3.28)$$

3.8.2 Sampling $\mathcal{Z}_{1:T}$

Although previous work on dynamic latent position cluster models (Sewell & Chen, 2017) update $\mathcal{Z}_{1:T}$ coordinate wise, we chose to utilize a blocked Gibbs sampler. In particular, we use a variant of the forward-backward algorithm for AR-HMMs (Rabiner, 1989) to jointly sample an actor's group assignments $Z_{1:T}^i$. We made this change to increase the mixing rate of the sampler. Previous work has shown that sampling $Z_{1:T}^i$ as a block as opposed to coordinate wise can lead to drastically faster mixing rates in HMMs. This is especially the case when the elements of $Z_{1:T}^i$ are highly related in the posterior distribution, which is likely due to our stickiness assumption (Scott, 2002). The full sampling algorithm is detailed in Algorithm 3.2. A similar sampler is found in E. Fox et al. (2011); however, they do not consider multiple state sequences with time-inhomogeneous transition probabilities as we do in this work.

We now demonstrate how to jointly sample an actor's group assignments $Z_{1:T}^i$. Recall that the model assumes that each actor's trajectory is an iid AR-HMM conditioned on the transition and emission parameters. To sample the group assignments as a block, we begin by factoring the joint conditional distribution in a forward fashion:

$$p(Z_{1:T}^i | \mathbf{X}_{1:T}^i) = p(Z_1^i | \mathbf{X}_{1:T}^i) \prod_{t=2}^T p(Z_t^i | Z_{t-1}^i, \mathbf{X}_{1:T}^i), \quad (3.29)$$

where we are implicitly conditioning on the model parameters $\boldsymbol{\pi}_0^1, \Pi_{2:T}, \boldsymbol{\mu}_{1:L}$, and $\sigma_{1:L}^2$ for clarity. The strategy is then to sample the cluster assignments according to this forward factorization. At the first time step we sample $Z_1^i \sim p(Z_1^i | \mathbf{X}_{1:T}^i)$, and then we draw $Z_t^i \sim p(Z_t^i | Z_{t-1}^i, \mathbf{X}_{1:T}^i)$ for $t = 2, \dots, T$. These conditional distributions can be efficiently sampled using backwards message variables. Specifically, we define the backwards message variables recursively as

$$m_{t,t-1}(Z_{t-1}^i) \propto \begin{cases} \sum_{Z_t^i} p(Z_t^i | Z_{t-1}^i) p(\mathbf{X}_t^i | Z_t^i, \mathbf{X}_{t-1}^i) m_{t+1,t}(Z_t^i), & t \leq T, \\ 1, & t = T + 1. \end{cases} \quad (3.30)$$

We can then express the conditional distributions in terms of the backwards message variables as follows:

$$\begin{aligned}
p(Z_t^i \mid Z_{t-1}^i, \mathbf{X}_{1:T}^i) &\propto p(Z_t^i, Z_{t-1}^i \mid \mathbf{X}_{1:T}^i) \\
&= \sum_{Z_{1:(t-2)}^i} \sum_{Z_{(t+1):T}^i} p(Z_{1:T}^i, \mathbf{X}_{1:T}^i) \\
&\propto p(Z_t^i \mid Z_{t-1}^i) p(\mathbf{X}_t^i \mid Z_t^i, \mathbf{X}_{t-1}^i) \sum_{Z_{(t+1):T}^i} \prod_{s=t+1}^T p(Z_s^i \mid Z_{s-1}^i) p(\mathbf{X}_s^i \mid Z_s^i, \mathbf{X}_{s-1}^i) \\
&= p(Z_t^i \mid Z_{t-1}^i) p(\mathbf{X}_t^i \mid Z_t^i, \mathbf{X}_{t-1}^i) \sum_{Z_{t+1}^i} p(Z_{t+1}^i \mid Z_t^i) p(\mathbf{X}_{t+1}^i \mid Z_{t+1}^i, \mathbf{X}_t^i) m_{t+2,t+1}(Z_{t+1}^i) \\
&= p(Z_t^i \mid Z_{t-1}^i) p(\mathbf{X}_t^i \mid Z_t^i, \mathbf{X}_{t-1}^i) m_{t+1,t}(Z_t^i).
\end{aligned} \tag{3.31}$$

Given transition probabilities $\Pi_{1:T}$, group parameters $\boldsymbol{\mu}_{1:L}, \sigma_{1:L}^2$, and latent positions $\mathcal{X}_{1:T}$, for each $i \in \{1, \dots, n\}$, update actor labels as follows:

1. Working sequentially backwards in time, calculate messages $m_{t,t-1}(k)$:

- (a) For each $k \in \{1, \dots, L\}$, initialize messages to

$$m_{T+1,T}(k) = 1.$$

- (b) For each $t \in \{T, \dots, 2\}$ and for each $k \in \{1, \dots, L\}$, compute

$$m_{t,t-1}(k) = \sum_{j=1}^L \pi_{kj}^t N(\mathbf{X}_t^i \mid \lambda \boldsymbol{\mu}_j + (1-\lambda) \mathbf{X}_{t-1}^i, \sigma_j^2 I_p) m_{t+1,t}(j), \quad t > 1.$$

2. Sample state assignments $Z_{1:T}^i$ working sequentially forward in time. In addition, update the cluster counts starting with $n_{jkt} = 0$. For $t \in \{1, \dots, T\}$:

- (a) For each $k \in \{1, \dots, L\}$, compute the probability

$$f_k(\mathbf{X}_t^i) = \begin{cases} \pi_{0k}^1 N(\mathbf{X}_1^i \mid \boldsymbol{\mu}_k, \sigma_k^2 I_p) m_{2,1}(k), & t = 1, \\ \pi_{Z_{t-1}^i k}^t N(\mathbf{X}_t^i \mid \lambda \boldsymbol{\mu}_k + (1-\lambda) \mathbf{X}_{t-1}^i, \sigma_k^2 I_p) m_{t+1,t}(k), & t > 1. \end{cases}$$

- (b) Sample a state assignment Z_t^i :

$$Z_t^i \sim \sum_{k=1}^L f_k(\mathbf{X}_t^i) \delta(Z_t^i, k).$$

Algorithm 3.2: Blocked Gibbs sampler for actor labels $Z_{1:T}^i$.

3.8.3 Sampling β

We follow the sampling strategy outlined in Teh et al. (2006) and E. B. Fox et al. (2011). These samplers use the Chinese restaurant franchise (CRF) metaphor for the hierarchical Dirichlet process to derive closed-form conditional distributions. In the CRF metaphor, each group $j \in \{1, \dots, J\}$ in Equation (3.12) is associated with a restaurant. The entire collection of restaurants (or franchise) shares a common menu of

dishes $\theta_{1:K}$, where K is the total number of draws from the base measure H . The metaphor describes the probabilistic mechanism used by the HDP to partition a collection of individuals (or customers) pre-assigned to J restaurants into K clusters that are identified by their common dish assignments.

For completeness and to properly describe the auxiliary variable samplers in the next section, we present a brief description of the CRF metaphor. As customers arrive at a restaurant, the host seats them at an occupied table with probability proportional to the number of people already seated there and assigns them to a new table with a probability proportional to α . When the first customer arrives at a table, a single dish from the global menu is served. The probability to receive an existing dish is proportional to the number of times that dish is served across all restaurants, and a new dish is prepared with probability proportional to γ . For the sticky HDP, the dish assignment process is modified to account for the stickiness parameter κ . In this case, each restaurant has a specialty dish. Once the waiter receives a table's order, the waiter has a probability $\rho = \kappa/(\alpha + \kappa)$ to override that order and serve the restaurant's specialty dish instead.

In the HDP-LPCM, we identify each row of the transition matrices, $\{\pi_g^t\}$, with a restaurant in the franchise. In other words, each pair (g, t) indexes the restaurants in the CRF metaphor in a way that a bijective map exists between restaurant $j \mapsto (g, t)$. Furthermore, the customers assigned to restaurant j correspond to all actors assigned to group g at time $t-1$. With these mappings in mind, we can directly apply the re-sampling algorithm for β described in E. B. Fox et al. (2011) with a few straightforward modifications to account for the multiple hidden state sequences.

3.8.4 Sampling $\mathbf{m}$, $\bar{\mathbf{m}}$, and $\mathbf{w}$

The sampler for β and the hyperparameter samplers presented in the next section require the introduction of three auxiliary variables $\mathbf{m}$, $\bar{\mathbf{m}}$, and $\mathbf{w}$. These variables keep track of certain statistics of the CRF sampling process that when made available preserve conjugacy in the model. The components of these variables are as follows: m_{jks} is the number of tables at restaurant j that ordered dish k at time s , $\bar{m}_{jks}$ is the number of tables at restaurant j that were served dish k at time s , and w_{jts} is a binary variable indicating whether table t at restaurant j at time s had their dish choice overridden by the waiter. Note that m_{jks} and $\bar{m}_{jks}$ may differ due to the waiter overriding a table's order. The algorithm derived to sampler these auxiliary variables is similar to the one found in E. B. Fox et al. (2011) with modifications that account for the increased number of restaurants. We present our auxiliary variable sampler in Algorithm 3.3. The sampler is based on step 3 of Algorithm 1 of E. B. Fox et al. (2011).

3.8.5 Hyperparameter Settings and Priors

The HDP-LPCM has many hyperparameters that must be set by the practitioner. If one has no a priori knowledge of appropriate hyperparameter settings, it can be advantageous to incorporate hyperparameter exploration into the sampling procedure. In this section, we detail additional hyperparameter samplers as well as our initial choices of hyperparameter values used in the applications of the following sections. The full hyperparameter sampler is presented in Algorithm 3.5 in Section 3.8.6 of the supplementary material.

In our experience, the hyperparameters that influence inference the most are τ^2 , b , γ , α_0 , α , and κ . The prior variance of the group means τ^2 controls the scale of the latent space. The prior scale of the group variances b controls the size of the clusters in the network. The concentration parameters γ , α_0 , and α influence the a priori number of clusters in the network with higher values indicating more groups. Finally, larger values of κ result in a priori higher probabilities of self-transitions. With this intuition about the role

Given n_{jks} , the number transitions from state j to state k at time s , and $\rho = \kappa/(\alpha + \kappa)$, sample the auxiliary variables $\mathbf{m}, \bar{\mathbf{m}}, \mathbf{w}$ as follows:

1. For each $k \in \{1, \dots, L\}$, set $m_{0k1} = \bar{m}_{0k1} = 0$, and $n = 0$. For each customer in restaurant 0 eating dish k at time 1, that is for $i = 1, \dots, n_{0k1}$, sample

$$x \sim \text{Bernoulli} \left(\frac{\alpha_0 \beta_k}{n + \alpha_0 \beta_k} \right).$$

Increment n by one, and if $x = 1$ increment m_{0k1} and $\bar{m}_{0k1}$ by one.

2. For each $(j, k) \in \{1, \dots, L\}^2$ and $s \in \{2, \dots, T\}$, set $m_{jks} = 0$, and $n = 0$. For each customer in restaurant j eating dish k at time s , that is for $i = 1, \dots, n_{jks}$, sample

$$x \sim \text{Bernoulli} \left(\frac{\alpha \beta_k + \kappa \delta(j, k)}{n + \alpha \beta_k + \kappa \delta(j, k)} \right).$$

Increment n by one, and if $x = 1$ increment m_{jks} by one.

3. For each $j \in \{1, \dots, L\}$ and $s \in \{2, \dots, T\}$, sample the number of overridden dish orders in restaurant j at time s :

$$w_{j \cdot s} \sim \text{Binomial} \left(m_{jjs}, \frac{\rho}{\rho + \beta_j(1 - \rho)} \right).$$

Set the number of tables in restaurant j ordering dish k at time s to:

$$\bar{m}_{jks} = \begin{cases} m_{jks}, & j \neq k, \\ m_{jjs} - w_{j \cdot s}, & j = k. \end{cases}$$

Algorithm 3.3: Sampler for auxiliary variables needed for the update of the global transition distribution β . Note that we do not need to incorporate overridden dishes for m_{0k1} because we draw π_0^1 from a $\text{DP}(\alpha_0, \beta)$, which has no stickiness parameter κ .

of these parameters, we proceed to describe our choice of hyperparameter priors and initial values.

Determining the scale of the latent space is a difficult task a priori due to the unobserved nature of the space. As such, we place the following priors on τ^2 and b :

$$\tau^2 \sim \Gamma^{-1}(a_\tau/2, b_\tau/2), \quad b \sim \Gamma(c/2, 2/d), \quad (3.32)$$

where $\Gamma^{-1}(a, b)$ indicates an Inverse Gamma distribution with shape parameter a and scale parameter b , and $\Gamma(c, d)$ represents a Gamma distribution with shape and scale parameters c and d , respectively. Under these priors, the updates for the variance of the cluster means τ^2 and the scale parameter for the group variances b are conjugate. We use a heuristic similar to the `latentnet` package (Krivitsky & Handcock, 2008) to set the remaining hyperparameter values. In particular, we expect the size of the latent space to grow with the number of actors n . For this reason, we chose a_τ and b_τ such that τ^2 has mean $\mathbb{E}[\tau^2] = n^{2/p}/50$ and standard deviation $4\mathbb{E}[\tau^2]$, where n is the number of nodes and p is the dimension of the latent space. In addition, we chose b, c , and d such that the mode of σ^2 is τ^2 with a standard deviation of $4\mathbb{E}[b]$. Furthermore, the initial values for τ^2 and b are set to their prior mean values and $a = 2$. The idea behind this choice is to set a prior that can generate group shapes ranging from the scale of the entire latent space to a much smaller scale.

To update the hyperparameters of the HDP prior γ, α_0, α , and κ , we utilize a straightforward modification

of the samplers in Teh et al. (2006) and E. B. Fox et al. (2011). Both samplers employ or extend the sampler developed in Escobar and West (1995), which we describe in Algorithm 3.4 in Section 3.8.6 of the supplementary material. For all simulations and data applications, we place the following disperse priors on the concentration parameters:

$$\gamma \sim \Gamma(1, 10), \quad \alpha_0 \sim \Gamma(1, 1), \quad \alpha + \kappa \sim \Gamma(5, 10), \quad \rho \sim \text{Beta}(8, 2). \quad (3.33)$$

Note that under these priors, $\mathbb{E}[\rho] = 0.8$, which matches our large prior belief in self-transitions. The initial values are $\gamma = 1$, $\alpha = \alpha_0 = 1$, and $\kappa = 4$. In our experience, the sampler is insensitive to these initial values.

3.8.6 Sampling the Hyperparameters

In this section, we present the samplers for the hyperparameters of the HDP-LPCM. The full conditionals for the prior variance of the group means τ^2 and the prior scale of the group variances b are conjugate given our choice of hyperpriors. These full conditionals are displayed in Equations (3.23) and (3.24) respectively. The remaining hyperparameters are the concentration parameters of the HDP. To sample these hyperparameters, we use the auxiliary variable samplers developed in Teh et al. (2006) and E. B. Fox et al. (2011), which we slightly modified to account for the resampling of transition distributions. These samplers utilize the auxiliary variable sampler developed by Escobar and West (1995). For completeness, we include this sampler in Algorithm 3.4. Our full hyperparameter sampler is detailed in Algorithm 3.5.

The auxiliary variable sampler for the concentration parameter γ in a $\text{DP}(\gamma, H)$ presented in Escobar and West (1995). Note that a $\Gamma(a, b^{-1})$ hyperprior is placed on γ .

1. Sample $\eta \sim \text{Beta}(\gamma + 1, N)$.
2. Define $\tilde{\pi}_1 = a + K - 1$, $\tilde{\pi}_2 = N(b - \log(\eta))$, then define $\pi = \tilde{\pi}_1 / (\tilde{\pi}_1 + \tilde{\pi}_2)$.
3. Sample the concentration parameter from a mixture of Gamma distributions

$$\gamma \sim \pi \Gamma(a + K, [b - \log(\eta)]^{-1}) + (1 - \pi) \Gamma(a + K - 1, [b - \log(\eta)]^{-1}).$$

Algorithm 3.4: Auxiliary variable sampler used to sample concentration parameters developed by Escobar and West (1995).

3.8.7 Initialization Scheme

Due to the high dimensionality of our model, we can greatly reduce the number of iterations required to reach convergence by choosing good initial values for the model parameters. To initialize the latent positions $\mathcal{X}_{1:T}^{(1)}$ and the intercept $\beta_0^{(1)}$, we used the MAP estimate from a short MCMC chain (1,000 iterations) of the model with no clustering of Sewell and Chen (2015b). Note that we modified their sampler to use the undirected likelihood with the linear predictor given in Equation (3.5). Due to the block sampling of the node labels $\mathcal{Z}_{1:T}$ and the fact that they are sampled before the other clustering parameters, the sampler is not affected by the initial values of the node labels. We initialized the group locations and shapes $(\mu_{1:L}, \sigma_{1:L}^2)^{(1)}$ by running a k-means algorithm for longitudinal data (Genolini & Falissard, 2010) on the initial latent positions with $k = L$ and using the empirical mean and variance estimates of the groups as initial values. We set the initial blending coefficient $\lambda^{(1)} = 0.9$ to encourage early clustering. We sampled the remaining parameters $\beta^{(1)}$,

Note that we set the following priors on the HDP parameters:

$$\begin{aligned}\gamma &\sim \Gamma(a_\gamma, b_\gamma^{-1}), \\ \alpha_0 &\sim \Gamma(a_{\alpha_0}, b_{\alpha_0}^{-1}), \\ \alpha + \kappa &\sim \Gamma(a_{\alpha+\kappa}, b_{\alpha+\kappa}^{-1}), \\ \rho &\sim \text{Beta}(a_\rho, b_\rho).\end{aligned}$$

Given a previous set of hyperparameters $(\tau^2)^{(\ell-1)}$, $b^{(\ell-1)}$, $\gamma^{(\ell-1)}$, $\alpha_0^{(\ell-1)}$, $\alpha^{(\ell-1)}$, and $\kappa^{(\ell-1)}$, update the current parameters as follows:

1. Initialize current parameters to the values of the $(\ell - 1)$ th iteration.
2. Update prior variance:

$$\tau^2 \sim \Gamma^{-1}((a_\tau + L)/2, (b_\tau + \sum_{g=1}^L \|\boldsymbol{\mu}_g\|_2^2)/2).$$

3. Update prior scale:

$$b \sim \Gamma((c + L a)/2, 2(d + \sum_{g=1}^L \sigma_g^{-2})^{-1}).$$

4. Update γ via Algorithm 3.4 with $K = \sum_{k=1}^L \mathbb{1}_{\{\bar{m}_{\cdot k} > 0\}}$, $N = \bar{m}_{\dots}$, $a = a_\gamma$, and $b = b_\gamma$.
5. Update α_0 via Algorithm 3.4 with $K = m_{0.1}$, $N = n$, $a = a_{\alpha+\kappa}$, and $b = b_{\alpha+\kappa}$.
6. Update $\alpha + \kappa$:

For $g \in \{1, \dots, L\}$ and $t \in \{2, \dots, T\}$ sample

$$\begin{aligned}r_{gt} &\sim \text{Beta}(\alpha + \kappa + 1, n_{g \cdot t}), \\ s_{gt} &\sim \text{Bernoulli}\left(\frac{n_{g \cdot t}}{n_{g \cdot t} + \alpha + \kappa}\right).\end{aligned}$$

Update concentration parameter:

$$\alpha + \kappa \sim \Gamma(a_{\alpha+\kappa} + m_{\dots} - s_{\dots}, [b_{\alpha+\kappa} - \sum_{g=1}^L \sum_{t=2}^T \log(r_{gt})]^{-1}).$$

7. Update ρ :

$$\rho \sim \text{Beta}(w_{\dots} + a_\rho, \sum_{s=2}^T m_{\dots s} - w_{\dots} + b_\rho),$$

where we exclude the considered dishes at $s = 1$ because the initial restaurant does not include a stickiness parameter.

Algorithm 3.5: Samplers for the hyperparameters in the HDP-LPCM. In all applications, we set $a_{\alpha+\kappa} = 5$, $b_{\alpha+\kappa} = 0.1$, $a_\rho = 8$, $b_\rho = 2$, $a_\gamma = 1$, $b_\gamma = 0.1$, $a_{\alpha_0} = 1$, and $b_{\alpha_0} = 1$.

$(\boldsymbol{\pi}_0^1)^{(1)}$, and $\Pi_{2:T}^{(1)}$ from their respective priors. We set the prior parameters on the intercept to $\mu_{\beta_0} = \beta_0^{(1)}$ and $\sigma_{\beta_0}^2 = 2$. Finally, we set the prior parameters on the blending coefficient to $\mu_\lambda = 0.9$ and $\sigma_\lambda^2 = 0.01$, which results in a truncated normal prior with an interquartile range of roughly (0.82, 0.93). This reflects our prior belief that the dynamic networks have a strong community structure. Note that since λ only governs the prior dynamics of the latent positions and does not appear in the likelihood, we found that the latent position estimates are relatively insensitive to this prior specification with enough data. Furthermore, we demonstrate that the estimation procedure is insensitive to these hyperparameter values in the sensitivity analysis of Section 3.8.10.

3.8.8 Slowly Merging Communities

In Section 3.3.2 of the main text, we briefly described how the HDP-LPCM could capture the gradual merging of two communities. Here, we expand on the generative model and demonstrate that our inference procedure actually recovers this process. We consider a network with three groups overall that is observed over $T = 5$ time steps. Specifically, the simulated networks have group centers

$$(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3) = \begin{pmatrix} 5 & -5 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

equal group shapes $\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = 1$, and an initial distribution $\boldsymbol{\pi}_0^1 = (1/2, 1/2)$. The time-inhomogeneous transition distributions are

$$\Pi_t = \begin{pmatrix} 1 - (t-1)/4 & 0 & (t-1)/4 \\ 0 & 1 - (t-1)/4 & (t-1)/4 \\ 0 & 0 & 1 \end{pmatrix},$$

for $t = 2, 3$, and $\Pi_4 = \Pi_5 = (\mathbf{0}_3, \mathbf{0}_3, \mathbf{1}_3)$, where $\mathbf{0}_3$ and $\mathbf{1}_3$ indicate 3-dimensional vectors of zeros and ones, respectively. These time-inhomogeneous transition matrices result in nodes being absorbed into group 3. In addition, we set $\lambda = 0.6$ so that the latent positions slower wander over to the center of the latent space. The first and second row of Figure 3.13 display the adjacency matrices and a latent space sampled from this generative model with $n = 120$ nodes. Next, we estimated the HDP-LPCM with $L = 10$ on the observed adjacency matrices using 10,000 samples after burn-in and tuning. The estimated latent space and cluster assignments are displayed on the third row of Figure 3.13. Note that we matched the latent space to the ground truth with a Procrustes rotation. We see that the HDP-LPCM is able to deduce the three community structure without introducing extra communities. In particular, the HDP-LPCM does not introduce two additional communities at $t = 2$ in order to cluster the nodes moving towards the third group center. Finally, we note that the global ARI is high at 0.95 for this simulation. Also the 95% credible interval for the blending coefficient λ is (0.603, 0.682), which is close to the actual value of 0.6.

3.8.9 Sensitivity to Group Separation and Stability

We present a simplified simulation study to evaluate the ability of the HDP-LPCM to recover a network's underlying community structure as the two most significant variables that influence dynamic community recovery vary: the separation between groups and the rate at which nodes switch groups. For simplicity, we simulated networks with a time-homogeneous community structure that has $|G_t| = 2$ groups for $1 \leq t \leq T$.

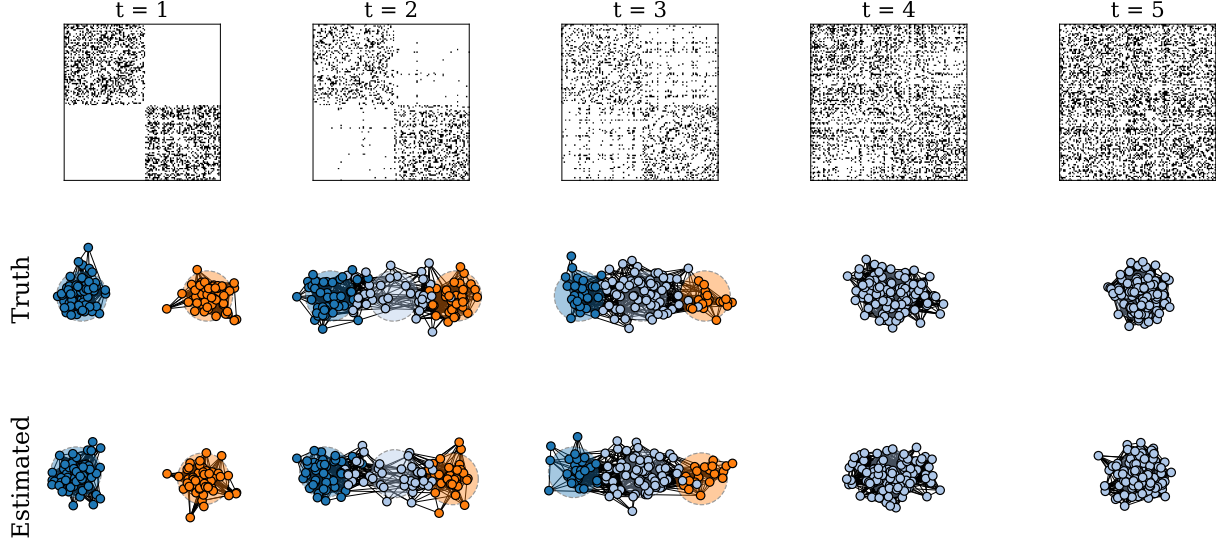


Figure 3.13: Observed adjacency matrices sorted by Z_1 (top), the true latent space (middle), and the estimated latent space (bottom) that results in two groups gradually merging under the HDP-LPCM.

The locations of the two groups are

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu} = (\mu, 0)' \quad \text{and} \quad \boldsymbol{\mu}_2 = -\boldsymbol{\mu} = (-\mu, 0)',$$

with equal group shapes $\sigma_1^2 = \sigma_2^2 = 0.25$. We set the initial distribution to $\boldsymbol{\pi}_0^1 = (1/2, 1/2)$, $\beta_0 = 1.0$ and $\lambda = 0.8$. We considered equal transition probabilities $\pi_{12}^t = \pi_{21}^t = \pi_{trans}$ for $1 \leq t \leq T$.

This model is a dynamic LPCM equivalent to the dynamic planted-partition model, where σ^2 and μ control the expected within-group and between-group probabilities. By setting the group shapes to 0.25, we fixed the initial probability of an edge between two arbitrary actors in the same group, $p_{in} = \mathbb{P}(Y_{ij1} = 1 \mid Z_1^i = Z_1^j, \beta_0, \boldsymbol{\theta})$, to roughly 0.44. The group separation, 2μ , governs the initial probability of an edge between two arbitrary actors in different groups, $p_{out} = \mathbb{P}(Y_{ij1} = 1 \mid Z_1^i \neq Z_1^j, \beta_0, \boldsymbol{\theta})$. We chose values of μ to give different levels of group separation indicated by a parameter $r = p_{out}/p_{in}$. Note that $r = 0$ results in completely separated communities while $r = 1$ results in indistinguishable communities.

In more detail, we define the following marginal probability

$$\begin{aligned} p_s &= \iint \mathbb{P}(Y_{ij1} = 1 \mid \mathbf{X}_1^i, \mathbf{X}_1^j, \beta_0) \mathbb{P}(\mathbf{X}_1^i, \mathbf{X}_1^j \mid Z_1^i, Z_1^j, \boldsymbol{\theta}) d\mathbf{X}_1^i d\mathbf{X}_1^j \\ &= \iint \frac{\exp(\beta_0 - \|\mathbf{X}_1^i - \mathbf{X}_1^j\|_2)}{1 + \exp(\beta_0 - \|\mathbf{X}_1^i - \mathbf{X}_1^j\|_2)} N(\mathbf{X}_1^i \mid \boldsymbol{\mu}, \sigma^2 I_2) N(\mathbf{X}_1^j \mid s \boldsymbol{\mu}, \sigma^2 I_2) d\mathbf{X}_1^i d\mathbf{X}_1^j. \end{aligned}$$

Setting $s = 1$ gives the initial probability of an edge between two arbitrary actors in group 1 while fixing $s = -1$ gives the initial probability of an edge between two arbitrary actors in group 1 and group 2. By symmetry and the fact that $\boldsymbol{\mu}_2 = -\boldsymbol{\mu}_1$ is a reflection about the origin, p_1 is also the expected probability of an edge between two actors in group 2. It follows that $p_{in} = p_1$ and $p_{out} = p_{-1}$. To select the value of μ with an appropriate r , we calculated r for a grid of 100 evenly spaced values of $\mu \in [0.1, 2.5]$ via Monte Carlo with 10,000 samples per point. This resulted in a look-up table we used to find the value of μ resulting in the closest match to r .

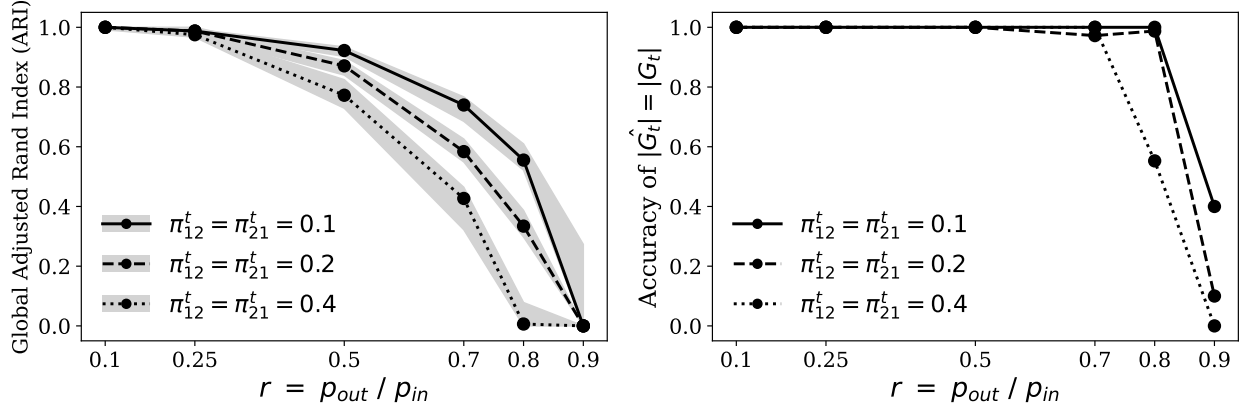


Figure 3.14: The HDP-LPCM’s median global ARI (left) and group detection accuracy (right) as a function of the community separation for different transition probabilities. Shaded regions denote the interquartile range over the 20 simulations.

For each combination of $r \in \{0.1, 0.25, 0.5, 0.7, 0.8, 0.9\}$ and $\pi_{trans} \in \{0.1, 0.2, 0.4\}$, we generated 20 networks with $n = 120$ nodes and $T = 4$ time steps. For each simulation, we estimated the HDP-LPCM using a Markov chain consisting of 50,000 samples. We used the initial 5,000 samples for tuning and the following 10,000 samples as burn-in leaving 35,000 samples for inference. We fixed the truncation level to $L = 5$. We calculated the global ARI for each simulation and recorded the fraction of networks over all simulations and time steps where the HDP-LPCM was able to detect two communities.

Figure 3.14 contains the simulation results. As we expected, as either the community overlap r or transition probability π_{trans} increases, the clustering performance deteriorates, and the true value of $|G_t|$ becomes harder to identify. Specifically, the HDP-LPCM fails to distinguish the two communities when $r = 0.9$ and $\pi_{trans} = 0.2$, which decreases to $r = 0.8$ when π_{trans} increases to 0.4. However, for $r \leq 0.5$, the HDP-LPCM performs well for all settings. Furthermore, in the low transition regime where $\pi_{trans} = 0.1$, which is a common occurrence in real-world networks, the clustering performance remains adequate for highly overlapping groups such as when $r = 0.8$.

3.8.10 Sensitivity Analysis

Here, we analyze the sensitivity of our estimation procedure to the choice of prior hyperparameters for the blending coefficient λ and the concentration parameters γ , α_0 , $\alpha + \kappa$. To this end, we sampled 50 networks from the time-homogeneous simulation and 50 networks from the time-inhomogeneous simulation presented in Section 3.5 of the main text. We used the same networks analyzed in the original simulation study so that the results are comparable. As in the original simulation study, we estimated the HDP-LPCM with $L = 10$ using a Markov chain consisting of 50,000 samples of which 35,000 were used for inference. On each network, we evaluated the clustering performance using the average and global ARI and VI.

For each simulation, we set the hyperparameters in the following way. For the blending coefficient, we used a truncated normal prior with a larger variance and varying mean. Specifically, we drew $\mu_\lambda \sim \text{Uniform}(0.5, 1)$ and fixed $\sigma_\lambda^2 = 1$. Also, we initialized λ to the value of μ_λ . For the concentration parameters, we explored exponential distribution priors with heavier tails than those used in the main text. Specifically, we fixed $a_\gamma = 1$, $a_{\alpha_0} = 1$, and $a_{\alpha+\kappa} = 1$, and drew $b_\gamma \sim \text{Uniform}(0.001, 1)$, $b_{\alpha_0} \sim \text{Uniform}(0.001, 1)$, and $b_{\alpha+\kappa} \sim \text{Uniform}(0.001, 1)$.

Table 3.2 contains the clustering results for the sensitivity analysis. Our estimation procedure continues to perform well. Indeed, the clustering results of the sensitivity study closely match the simulation results in the main text. Next, we display the group number’s posterior distributions for the time-homogeneous and time-inhomogeneous simulations in Figure 3.15 and Figure 3.16, respectively. We also display the estimated number of groups in Figure 3.17. The posteriors and estimated group counts closely match the results from the original simulation study. Overall, we conclude that the estimation method is not very sensitive to the choice of hyperparameters.

Simulation	Average ARI	Global ARI	Average VI	Global VI
Homogeneous (Section 3.5.3)	0.900 (0.031)	0.900 (0.031)	0.285 (0.076)	0.329 (0.085)
Inhomogeneous (Section 3.5.4)	0.925 (0.021)	0.928 (0.066)	0.205 (0.048)	0.258 (0.131)
Homogeneous (Sensitivity)	0.895 (0.030)	0.897 (0.030)	0.293 (0.075)	0.341 (0.080)
Inhomogeneous (Sensitivity)	0.930 (0.021)	0.934 (0.035)	0.200 (0.051)	0.254 (0.072)

Table 3.2: Simulation results testing prior hyperparameter sensitivity. The median values of the average and global ARI/VI are reported. Standard deviations are displayed in parentheses.

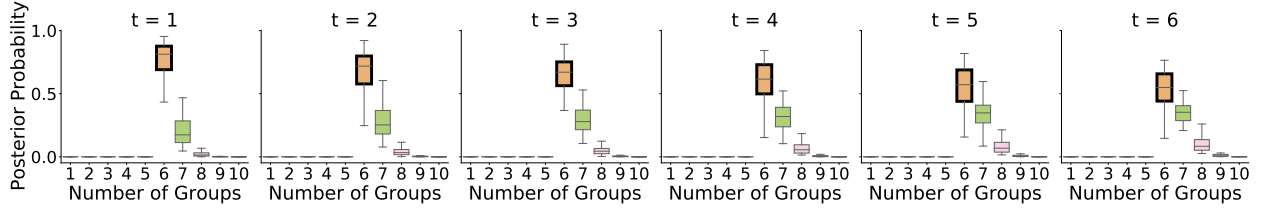


Figure 3.15: Boxplots showing the posterior distribution of $|G_t|$ over the 50 time-homogeneous simulations during the sensitivity analysis with bolded boxes indicating the true number of groups.

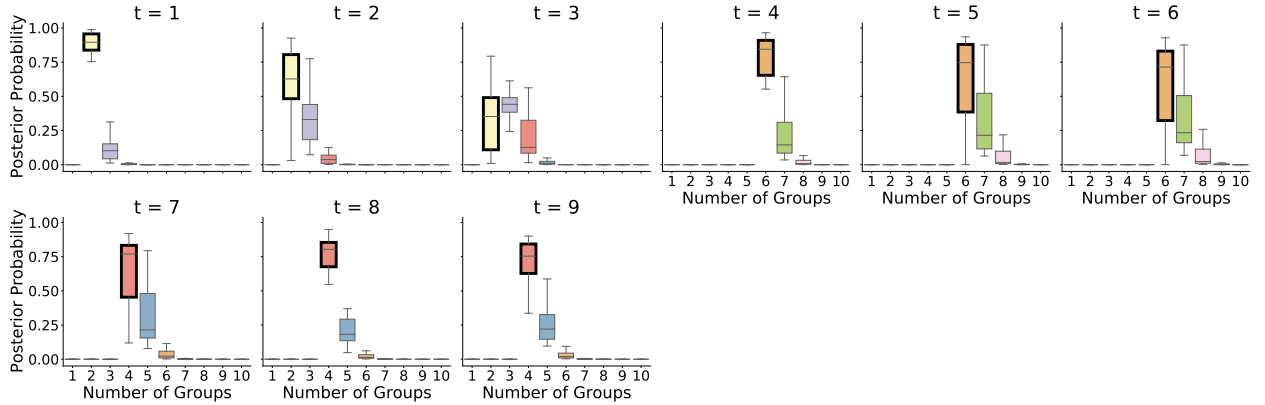


Figure 3.16: Boxplots showing the posterior distribution of $|G_t|$ over the 50 time-inhomogeneous simulations during the sensitivity analysis. Bolded boxes indicate the true number of communities.

3.8.11 Sampson’s Monastery

In this section, we apply the HDP-LPCM to dynamic networks of social relations between 18 monks in an isolated New England monastery (Sampson, 1969). The networks consists of ‘liking’ relations among the

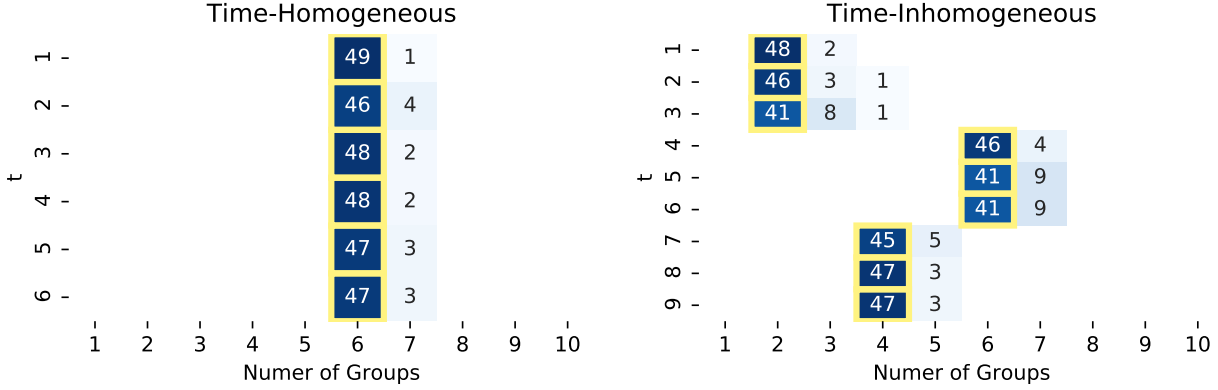


Figure 3.17: The number of groups estimated at each time step by the HDP-LPCM on the time-homogeneous (left) and time-inhomogeneous (right) simulations during the sensitivity analysis. The yellow cells contain the true number of communities.

monks, as measured by Sampson for his doctoral dissertation. Specifically, as a resident at the monastery, Sampson asked each monk to name up to three other monks that he liked most on three different occasions during his study. Based on these measurements, Sampson divided the monks into three tight factions: the Loyal Opposition (7 members), the Outcasts (4 members), and the Young Turks (7 members). During his stay, the existence of three factions was confirmed when a ‘crisis in the cloister’ occurred, resulting in the expulsion of the leaders of the Young Turks (John Bosco and Gregory) and then the voluntary departure of several others. For this reason, Sampson’s labels are often used as ground truth in the network literature.

We analyze the undirected binary version of Sampson’s networks. In this case, an edge between monks i and j exists at time t if monk i nominated monk j (or vice versa) as a friend during time t . The result is $T = 3$ undirected networks with $n = 18$ actors. Since this dataset is commonly analyzed using latent position and latent cluster models, we will compare our results to findings of previous analyses to critique our model’s performance. In addition, we extend past analyses by providing a measure of stability for the network’s group structure. In particular, we aim to answer the question of whether the three factions were present over all observational periods or split-off during a particular time point.

We fit the HDP-LPCM using 15,000 iterations for tuning, 20,000 iterations for burn-in, which left a final chain of length 165,000 iterations for inference. The trace plots of the unnormalized log-posterior, the intercept β_0 and the blending coefficient λ are displayed in Figure 3.18. We also provide ACF plots and a kernel density estimate of the marginal posterior distributions. Visual inspection of the trace plots indicates that the algorithm has converged.

Our analysis corroborates many of the previous findings about the group structure of Sampson’s monastery. The posterior mean of the blending coefficient λ is 0.869, which confirms Sampson’s assertion that a strong group effect drives the dynamics of the monk’s social network. Furthermore, previous static and dynamic analysis widely agree that a 3 or 4 component model adequately describes the network (Handcock et al., 2007; Krivitsky et al., 2009; Ryan et al., 2017). The posterior distribution over the number of groups at each time step G_t is displayed in Figures 3.19(a) - 3.19(c). These distributions confirm that 3 groups is the most probable number of groups at each point in time.

A distinctive advantage of our approach compared to other latent space models is the ability to quantify the stability of the factions within the monastery. To do this, we follow the decision theoretic procedure

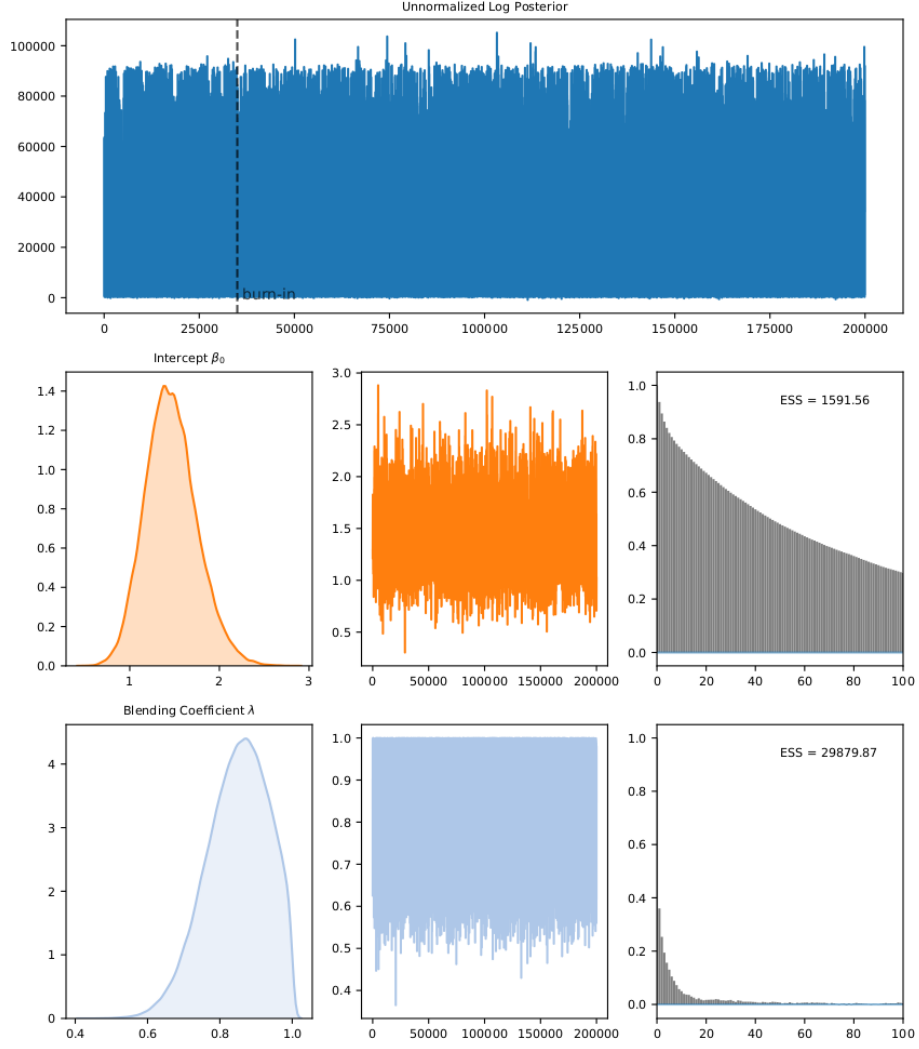


Figure 3.18: Various diagnostic plots for the MCMC algorithm used to analyze Sampson's monastery network. Trace plot of the unnormalized posterior value of each iteration of the MCMC chain (first row). Kernel density estimate of the marginal posterior, trace plot, and ACF plot for β_0 (second row), and λ (third row). The effective sample size (ESS) of the β_0 and λ chains are displayed in the upper right corners of the ACF plots.

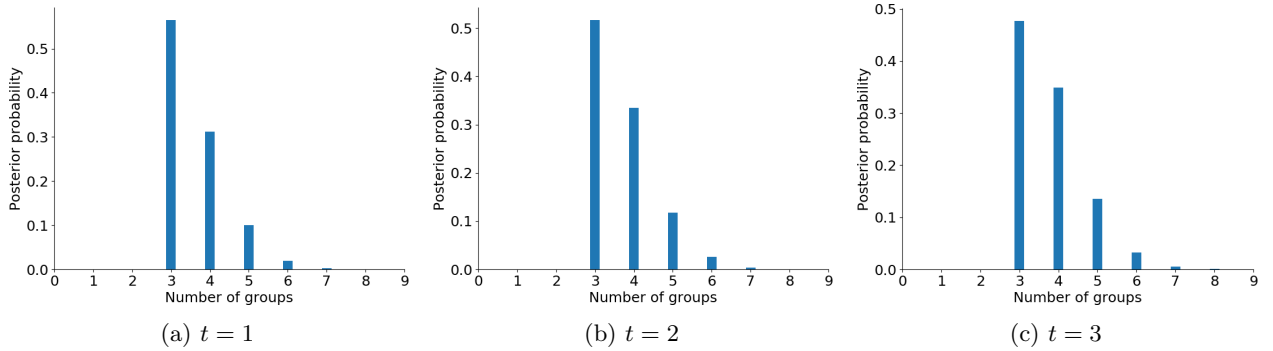


Figure 3.19: (a) - (c) Posterior probabilities over the number of groups for Sampson's monastery network at times $t = 1, 2, 3$. For each time step the posterior places the most mass on $G_t = 3$ groups. This indicates that the monastery has a static group structure.

described in Section 3.4.2 to select a representative partition that summarizes the posterior distribution. This partition’s associated alluvial diagram (Figure 3.20) and latent space (Figure 3.21) both indicate that there is a stable three group structure during all time points. Furthermore, we observe that the monk’s group memberships do not change over time. As further confirmation of our method, we note that the HDP-LPCM assigns all monks to the same communities as Sampson’s ground truth labels. Overall, our method is not only consistent with previous latent space analysis of Sampson’s monastery network, but also expands the literature through its indication that the group structure is static over the observed time period.

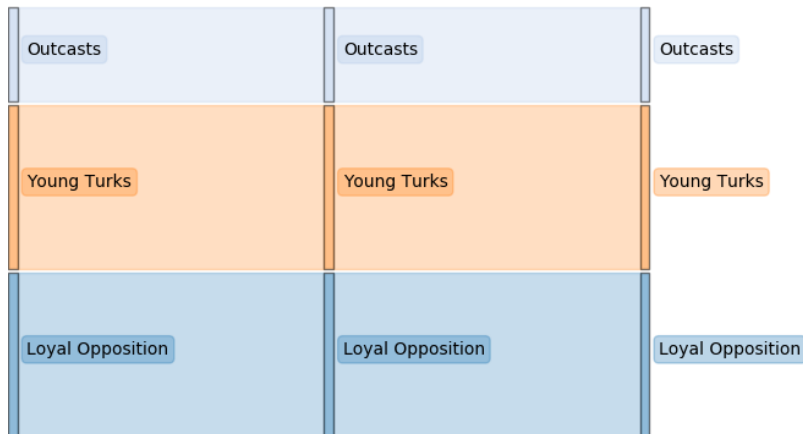


Figure 3.20: Alluvial diagram showing the dynamics of the group memberships estimated by our model on Sampson’s monastery network. Each line represents the flux of nodes going from one group to the next at time t to $t + 1$. The thickness of the lines is proportional to the number of nodes and the total height represents all nodes. In the case of Sampson’s monastery, the group memberships of each monk is static over all time periods.

3.8.12 Additional Figures

In this section, we include the remaining figures for the two datasets analyzed in the real data applications of the main text (Section 3.6). This includes the trace plots for the MCMC algorithm as well as additional latent space visualizations.

The trace plots of the unnormalized log-posterior, the intercept β_0 and the blending coefficient λ for the HDP-LPCM fit to the international military alliances network are displayed in Figure 3.22. The latent spaces for 1955-1959, 1965-1969, 1970-1974, and 1975-1979 are displayed in Figures 3.23, 3.24, 3.25, 3.26 respectively.

The trace plots of the unnormalized log-posterior, the intercept β_0 and the blending coefficient λ for the HDP-LPCM fit to the Game of Thrones networks are displayed in Figure 3.27. The latent spaces for seasons 3 and 4 are displayed in Figures 3.28 and 3.29 respectively.

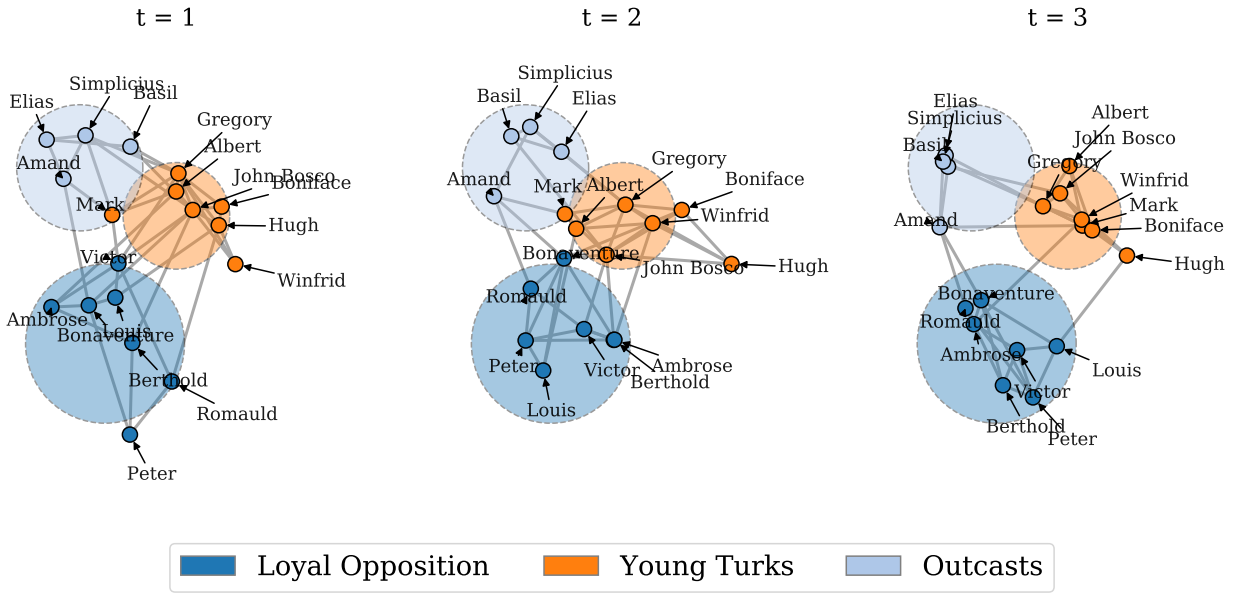


Figure 3.21: Estimated latent space for Sampson's monastery network. The three groups inferred by the model are labeled according to the corresponding faction in Sampson's ground truth labels (Loyal Opposition, Young Turks, Outcasts). The group shapes σ_g are displayed as two-standard deviation ellipses centered around the group means μ_g . The names of each monk are annotated. The undirected edges are also displayed.

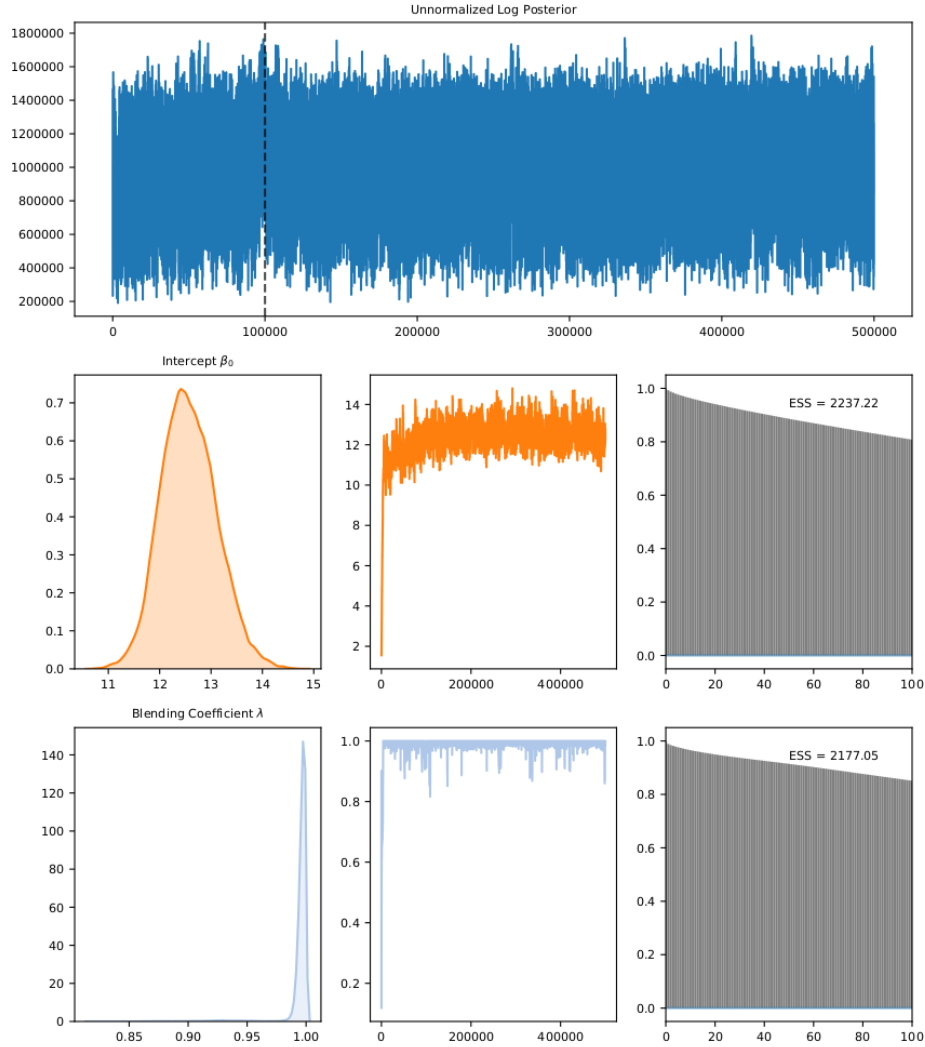


Figure 3.22: Various diagnostic plots for the MCMC algorithm used to analyze international military alliances network. Trace plot of the unnormalized posterior value of each iteration of the MCMC chain (first row). Kernel density estimate of the marginal posterior, trace plot, and ACF plot for β_0 (second row), and λ (third row). The effective sample size (ESS) of the β_0 and λ chains are displayed in the upper right corners of the ACF plots.

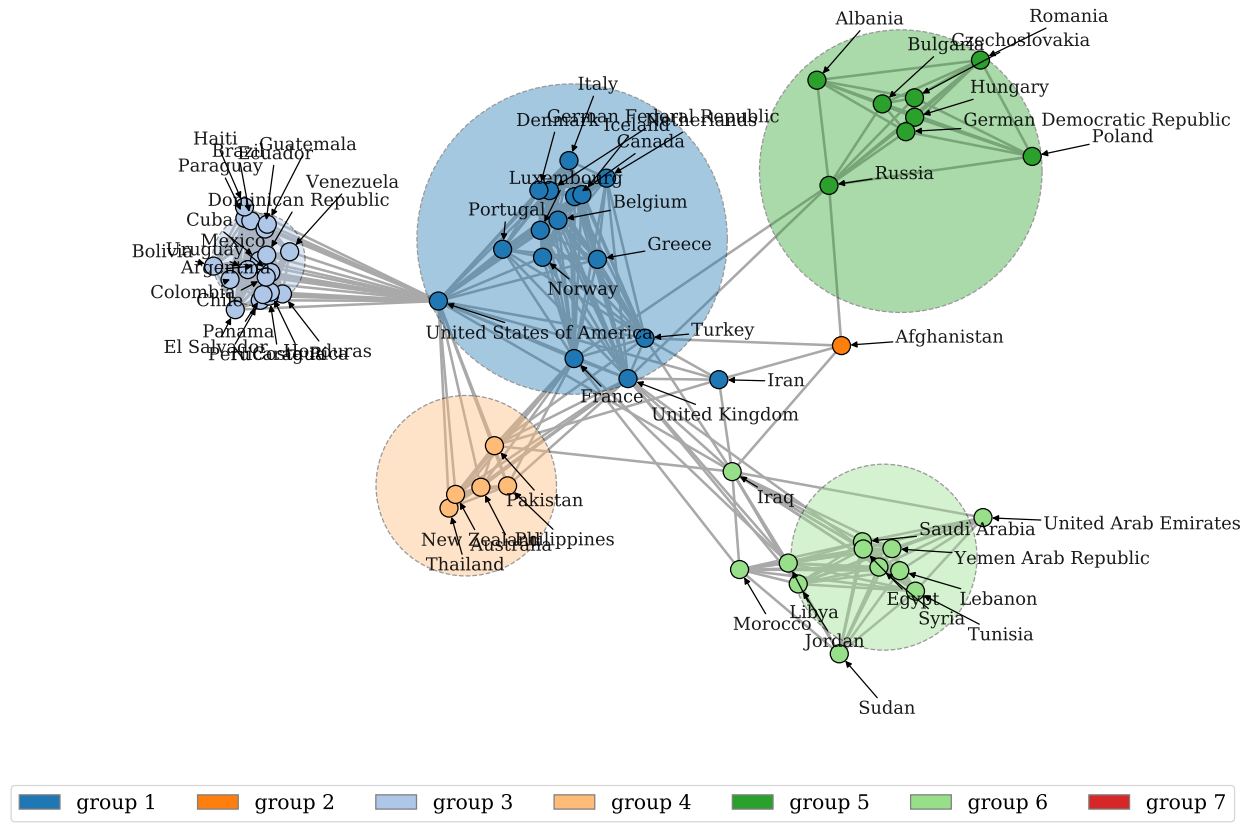


Figure 3.23: Latent space of international military alliances for the years 1955 - 1959. The group shapes σ_g are displayed as two-standard deviation ellipses centered around the group means μ_g . The names of each nation are annotated. The undirected edges are also displayed. For clarity, all unconnected nodes and group 2's two-standard deviation ellipses are removed.

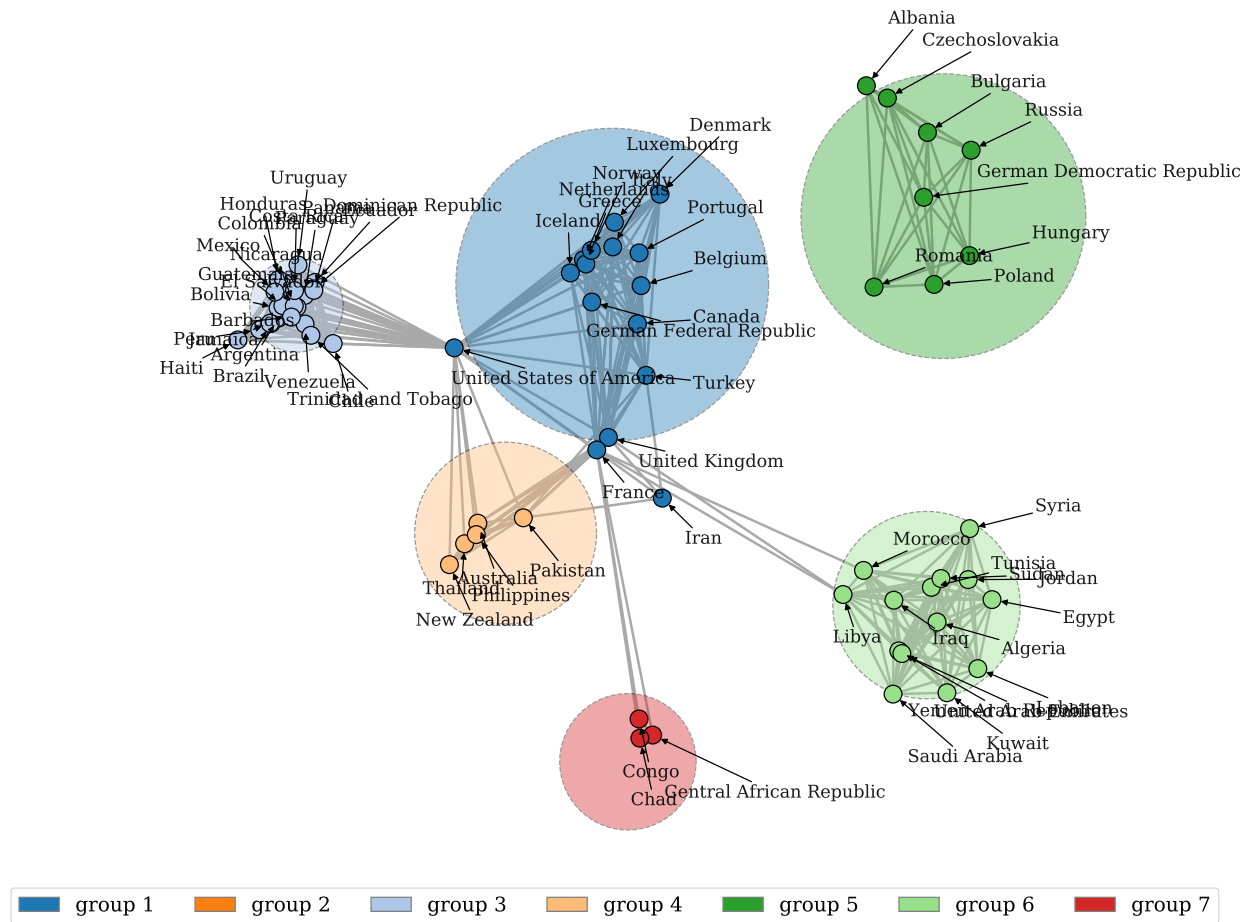


Figure 3.24: Latent space of international military alliances for the years 1965 - 1969. The group shapes σ_g are displayed as two-standard deviation ellipses centered around the group means μ_g . The names of each nation are annotated. The undirected edges are also displayed. For clarity, all unconnected nodes and group 2's two-standard deviation ellipses are removed.

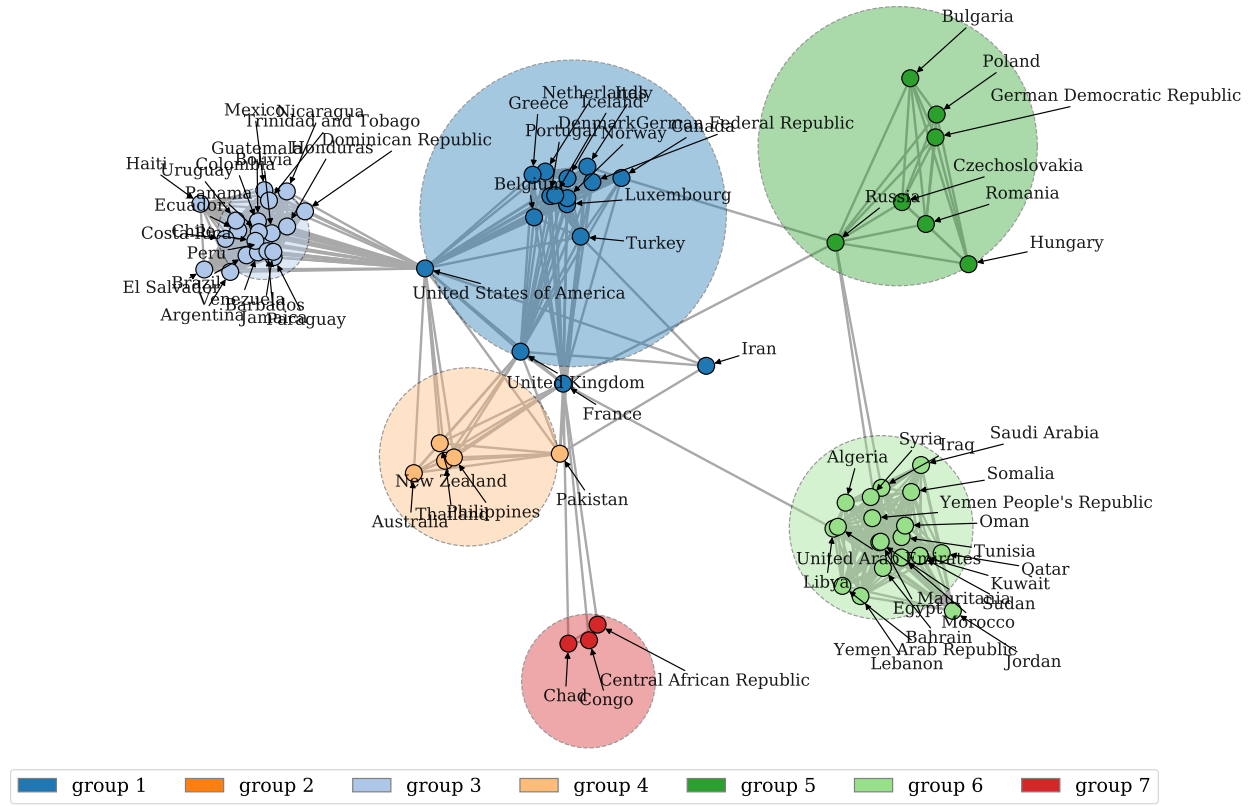


Figure 3.25: Latent space of international military alliances for the years 1970 - 1974. The group shapes σ_g are displayed as two-standard deviation ellipses centered around the group means μ_g . The names of each nation are annotated. The undirected edges are also displayed. For clarity, all unconnected nodes and group 2's two-standard deviation ellipses are removed.

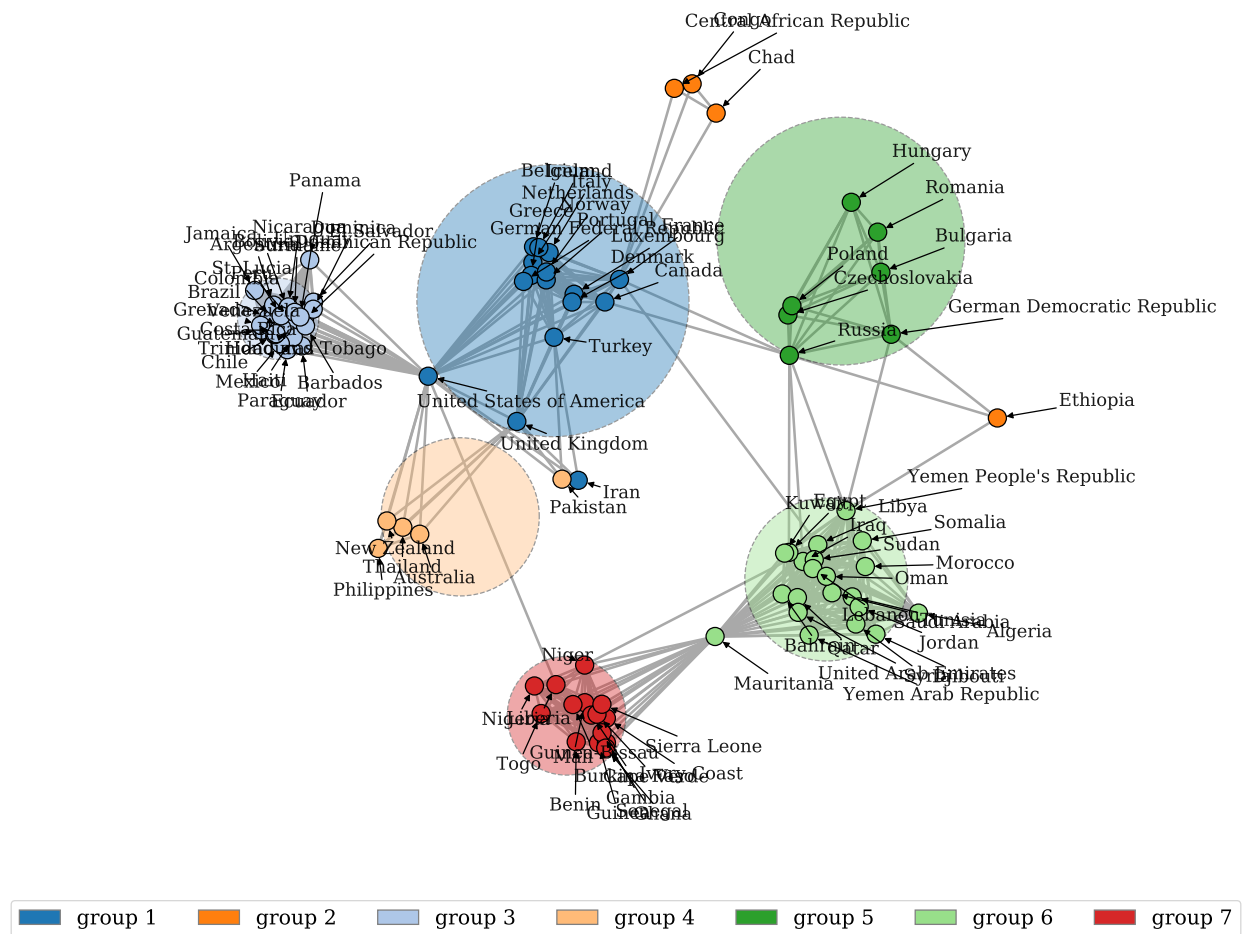


Figure 3.26: Latent space of international military alliances for the years 1975 - 1979. The group shapes σ_g are displayed as two-standard deviation ellipses centered around the group means μ_g . The names of each nation are annotated. The undirected edges are also displayed. For clarity, all unconnected nodes and group 2's two-standard deviation ellipses are removed.

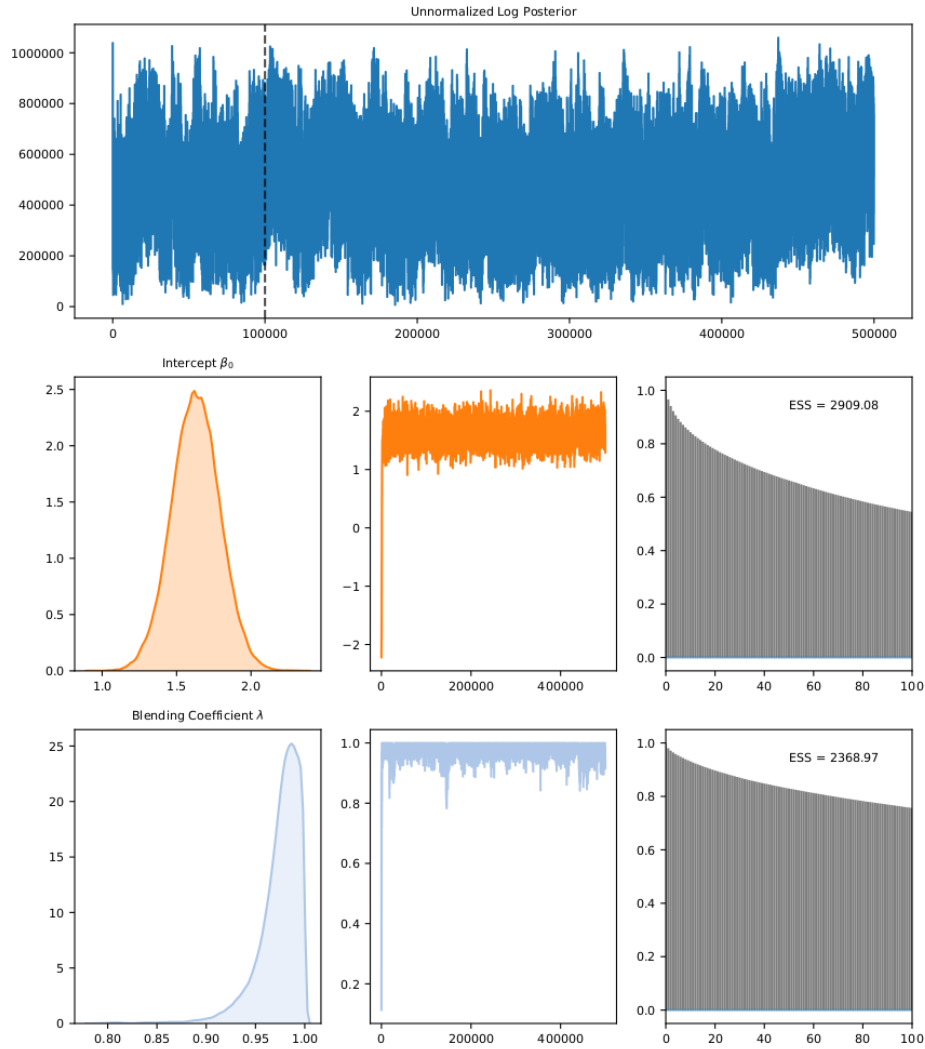


Figure 3.27: Various diagnostic plots for the MCMC algorithm used to analyze Game of Thrones network. Trace plot of the unnormalized posterior value of each iteration of the MCMC chain (first row). Kernel density estimate of the marginal posterior, trace plot, and ACF plot for β_0 (second row), and λ (third row). The effective sample size (ESS) of the β_0 and λ chains are displayed in the upper right corners of the ACF plots.

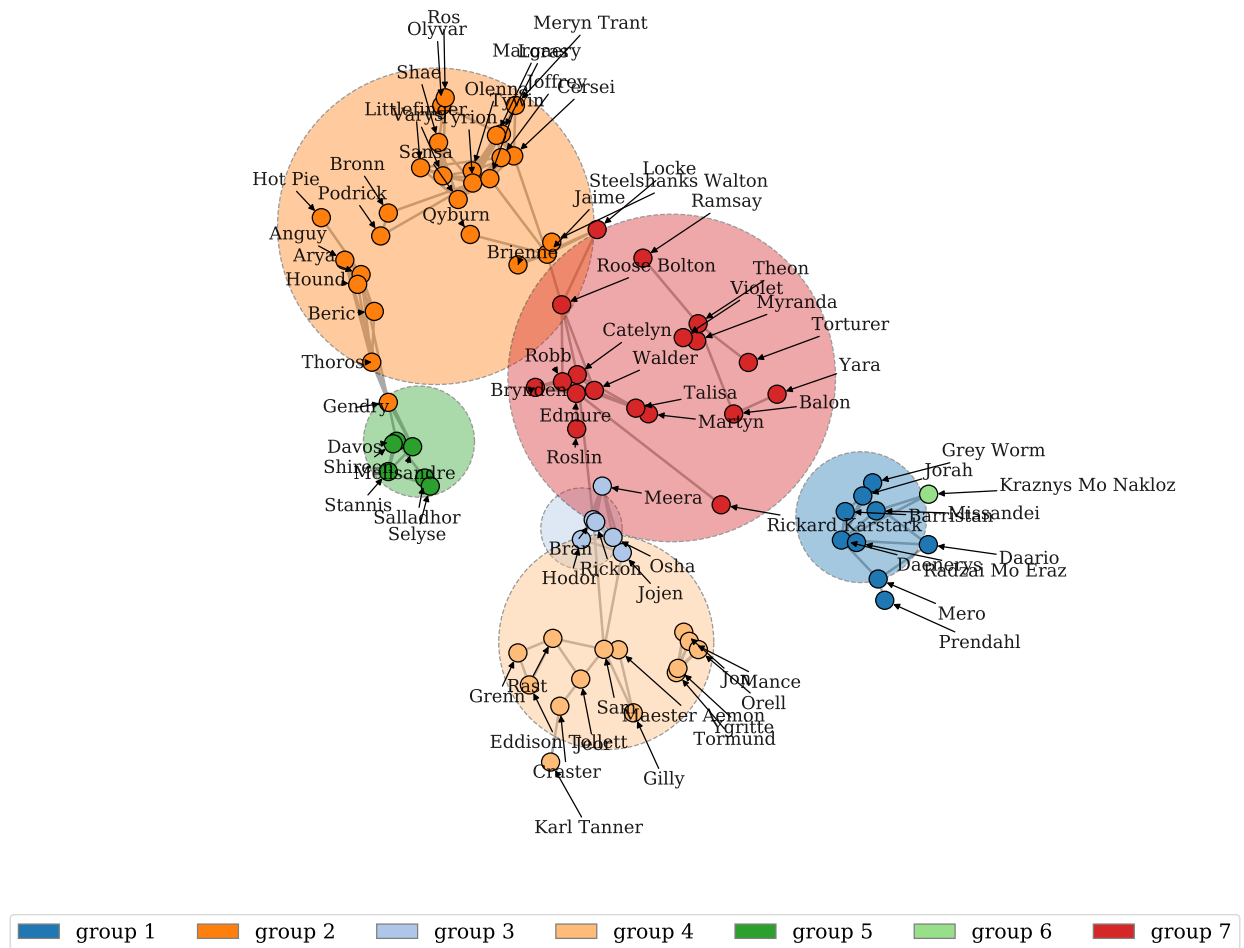


Figure 3.28: Latent space of the Game of Thrones character interaction network for season 3. The group shapes σ_g are displayed as two-standard deviation ellipses centered around the group means μ_g . The names of each character are annotated. The undirected edges are also displayed. For clarity, all unconnected nodes and group 6's two-standard deviation ellipses are removed.

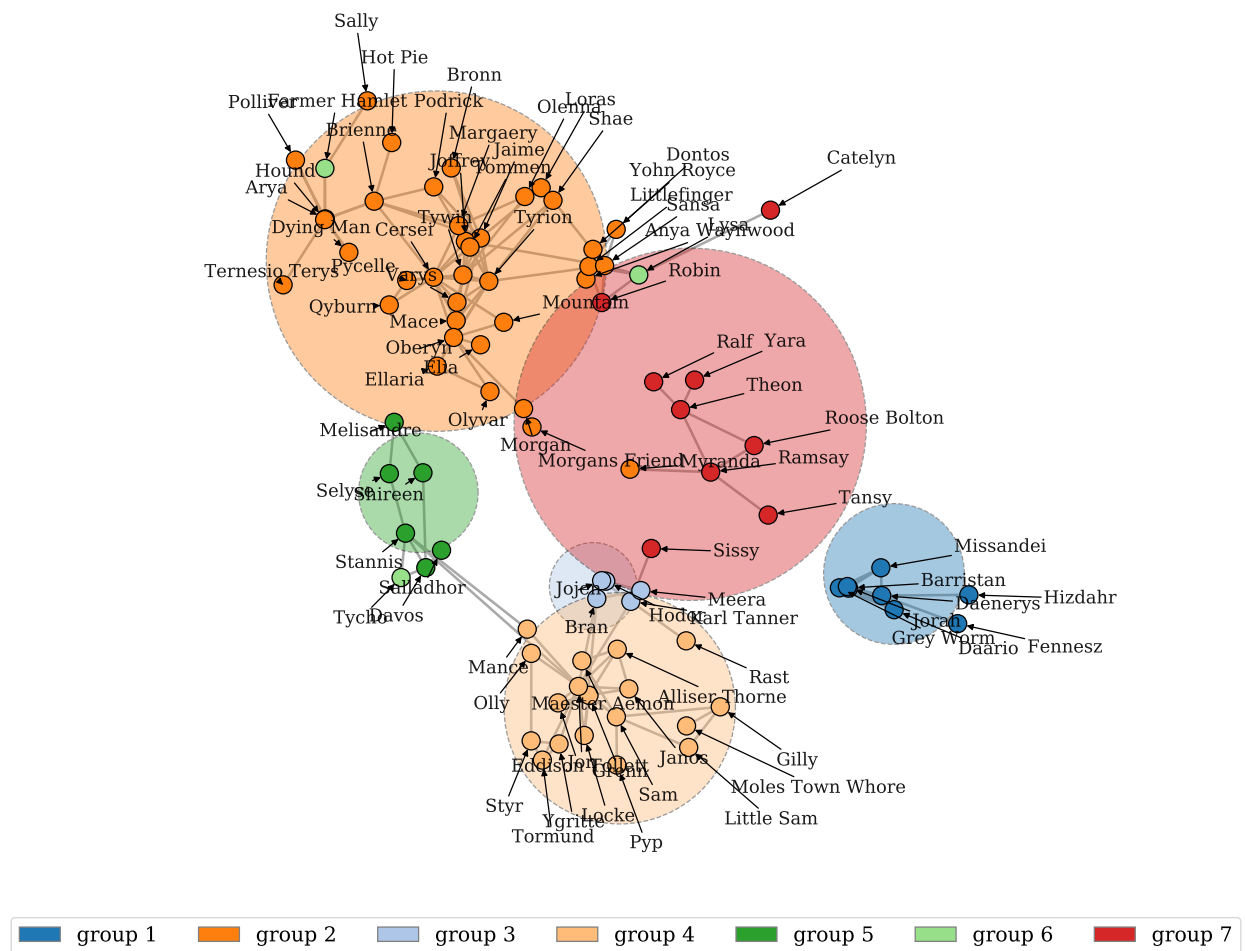


Figure 3.29: Latent space of the Game of Thrones character interaction network for season 4. The group shapes σ_g are displayed as two-standard deviation ellipses centered around the group means μ_g . The names of each character are annotated. The undirected edges are also displayed. For clarity, all unconnected nodes and group 6's two-standard deviation ellipses are removed.

Chapter 4

An Eigenmodel for Dynamic Multilayer Networks

4.1 Introduction

Dynamic multilayer networks are a prevalent form of relational data with applications in epidemiology, sociology, biology, and other fields (Boccaletti et al., 2014).¹ Unlike static single-layer networks, which are limited to recording one dyadic relation among a set of actors at a single point in time, dynamic multilayer networks contain several types of dyadic relations, called layers, observed over a sequence of times. For instance, social networks contain several types of social relationships jointly evolving over time: friendship, vicinity, coworker-ship, partnership, and others. Also, international relations unfold through daily political events involving two countries, e.g., offering aid, verbally condemning, or participating in military conflict (P. D. Hoff, 2015). Lastly, the spread of information on social media occurs on a dynamic multilayer network, e.g., hourly interactions among Twitter users such as liking, replying to, and re-tweeting each other’s content (Domenico et al., 2013). Proper statistical modeling of dynamic multilayer networks is essential for an accurate understanding of these complex systems.

The statistical challenge in modeling multiple co-evolving networks is maintaining a concise representation while also adequately describing important network characteristics. These characteristics include the dyadic dependencies in each individual static relation, such as degree heterogeneity and transitivity, the autocorrelation of the individual dyadic time series, and the common structures shared among the various relations. We provide an example of such network characteristics in Figure 4.1, which displays the monthly time series of four dyadic relations between Iraq and other countries from 2009 to 2017. A complete description of the data can be found in Section 4.5. Within a layer (e.g., verbal cooperation), the individual time series (rows) are correlated with each other while also exhibiting strong autocorrelation. Furthermore, the four relations share a clear homogeneous structure, which is made especially evident after the abrupt change in all dyadic time series in late 2014 due to an American-led intervention in Iraq. We explore this event in more detail in Section 4.5. A statistical network model should decompose these dependencies in an interpretable way.

To date, the statistics literature contains an expansive collection of network models designed to capture

¹This chapter is based on research published as: Loyal, J. D., and Chen, Y. (2021). An eigenmodel for dynamic multilayer networks. *arXiv preprint arXiv:2103.12831*

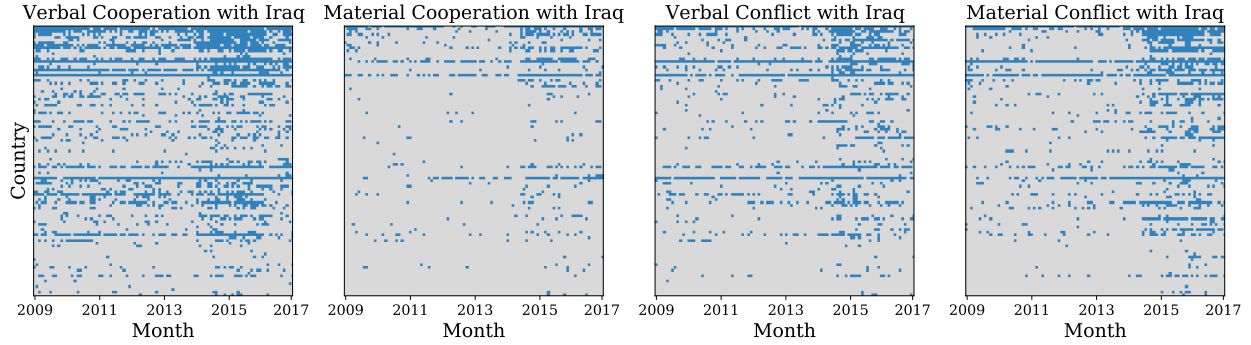


Figure 4.1: Monthly cooperation and conflict relations between Iraq and other countries from 2009 to 2017. A blue (gray) square indicates a relation occurred (did not occur) between Iraq and that nation during that month.

specific network properties. See Goldenberg et al. (2010) and Loyal and Chen (2020) for a comprehensive review. An important class of network models is latent space models (LSMs) proposed in P. D. Hoff et al. (2002). The key idea behind LSMs is that each actor is assigned a vector in some low-dimensional latent space whose pairwise distances under a specified similarity measure determine the network’s dyad-wise connection probabilities. The LSM interprets these latent features as an actor’s unmeasured characteristics such that actors that are close in the latent space are more likely to form a connection. This interpretation naturally explains the high levels of homophily (assortativity) and transitivity in real-world networks. A series of works expanded the network characteristics captured by LSMs (Handcock et al., 2007; P. D. Hoff, 2005, 2008; Krivitsky et al., 2009; Z. Ma et al., 2020), such as community structure, degree heterogeneity, heterophily (disassortativity), etc. Furthermore, researchers have adopted the LSM formulation to model both dynamic networks (Durante & Dunson, 2014; He & Hoff, 2019; Sarkar & Moore, 2005; Sewell & Chen, 2015b) and static multilayer networks (D’Angelo et al., 2019; Gollini & Murphy, 2016; Salter-Townshend & McCormick, 2017; L. Wang et al., 2019; Zhang et al., 2020).

Currently, the statistical methodology for modeling dynamic multilayer networks is limited. Snijders et al. (2013) introduced a stochastic actor-oriented model which represents the networks as co-evolving continuous-time Markov processes. In addition, P. D. Hoff (2015) introduced a multilinear tensor regression framework where dynamic multilayer networks are modeled through tensor autoregression. To our knowledge, the only existing LSM for dynamic multilayer networks is the Bayesian nonparametric model proposed in Durante et al. (2017). Although highly flexible, this model lacks interpretability due to strong non-identifiable issues. Furthermore, this model’s applications are limited to small networks with only a few dozen nodes and time points due to the model’s high computational complexity. Currently, the LSM literature lacks models that decompose the complexity of dynamic multilayer networks into interpretable components and scale to the large networks commonly analyzed in practice.

To address these needs, we develop a new Bayesian dynamic bilinear latent space model that is flexible, interpretable, and computationally efficient. Our approach identifies a common time-varying structure shared by all layers while also accounting for layer-wise variation. Intuitively, our model posits that actors have intrinsic traits that influence how they connect in each layer. Specifically, we identify a common structure in which we represent each node by a single latent vector shared across layers. Also, we introduce node-specific additive random effects (or socialities) to adjust for heavy-tailed degree distributions (Rastelli et al., 2016). The model accounts for layer-wise heterogeneity in two ways. First, the layers assign different

amounts of homophily (heterophily) to each latent trait. Second, to capture the dependence of an actor’s degree on relation type, we allow the additive random effects to vary by layer. Lastly, we propagate the latent variables through time via a discrete Markov process. These correlated changes capture the network’s structural evolution and temporal autocorrelation.

To estimate our model, we derive a variational inference algorithm (Blei et al., 2017; Wainwright & Jordan, 2008) that scales to networks much larger than those analyzed by previous approaches. We base our inference on a structured mean-field approximation to the posterior. Our approximation improves upon previous variational approximations found in the dynamic latent space literature (Sewell & Chen, 2017) by retaining the latent variable’s temporal dependencies. Furthermore, we derive a coordinate ascent variational inference algorithm that consists of closed-form updates. Our work leads to a novel approach to fitting dynamic latent space models using techniques from the linear Gaussian state space model (GSSM) literature.

The structure of this chapter is as follows. In Section 4.2, we present our Bayesian parametric model for dynamic multilayer networks and discuss identifiability issues. Section 4.3 outlines our structured mean-field approximation and the coordinate ascent variational inference algorithm used for estimation. Section 4.4 demonstrates the accuracy and scalability of our inference algorithm on simulated networks of various sizes. In Section 4.5, we apply our model to two real-world networks taken from international relations and epidemiology. Section 4.6 concludes with a discussion of various model extensions and future research directions. Finally, the supplementary material in Section 4.7 contains proofs, in-depth derivations of the variational inference algorithm, implementation details, and additional results and figures.

Notation. We write $[N] = \{1, \dots, N\}$. We use the notation $B_{1:L}$ to refer to the sequence $(B_1, B_2, \dots, B_L)$ where B_ℓ is any indexed object. Also, for objects with a double index, we use the notation $C_{1:M, 1:N}$ to refer to the collection $(C_{mn})_{(m,n) \in [M] \times [N]}$. We use $\mathbb{1}_{\{x=a\}}$ to denote the Boolean indicator function, which evaluates to 1 when $x = a$ and 0 otherwise. We denote an n -dimensional vector of ones by $\mathbf{1}_n$ and the $n \times n$ identity matrix by I_n . Furthermore, given a vector $\mathbf{v} \in \mathbb{R}^d$, we use $\text{diag}(\mathbf{v})$ to indicate a $d \times d$ diagonal matrix with the elements of $\mathbf{v}$ on the diagonal. Lastly, we use $I_{p,q} = \text{diag}(1, \dots, 1, -1, \dots, -1)$ to denote a diagonal matrix with p ones followed by q negative ones on the diagonal.

4.2 An Eigenmodel for Dynamic Multilayer Networks

In this section, we develop our Bayesian model for dynamic multilayer networks. To begin, we formally introduce dynamic multilayer network data. Dynamic multilayer networks consist of K relations measured over T time points between the same set of n nodes (or actors). We collect these relations in binary adjacency matrices $\mathbf{Y}_t^k \in \{0, 1\}^{n \times n}$ for $1 \leq k \leq K$ and $1 \leq t \leq T$. The entries Y_{ijt}^k indicate the presence ($Y_{ijt}^k = 1$) or absence ($Y_{ijt}^k = 0$) of an edge between actors i and j in layer k at time t . This chapter only considers undirected networks without self-loops so that $\mathbf{Y}_t^k$ is a symmetric matrix. We discuss extensions of our model to weighted and directed networks in Section 4.6.

4.2.1 The Model

Here, we propose our new eigenmodel for dynamic multilayer networks with the goal of capture the correlations between different dyads within a network, the dyads’ autocorrelation over time, and the dependence

between layers. Specifically, we assume that the dyads are independent Bernoulli random variables conditioned on the latent parameters:

$$\mathbb{P}(\mathbf{Y}_{1:T}^1, \dots, \mathbf{Y}_{1:T}^K \mid \boldsymbol{\delta}_{1:K, 1:T}, \boldsymbol{\Lambda}_{1:K}, \mathcal{X}_{1:T}) = \prod_{k=1}^K \prod_{t=1}^T \prod_{j < i} \mathbb{P}(Y_{ijt}^k \mid \delta_{k,t}^i, \delta_{k,t}^j, \Lambda_k, \mathbf{X}_t^i, \mathbf{X}_t^j),$$

where

$$\text{logit} \left[\mathbb{P}(Y_{ijt}^k = 1 \mid \delta_{k,t}^i, \delta_{k,t}^j, \Lambda_k, \mathbf{X}_t^i, \mathbf{X}_t^j) \right] = \Theta_{ijt}^k = \delta_{k,t}^i + \delta_{k,t}^j + \mathbf{X}_t^{i\top} \Lambda_k \mathbf{X}_t^j. \quad (4.1)$$

In Equation (4.1), layer k 's log-odds matrix at time t , $\boldsymbol{\Theta}_t^k \in \mathbb{R}^{n \times n}$ with elements Θ_{ijt}^k , contains two latent random effects that induce essential unconditional dependencies in the dynamic multilayer network's dyads. We defer specification of their distributions until the next section. First, the sociality effects $\boldsymbol{\delta}_{k,t} = (\delta_{k,t}^1, \dots, \delta_{k,t}^n)^\top \in \mathbb{R}^n$ model degree heterogeneity and node-level autocorrelation. Second, the time-varying latent positions $\mathcal{X}_t = (\mathbf{X}_t^1, \dots, \mathbf{X}_t^n)^\top \in \mathbb{R}^{n \times d}$, where d is the latent space's dimension, induce clusterability (P. D. Hoff, 2008) in the networks and dyadic autocorrelation. Finally, the homophily coefficients $\Lambda_k = \text{diag}(\boldsymbol{\lambda}_k) \in \mathbb{R}^{d \times d}$ are diagonal matrices that quantify each relation's level of homophily along a latent dimension. For each λ_{kh} ($1 \leq h \leq d$), positive values ($\lambda_{kh} > 0$) indicate homophily along the h th latent dimension in layer k , negative values ($\lambda_{kh} < 0$) indicate heterophily along the h th latent dimension in layer k , and a zero value ($\lambda_{kh} = 0$) indicates the h th latent dimension does not contribute to the connection probability in layer k . Furthermore, the model captures common structures among the layers by sharing a common set of latent trajectories.

In matrix form, the log-odds matrices are

$$\boldsymbol{\Theta}_t^k = \boldsymbol{\delta}_{k,t} \mathbf{1}_n^\top + \mathbf{1}_n \boldsymbol{\delta}_{k,t}^\top + \mathcal{X}_t \Lambda_k \mathcal{X}_t^\top.$$

To ensure identifiability of the model parameters, we require both a centered latent space, that is $J_n \mathcal{X}_t = \mathcal{X}_t$ where $J_n = I_n - (1/n) \mathbf{1}_n \mathbf{1}_n^\top$, and $\Lambda_r = I_{p,q}$, where $p + q = d$, for some reference layer $r \in \{1, \dots, K\}$. In an applied setting, one could take a particular interesting layer as the reference layer. Otherwise, as in this work, we select $r = 1$ as the reference. As we elaborate in Section 4.2.5, we use these conditions to identify the socialities $\boldsymbol{\delta}_{k,t}$ and to identify $\mathcal{X}_t$ up to a common linear transformation of its rows. At the same time, we show that the bilinear term, $\mathcal{X}_t \Lambda_k \mathcal{X}_t^\top$, is directly identifiable.

Overall, the proposed model's parameterization reduces the dimensionality of dynamic multi-relational data. The model contains $nTK + nTd + Kd$ parameters, which, for typical values of d , is much less than the $KTn(n-1)/2$ dyads that originally summarized the dynamic multilayer network. In this work, we fix $d = 2$, which allows us to use the latent space for network visualization. For a discussion on data-driven choices of d , see Section 4.6. Next, we elaborate on the interpretation of the model's parameters and our inclusion of temporal correlation in the random effects.

4.2.2 Layer-Specific Social Trajectories

An actor's sociality, $\delta_{k,t}^i$, represents their global popularity in layer k at time t . In particular, holding all other parameters fixed, the larger an actor's sociality $\delta_{k,t}^i$, the more likely they are to connect with other nodes in the k th layer at time t regardless of their position in the latent space. Formally, $\delta_{k,t}^i$ is the conditional log-odds ratio of actor i forming a connection with another actor in layer k at time t compared to an actor with the same latent position as actor i but with $\delta_{k,t}^i = 0$. Hub nodes are an example of nodes with a high

sociality, while isolated nodes have a low sociality. An actor’s sociality can differ between layers. We find this flexibility necessary to model real-world multilayer relations. For example, in the international relations network presented in the introduction, a peaceful nation might participate in many cooperative relations while rarely engaging in conflict relations.

The i th actor’s social trajectory in layer k , $\delta_{k,1:T}^i$, measures their time-varying sociality in the k th layer. For example, a nation’s propensity to engage in militaristic relations might increase after a regime change. We assume that the social trajectories are independent across layers k and individuals i and propagate them through time via a shared Markov process:

$$\delta_{k,1}^i \stackrel{\text{iid}}{\sim} N(0, \tau_\delta^2), \quad \delta_{k,t}^i \sim N(\delta_{k,t-1}^i, \sigma_\delta^2), \quad t = 2, \dots, T, \quad k = 1, \dots, K,$$

where iid stands for independent and identically distributed. In the previous expression, τ_δ^2 measures the sociality effects’ initial variation over all layers. Similarly, σ_δ^2 measures the sociality effects’ variation over time. In particular, a small value of σ_δ^2 indicates that most social trajectories are flat with little dynamic variability. We place the following conjugate priors on the variance parameters: $\tau_\delta^2 \sim \Gamma^{-1}(a_{\tau_\delta^2}/2, b_{\tau_\delta^2}/2)$ and $\sigma_\delta^2 \sim \Gamma^{-1}(c_{\sigma_\delta^2}/2, d_{\sigma_\delta^2}/2)$.

4.2.3 Dynamic Latent Features Shared Between Layers

Like other latent space models, we assume that the probability of two actors forming a connection depends on their latent representations in an unobserved latent space. Specifically, we assign every actor a latent feature $\mathbf{X}_t^i \in \mathbb{R}^d$ at each time point. Relations in dynamic networks typically have strong autocorrelations wherein the dyadic relations and latent features slowly vary over time. These autocorrelations are captured by a distribution that assumes the latent positions propagate through time via a shared Markov process (Sarkar & Moore, 2005; Sewell & Chen, 2015b):

$$\mathbf{X}_1^i \stackrel{\text{iid}}{\sim} N(0, \tau^2 I_d), \quad \mathbf{X}_t^i \sim N(\mathbf{X}_{t-1}^i, \sigma^2 I_d), \quad t = 2, \dots, T.$$

Intuitively, these dynamics assume that changes in the network’s connectivity patterns are partly due to changes in the actor’s latent features. Like the social trajectories’ dynamics, the τ^2 parameter in the previous expression measures the latent space’s initial variation or size. Also, σ^2 measures the step size of each latent position’s Gaussian random-walk. We place the following conjugate priors on the variance parameters: $\tau^2 \sim \Gamma^{-1}(a_{\tau^2}/2, b_{\tau^2}/2)$ and $\sigma^2 \sim \Gamma^{-1}(c_{\sigma^2}/2, d_{\sigma^2}/2)$.

4.2.4 Layer-Specific Homophily Levels

The proposed model posits that the multilayer networks are correlated because they share a single set of latent positions $\mathcal{X}_t$ among all layers. Mathematically, this restriction allows the model to capture common structures across layers. The homophily coefficients $\Lambda_k = \text{diag}(\boldsymbol{\lambda}_k)$ for $\boldsymbol{\lambda}_k \in \mathbb{R}^d$ allow for variability between the layers. Intuitively, the model assumes that two relations differ because they put distinct weights on the latent features. For example, homophilic features in a friendship relation may be heterophilic in a combative relation. For interpretability, we restrict Λ_k to a diagonal matrix. We place independent multivariate Gaussian priors on the diagonal elements

$$\boldsymbol{\lambda}_k \stackrel{\text{iid}}{\sim} N(0, \sigma_\lambda^2 I_d), \quad k = 2, \dots, K.$$

For identifiability reasons, the homophily coefficients take values of ± 1 in the first layer. We enforce this reference layer constraint by re-parameterizing the reference layer's diagonal elements in terms of Bernoulli random variables

$$\lambda_{1h} = 2u_h - 1, \quad u_h \stackrel{\text{iid}}{\sim} \text{Bernoulli}(\rho), \quad h = 1, \dots, d,$$

where ρ is the prior probability of an assortative relationship along a latent dimension. Under this constraint, we interpret the other layer's homophily coefficients in comparison to the reference. For example, if $\lambda_{11} = -1$ and $\lambda_{21} = 2$, then the second layer weights dimension one twice as heavily as the reference layer while exhibiting homophily instead of heterophily.

4.2.5 Identifiability and Interpretability

Here, we present sufficient conditions for identifiability and their implications on inference. For bilinear latent space models with sociality effects, it is natural to require the matrix of latent positions to be centered and full rank (MacDonald et al., [in press](#); Zhang et al., 2020). In addition to the previous conditions, Proposition 4.1 shows that restricting the reference layer's homophily coefficients to take values ± 1 is sufficient to identify our model up to a restricted linear transformation of the latent space. However, the form of this linear transformation is difficult to interpret. As such, we provide stronger conditions that are sufficient to restrict the linear transformations to interpretable forms. We provide the proofs in Section 4.7.1.

Proposition 4.1 (Identifiability Conditions). *Suppose that two sets of parameters $\{\boldsymbol{\delta}_{1:K,1:T}, \mathcal{X}_{1:T}, \Lambda_{1:K}\}$ and $\{\tilde{\boldsymbol{\delta}}_{1:K,1:T}, \tilde{\mathcal{X}}_{1:T}, \tilde{\Lambda}_{1:K}\}$ satisfy the following conditions:*

A4.1. $J_n \mathcal{X}_t = \mathcal{X}_t$ and $J_n \tilde{\mathcal{X}}_t = \tilde{\mathcal{X}}_t$ for $t = 1, \dots, T$ where $J_n = I_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T$.

A4.2. $\text{rank}(\mathcal{X}_t) = \text{rank}(\tilde{\mathcal{X}}_t) = d$ for $t = 1, \dots, T$.

A4.3. For at least one $r \in \{1, \dots, K\}$, $\Lambda_r = I_{p,q}$ and $\tilde{\Lambda}_r = I_{p',q'}$.

Then the model is identifiable up to a linear transformation of the latent space, that is, if for all $1 \leq k \leq K$ and $1 \leq t \leq T$ we have that

$$\boldsymbol{\delta}_{k,t} \mathbf{1}_n^T + \mathbf{1}_n \boldsymbol{\delta}_{k,t}^T + \mathcal{X}_t \Lambda_k \mathcal{X}_t^T = \tilde{\boldsymbol{\delta}}_{k,t} \mathbf{1}_n^T + \mathbf{1}_n \tilde{\boldsymbol{\delta}}_{k,t}^T + \tilde{\mathcal{X}}_t \tilde{\Lambda}_k \tilde{\mathcal{X}}_t^T,$$

then for all $1 \leq k \leq K$ and $1 \leq t \leq T$ we have that

$$\tilde{\boldsymbol{\delta}}_{k,t} = \boldsymbol{\delta}_{k,t}, \quad \tilde{\mathcal{X}}_t = \mathcal{X}_t M_t, \quad \tilde{\Lambda}_k = M_t^T \Lambda_k M_t,$$

where each $M_t \in \mathbb{R}^{d \times d}$ satisfies $M_t I_{p',q'} M_t^T = I_{p,q}$ for $1 \leq t \leq T$.

Assumption A4.1 alone, which centers the latent space, is sufficient to remove any confounding between the social trajectories and the latent positions. This issue arises because the likelihood is invariant to translations in the latent space. Indeed,

$$\begin{aligned} \delta_{k,t}^i + \delta_{k,t}^j + \mathbf{X}_t^i{}^T \Lambda_k \mathbf{X}_t^j &= \delta_{k,t}^i + \delta_{k,t}^j + (\mathbf{X}_t^i - \mathbf{c} + \mathbf{c})^T \Lambda_k (\mathbf{X}_t^j - \mathbf{c} + \mathbf{c}), \\ &= \tilde{\delta}_{k,t}^i + \tilde{\delta}_{k,t}^j + \tilde{\mathbf{X}}_t^i{}^T \Lambda_k \tilde{\mathbf{X}}_t^j, \end{aligned}$$

where $\tilde{\mathbf{X}}_t^i = \mathbf{X}_t^i - \mathbf{c}$ and $\tilde{\delta}_{k,t}^i = \delta_{k,t}^i + \tilde{\mathbf{X}}_t^i{}^T \Lambda_k \mathbf{c} + \mathbf{c}^T \Lambda_k \tilde{\mathbf{X}}_t^i / 2$. Such confounding is present in previous bilinear latent space models that treat the latent positions as random effects (P. D. Hoff et al., 2002; Krivitsky et al.,

2009). Our prior specification does not directly enforce the centering constraint. Instead, we let all the parameters float because including this redundancy can speed up the variational algorithm proposed in the next section (Liu & Wu, 1999; Qi & Jaakkola, 2006; van Dyk & Meng, 2001). However, when summarizing the model, we want to identify the social and latent trajectories so that the sociality effects are no longer confounded with the bilinear term. Therefore, after estimation, we perform posterior inference on $\tilde{\mathbf{X}}_t^i$ and $\tilde{\delta}_{k,t}^i$ with $\mathbf{c} = (1/n) \sum_{i=1}^n \mathbf{X}_t^i$. See Section 4.3.5 for details.

Under Proposition 4.1, the latent space is identifiable up to a restricted set of linear transformations, M_t , that are difficult to interpret. For this reason, we provide the following proposition, which reduces the set of linear transformations to a well-studied group of transformations when the latent space dimension $d \leq 3$, which is often the case in practice.

Proposition 4.2. *Consider the same setup as in Proposition 4.1 and assume that conditions A4.1–A4.3 are satisfied. If $1 \leq d \leq 3$, then $I_{p,q} = I_{p',q'}$ so that each M_t is in the indefinite orthogonal group, i.e., $M_t I_{p,q} M_t^T = I_{p,q}$ for $1 \leq t \leq T$.*

Invariance under the indefinite orthogonal group is common in LSMs that allow a disassortative latent space (Rubin-Delanchy et al., 2017). Furthermore, this group reduces to the orthogonal group, another source of non-identifiability in many LSMs, when p or q equals d . The most notable property of the indefinite orthogonal group is that it does not preserve Euclidean distances. This implies that any inference based on Euclidean distances in the latent space is not well-defined. For a detailed discussion, we refer the reader to Rubin-Delanchy et al. (2017), who studied inference under such a non-identifiability in the context of a generalized random dot product graph (RDPG) model. In particular, they showed that any post hoc clustering of the latent positions should use a Gaussian mixture model with elliptical covariance matrices since the clustering results are invariant to indefinite orthogonal transformations. This observation is essential if one intends to use the latent positions for community detection.

For a general d , a mild condition on the homophily coefficients is enough to restrict M_t to a signed-permutation matrix. In essence, we show that requiring the layers to measure different types of relations removes the latent space’s invariance under the general indefinite orthogonal group. We state the result in the following proposition.

Proposition 4.3. *Consider the same setup as in Proposition 4.1 and assume that conditions A4.1–A4.3 are satisfied. In addition, suppose the following condition is satisfied:*

A4.4. *For at least one layer $k \neq r$, $\text{rank}(\Lambda_k) = \text{rank}(\tilde{\Lambda}_k) = d$ and both $\Lambda_k \Lambda_r$ and $\tilde{\Lambda}_k \tilde{\Lambda}_r$ have distinct diagonal elements.*

Then $I_{p,q} = I_{p',q'}$ and each $\{M_t\}_{t=1}^T$ is a signed permutation matrix, i.e., $M_t = P \text{diag}(\mathbf{s})$ where $\mathbf{s} \in \{\pm 1\}^d$ and P is a $d \times d$ permutation matrix.

Proposition 4.3 says that Assumptions A4.1–A4.4 are strong enough to identify $\mathcal{X}_t$ up to sign-flips and permutations of its columns and the homophily coefficients $\Lambda_k = \text{diag}(\boldsymbol{\lambda}_k)$ up to the same set of permutations applied to the rows of $\boldsymbol{\lambda}_k$. Intuitively, Assumption A4.4 asserts that at least one layer should measure a homophily pattern distinct from the reference layer. For example, Assumption A4.4 is not satisfied for layers whose homophily coefficients satisfy $\Lambda_k = \alpha I_{p,q}$ for any scalar α . While mildly restrictive, we expect Assumption A4.4 to hold when the layers measure different phenomena. For example, we expect the cooperation and conflict layers that make up the international relation networks studied in the real data analysis of Section 4.5 to have distinct homophily patterns. Most importantly, Assumption A4.4 holds with

probability one under our choice of priors. As such, we assume each M_t is restricted to a signed permutation matrix for any value of d going forward.

4.3 Variational Inference

We presented the eigenmodel for multilayer dynamic networks and discussed issues of identifiability. Now, we turn to the problem of parameter estimation and inference. We take a Bayesian approach to inference with the goal of providing both posterior mean and credible intervals for the model’s parameters. However, the large amount of dyadic relations that comprise multilayer dynamic networks makes Markov chain Monte Carlo inference impractical for all but small networks. For this reason, we employ a variational approach (Wainwright & Jordan, 2008). For notational convenience, we collect the latent variables in $\boldsymbol{\theta} = \{\boldsymbol{\delta}_{1:K,1:T}, \Lambda_{1:K}, \mathcal{X}_{1:T}\}$ and the state space parameters in $\boldsymbol{\phi} = \{\tau^2, \sigma^2, \tau_\delta^2, \sigma_\delta^2\}$.

We aim to approximate the intractable posterior distribution $p(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathbf{Y}_{1:T}^1, \dots, \mathbf{Y}_{1:T}^K)$ with a tractable variational distribution $q(\boldsymbol{\theta}, \boldsymbol{\phi})$ that minimizes the KL divergence between $q(\boldsymbol{\theta}, \boldsymbol{\phi})$ and $p(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathbf{Y}_{1:T}^1, \dots, \mathbf{Y}_{1:T}^K)$. It can be shown that minimizing this divergence is equivalent to maximizing the evidence lower bound (ELBO), a lower bound on the data’s marginal log-likelihood

$$\mathcal{L}(q) = \mathbb{E}_{q(\boldsymbol{\theta}, \boldsymbol{\phi})} [\log p(\mathbf{Y}_{1:T}^1, \dots, \mathbf{Y}_{1:T}^K, \boldsymbol{\theta}, \boldsymbol{\phi}) - \log q(\boldsymbol{\theta}, \boldsymbol{\phi})] \leq \log p(\mathbf{Y}_{1:T}^1, \dots, \mathbf{Y}_{1:T}^K).$$

In general, the ELBO is not concave; however, optimization procedures often converge to a reasonable optimum. One still has the flexibility to specify the variational distribution’s form, although the need to evaluate and sample from it often guides this choice. A convenient form is the structured mean-field approximation, which factors $q(\boldsymbol{\theta}, \boldsymbol{\phi})$ into a product over groups of dependent latent variables. Furthermore, when the model consists of conjugate exponential family distributions, this form lends itself to a simple coordinate ascent optimization algorithm with optimal closed-form coordinate updates. For an introduction to variational inference, see Blei et al. (2017).

In what follows, we present a structured mean-field variational inference algorithm that preserves the eigenmodel’s essential statistical dependencies and maintains closed-form coordinate updates. Normally, the absence of conditional conjugacy in latent space models poses a challenge for closed-form variational inference. Indeed, popular solutions require additional approximations of the expected log-likelihood (Gollini & Murphy, 2016; Salter-Townshend & Murphy, 2013), which may bias parameter estimates. Another challenge is that the standard mean-field variational approximation is inadequate for our model due to the latent variable’s temporal dependencies. Our solution employs Pólya-gamma augmentation (Polson et al., 2013) and variational Kalman smoothing (Beal, 2003) to produce a new and widely applicable variational inference algorithm for bilinear latent space models for dynamic networks.

4.3.1 Pólya-gamma Augmentation

As previously mentioned, we use Pólya-gamma augmentation to render the model conditionally conjugate. For each dyad in the dynamic multilayer network, we introduce auxiliary Pólya-gamma latent variables $\omega_{ijt}^k \stackrel{\text{iid}}{\sim} \text{PG}(1, 0)$, where $\text{PG}(b, c)$ denotes a Pólya-gamma distribution with parameters $b > 0$ and $c \in \mathbb{R}$. For convenience, we use $\boldsymbol{\omega}$ to denote the collection of all Pólya-gamma auxiliary variables. As shown in Polson

et al. (2013), the joint distribution is now proportional to

$$p(\mathbf{Y}_{1:T}^1, \dots, \mathbf{Y}_{1:T}^K, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) \propto p(\boldsymbol{\theta})p(\boldsymbol{\phi})p(\boldsymbol{\omega}) \prod_{k=1}^K \prod_{t=1}^T \prod_{j < i} \exp \{ z_{ijt}^k \psi_{ijt}^k - \omega_{ijt}^k (\psi_{ijt}^k)^2 / 2 \},$$

where $z_{ijt}^k = Y_{ijt}^k - 1/2$ and $\psi_{ijt}^k = \delta_{k,t}^i + \delta_{k,t}^j + \mathbf{X}_t^{iT} \Lambda_k \mathbf{X}_t^j$. This joint distribution results in each latent variable's full conditional distribution lying within the exponential family, a property sufficient for closed-form variational inference.

4.3.2 The Structured Mean-Field Approximation

We use the following structured mean-field approximation to the augmented model's posterior

$$q(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) = \left[\prod_{h=1}^d q(\lambda_{1h}) \right] \left[\prod_{k=2}^K q(\boldsymbol{\lambda}_k) \right] \left[\prod_{k=1}^K \prod_{i=1}^n q(\delta_{k,1:T}^i) \right] \left[\prod_{i=1}^n q(\mathbf{X}_{1:T}^i) \right] \left[\prod_{k=1}^K \prod_{t=1}^T \prod_{j < i} q(\omega_{ijt}^k) \right] \\ \times q(\tau^2) q(\sigma^2) q(\tau_\delta^2) q(\sigma_\delta^2). \quad (4.2)$$

This factorization is attractive because it maintains the essential temporal dependencies in the posterior distribution. Since we use optimal variational factors, preserving these dependencies increases the approximate posterior distribution's accuracy.

4.3.3 Coordinate Ascent Variational Inference Algorithm

To maximize the ELBO, we employ coordinate ascent variational inference (CAVI). CAVI performs coordinate ascent on one variational factor at a time, holding the rest fixed. The optimal coordinate updates take a simple form: set each variational factor to the corresponding latent variable's expected full conditional probability under the remaining factors. For example, the update for $q(\mathbf{X}_{1:T}^i)$ is given by

$$\log q(\mathbf{X}_{1:T}^i) = \mathbb{E}_{-q(\mathbf{X}_{1:T}^i)} [\log p(\mathbf{X}_{1:T}^i \mid \cdot)] + c,$$

where $\mathbb{E}_{-q(\mathbf{X}_{1:T}^i)} [\cdot]$ indicates an expectation taken with respect to all variational factors except $q(\mathbf{X}_{1:T}^i)$, $p(\mathbf{X}_{1:T}^i \mid \cdot)$ is the full conditional distribution of $\mathbf{X}_{1:T}^i$, and c is a normalizing constant. When the full conditionals are members of the exponential family, a coordinate update involves calculating the natural parameter's expectations under the remaining variational factors.

The CAVI algorithm alternates between optimizing $q(\boldsymbol{\omega})$, $q(\boldsymbol{\delta}_{1:K,1:T})$, $q(\mathcal{X}_{1:T})$, $q(\Lambda_{1:K})$, and $q(\boldsymbol{\phi})$. Algorithm 4.1 outlines the full CAVI algorithm and defines some notation used throughout the rest of the chapter. We summarize each variational factor's coordinate update in the following sections. Section 4.7.2 and Section 4.7.3 provide the full details and derivations of the coordinate updates and the variational Kalman smoothers, respectively.

Updating $q(\omega_{ijt}^k)$

By the exponential tilting property of the Pólya-gamma distribution, we have

$$\log q(\omega_{ijt}^k) = \mathbb{E}_{-q(\omega_{ijt}^k)} [p_{\text{PG}}(\omega_{ijt}^k \mid 1, \psi_{ijt}^k)] + c,$$

Define the following expectations taken with respect to the full variational posterior:

$$\begin{aligned}\mathbb{E}[\mathbf{X}_t^i] &= \boldsymbol{\mu}_t^i, \quad \text{Var}(\mathbf{X}_t^i) = \Sigma_t^i, \quad \text{Cov}(\mathbf{X}_t^i, \mathbf{X}_{t+1}^i) = \Sigma_{t,t+1}^i, \\ \mathbb{E}[\delta_{k,t}^i] &= \mu_{\delta_{k,t}^i}, \quad \text{Var}(\delta_{k,t}^i) = \sigma_{\delta_{k,t}^i}^2, \quad \text{Cov}(\delta_{k,t}^i, \delta_{k,t+1}^i) = \sigma_{\delta_{k,t,t+1}^i}^2, \\ \mathbb{E}[\boldsymbol{\lambda}_k] &= \boldsymbol{\mu}_{\boldsymbol{\lambda}_k}, \quad \text{Var}(\boldsymbol{\lambda}_k) = \Sigma_{\boldsymbol{\lambda}_k}, \quad \mathbb{E}[\omega_{ijt}^k] = \mu_{\omega_{ijt}^k}.\end{aligned}$$

Iterate the following steps until convergence:

1. Update each $q(\omega_{ijt}^k) = \text{PG}(1, c_{ijt}^k)$ as in Algorithm 4.2.

2. Update

$$\begin{aligned}q(\delta_{k,1:T}^i) &: \text{a Gaussian state space model for } i \in \{1, \dots, n\} \text{ and } k \in \{1, \dots, K\}, \\ q(\tau_\delta^2) &= \Gamma^{-1}(\bar{a}_{\tau_\delta^2}/2, \bar{b}_{\tau_\delta^2}/2), \\ q(\sigma_\delta^2) &= \Gamma^{-1}(\bar{c}_{\sigma_\delta^2}/2, \bar{d}_{\sigma_\delta^2}/2),\end{aligned}$$

using a variational Kalman smoother as in Algorithm 4.3.

3. Update

$$\begin{aligned}q(\mathbf{X}_{1:T}^i) &: \text{a Gaussian state space model for } i \in \{1, \dots, n\}, \\ q(\tau^2) &= \Gamma^{-1}(\bar{a}_{\tau^2}/2, \bar{b}_{\tau^2}/2), \\ q(\sigma^2) &= \Gamma^{-1}(\bar{c}_{\sigma^2}/2, \bar{d}_{\sigma^2}/2),\end{aligned}$$

using a variational Kalman smoother as in Algorithm 4.4.

4. Update $q(\lambda_{1h}) = p_{\lambda_{1h}}^{\mathbb{1}_{\{\lambda_{1h}=1\}}} (1 - p_{\lambda_{1h}})^{\mathbb{1}_{\{\lambda_{1h}=-1\}}}$ for $h \in \{1, \dots, d\}$ as in Algorithm 4.5.

5. Update $q(\boldsymbol{\lambda}_k) = N(\boldsymbol{\mu}_{\boldsymbol{\lambda}_k}, \Sigma_{\boldsymbol{\lambda}_k})$ for $k \in \{2, \dots, K\}$ as in Algorithm 4.5.

Algorithm 4.1: Coordinate ascent variational inference for the eigenmodel for dynamic multilayer networks. Section 4.7.2 contains the details of Algorithms 4.2—4.5. Iterations are performed until successive differences of the expected log-likelihood, Equation (4.5), drop below a tolerance threshold.

where $p_{PG}(\omega \mid b, c)$ is the density of $PG(b, c)$ random variable. This density is a member of the exponential family with natural parameter $-(\psi_{ijt}^k)^2/2$. We provide the full coordinate update, which involves taking the expectation of $(\psi_{ijt}^k)^2$, in Algorithm 4.2 of Section 4.7.2.

Updating $q(\delta_{k,1:T}^i)$, $q(\tau_\delta^2)$, $q(\sigma_\delta^2)$

Under the Pólya-gamma augmentation scheme, the conditional distributions of the social trajectories take the form of linear Gaussian state space models. In particular,

$$\log q(\delta_{k,1:T}^i) = \log h(\delta_{k,1}^i) + \sum_{t=2}^T \log h(\delta_{k,t}^i \mid \delta_{k,t-1}^i) + \sum_{t=1}^T \log h(\mathbf{z}_{k,t}^i \mid \delta_{k,t}^i) + c, \quad (4.3)$$

where

$$\begin{aligned} \log h(\delta_{k,1}^i) &= \mathbb{E}_{q(\tau_\delta^2)} [\log N(\delta_{k,1}^i \mid 0, \tau_\delta^2)], \\ \log h(\delta_{k,t}^i \mid \delta_{k,t-1}^i) &= \mathbb{E}_{q(\sigma_\delta^2)} [\log N(\delta_{k,t}^i \mid \delta_{k,t-1}^i, \sigma_\delta^2)], \\ \log h(\mathbf{z}_{k,t}^i \mid \delta_{k,t}^i) &= \mathbb{E}_{-q(\delta_{k,1:T}^i)} \left[\sum_{j \neq i} \log N(z_{ijt}^k \mid \omega_{ijt}^k \delta_{k,t}^i + \omega_{ijt}^k (\delta_{k,t}^j + \mathbf{X}_t^{i\top} \Lambda_k \mathbf{X}_t^j), \omega_{ijt}^k) \right]. \end{aligned}$$

In the previous expressions, $\mathbf{z}_{k,t}^i \in \mathbb{R}^{n-1}$ is a vector that consists of stacking z_{ijt}^k for $j \neq i$ and $N(\mathbf{x} \mid \boldsymbol{\mu}, \Sigma)$ is the density of a $N(\boldsymbol{\mu}, \Sigma)$ random variable. Because all densities involved are Gaussian, the expectations yield Gaussian densities with natural parameters that depend on the remaining variational factors. Thus, we recognize the optimal variational distribution as a GSSM. The expected sufficient statistics needed to update the remaining variational factors can be computed with either the variational Kalman smoother (Beal, 2003) or a standard Kalman smoother under an augmented state space model (Barber & Chiappa, 2007). We use the variational Kalman smoother. Furthermore, the inverse-gamma priors on the state space parameters result in fully conjugate coordinate updates for τ_δ^2 and σ_δ^2 . The update for the social trajectories is presented in Algorithm 4.3 of Section 4.7.2.

Updating $q(\mathbf{X}_{1:T}^i)$, $q(\tau^2)$, $q(\sigma^2)$

Similar to the social trajectories, the conditional distributions of the latent trajectories are also GSSMs. Specifically,

$$\log q(\mathbf{X}_{1:T}^i) = \log h(\mathbf{X}_1^i) + \sum_{t=2}^T \log h(\mathbf{X}_t^i \mid \mathbf{X}_{t-1}^i) + \sum_{t=1}^T \log h(\mathbf{z}_t^i \mid \mathbf{X}_t^i) + c, \quad (4.4)$$

where

$$\begin{aligned} \log h(\mathbf{X}_1^i) &= \mathbb{E}_{q(\tau^2)} [\log N(\mathbf{X}_1^i \mid 0, \tau^2)], \\ \log h(\mathbf{X}_t^i \mid \mathbf{X}_{t-1}^i) &= \mathbb{E}_{q(\sigma^2)} [\log N(\mathbf{X}_t^i \mid \mathbf{X}_{t-1}^i, \sigma^2)], \\ \log h(\mathbf{z}_t^i \mid \mathbf{X}_t^i) &= \mathbb{E}_{-q(\mathbf{X}_{1:T}^i)} \left[\sum_{k=1}^K \sum_{j \neq i} \log N(z_{ijt}^k \mid \omega_{ijt}^k (\delta_{k,t}^i + \delta_{k,t}^j) + \omega_{ijt}^k \mathbf{X}_t^{i\top} \Lambda_k \mathbf{X}_t^j, \omega_{ijt}^k) \right]. \end{aligned}$$

In the previous expressions, $\mathbf{z}_t^i \in \mathbb{R}^{K(n-1)}$ is a vector formed by stacking z_{ijt}^k for $j \neq i$ and $k = 1, \dots, K$. Once again, we recognize that $q(\mathbf{X}_{1:T}^i)$ is a GSSM; therefore, we can calculate the expected sufficient statistics

with the variational Kalman smoother. Also, the inverse-gamma priors on τ^2 and σ^2 result in closed form coordinate updates. The updates for the latent trajectories are presented in Algorithm 4.4 of Section 4.7.2.

Updating $q(\Lambda_k)$

Given the augmented model's conjugacy, the homophily coefficients will be Bernoulli for the reference layer and Gaussian for the other layers. The corresponding coordinate updates, which involve calculating the Bernoulli probabilities and performing standard Bayesian linear regression, are presented in Algorithm 4.5 of Section 4.7.2.

4.3.4 Convergence Criteria

Although it is possible to calculate the ELBO to determine convergence, evaluating the state space terms is computationally expensive. Instead, we monitor the expected log-likelihood

$$\mathcal{F}(q) = \sum_{k=1}^K \sum_{t=1}^T \sum_{j < i} (Y_{ijt}^k - 1/2) \mathbb{E}_{q(\boldsymbol{\theta})} [\psi_{ijt}^k] - \frac{1}{2} \mathbb{E}_{q(\omega_{ijt}^k)} [\omega_{ijt}^k] \mathbb{E}_{q(\boldsymbol{\theta})} [(\psi_{ijt}^k)^2], \quad (4.5)$$

which upper bounds the ELBO. We say the algorithm converged when the difference in the expected log-likelihood is less than 10^{-2} between iterations or the number of iterations exceeded 1,000. Due to the ELBO's non-convexity, we run the algorithm with ten different random initializations and choose the model with the highest expected log-likelihood. For details on our initialization procedure and hyper-parameter settings, see Section 4.7.4.

4.3.5 Inference of Identifiable Parameters

Recall that a centered latent space is a sufficient condition for parameter identifiability. As such, we make inference on the following parameters based on the approximate posterior:

$$\begin{aligned} \tilde{\mathbf{X}}_t^i &= \mathbf{X}_t^i - \frac{1}{n} \sum_{j=1}^n \mathbf{X}_t^j, \\ \tilde{\delta}_{k,t}^i &= \delta_{k,t}^i + \tilde{\mathbf{X}}_t^{iT} \Lambda_k \mathbf{c} + \frac{1}{2} \mathbf{c}^T \Lambda_k \mathbf{c}, \end{aligned}$$

where $\mathbf{c} = (1/n) \sum_{j=1}^n \mathbf{X}_t^j$. Under our approximation, the marginal posterior distributions of the $\tilde{\mathbf{X}}_t^i$'s are Gaussian with moments

$$\begin{aligned} \mathbb{E}_{q(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega})} [\tilde{\mathbf{X}}_t^i] &= \tilde{\boldsymbol{\mu}}_t^i = \boldsymbol{\mu}_t^i - \frac{1}{n} \sum_{j=1}^n \boldsymbol{\mu}_t^j, \\ \text{Var}(\tilde{\mathbf{X}}_t^i) &= \tilde{\Sigma}_t^i = \left(1 - \frac{1}{n}\right)^2 \Sigma_t^i + \left(\frac{1}{n}\right)^2 \sum_{j \neq i} \Sigma_t^j, \end{aligned} \quad (4.6)$$

where the variance is respect to $q(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega})$ as well. We calculate each $\tilde{\delta}_{k,t}^i$'s posterior mean and 95% credible interval using 2,500 samples from the approximate posterior distribution because their approximate posterior distributions lacks an analytic form.

4.4 Simulation Studies

This section presents a simulation study designed to assess the scaling of the proposed algorithm’s estimation error and dyad-wise prediction error. We considered three scenarios: *Scenario 1.* an increase in the number of nodes with $(n, K, T) \in \{50, 100, 200, 500, 1000\} \times \{5\} \times \{10\}$, *Scenario 2.* an increase in the number of layers with $(n, K, T) \in \{100\} \times \{5, 10, 20\} \times \{10\}$, and *Scenario 3.* an increase in the number of time points with $(n, K, T) \in \{100\} \times \{5\} \times \{10, 50, 100\}$. For each scenario, we sampled 30 independent parameter settings as follows:

1. Generate the reference homophily coefficients: $\lambda_{1h} = 2u_h - 1$ for $1 \leq h \leq d$, where $u_h \stackrel{\text{iid}}{\sim} \text{Bernoulli}(0.5)$.
2. Generate the remaining homophily coefficients: $\lambda_k \stackrel{\text{iid}}{\sim} U[-2, 2]^d$ for $2 \leq k \leq K$.
3. Generate initial sociality effects: $\delta_{k,1}^i \stackrel{\text{iid}}{\sim} U[-4, 4]$ for $1 \leq i \leq n$ and $1 \leq k \leq K$.
4. Generate the social trajectories: For $t = 2, \dots, T$, sample $\delta_{k,t}^i \sim N(\delta_{k,t-1}^i, 0.1)$ for $1 \leq i \leq n$ and $1 \leq k \leq K$.
5. Generate initial latent positions: $\mathbf{X}_1^i \stackrel{\text{iid}}{\sim} N(0, 4I_d)$ for $1 \leq i \leq n$.
6. Generate the latent trajectories: For $t = 2, \dots, T$, sample $\mathbf{X}_t^i \sim N(\mathbf{X}_{t-1}^i, 0.05I_d)$ for $1 \leq i \leq n$.
7. Center the latent space: For $t = 1, \dots, T$, set $\tilde{\mathbf{X}}_t^i = \mathbf{X}_t^i - (1/n) \sum_{j=1}^n \mathbf{X}_t^j$ for $1 \leq i \leq n$.

We set the dimension of the latent space $d = 2$. We sampled a single undirected adjacency matrix for each generated model using the dyad-wise probabilities in Equation (4.1).

To evaluate the estimated model’s accuracy, we computed relative errors according to the Frobenius norm of the difference between the parameters’ posterior means and their true values. Because the true homophily coefficients are distinct, Proposition 4.3 states that the latent positions are identifiable up to column permutations and sign-flips. To account for this invariance, we calculated the latent position’s time-averaged relative error as

$$\frac{1}{T} \sum_{t=1}^T \min_{P \in \Pi_d, \mathbf{s} \in \{-1, 1\}^d} \frac{\|\tilde{\mathcal{X}}_t - \hat{\mathcal{X}}_t P \text{diag}(\mathbf{s})\|_F^2}{\|\tilde{\mathcal{X}}_t\|_F^2},$$

where Π_d is the set of permutation matrices on d elements, $\|\cdot\|_F$ is the Frobenius norm, $\tilde{\mathcal{X}}_t = (\tilde{\mathbf{X}}_t^1, \dots, \tilde{\mathbf{X}}_t^n)^T$, and $\hat{\mathcal{X}}_t = (\tilde{\boldsymbol{\mu}}_t^1, \dots, \tilde{\boldsymbol{\mu}}_t^n)^T$ where $\tilde{\boldsymbol{\mu}}_t^i$ is defined in Equation (4.6). Similarly, we computed the relative error of the homophily coefficients accounting for invariance under simultaneous permutations of their rows and columns:

$$\min_{P \in \Pi_d} \frac{\sum_{k=1}^K \|\Lambda_k - P^T \text{diag}(\boldsymbol{\mu}_{\lambda_k}) P\|_F^2}{\sum_{k=1}^K \|\Lambda_k\|_F^2}.$$

Lastly, we calculated the relative errors for the centered social trajectories and the dyad-wise probabilities, both of which do not have identifiability issues. For computational expediency, we calculated the dyad-wise probabilities by plugging-in the posterior means into Equation (4.1), e.g.,

$$\hat{\mathbb{P}}(Y_{ijt}^k = 1 \mid \mu_{\delta_{k,t}^i}, \mu_{\delta_{k,t}^j}, \boldsymbol{\mu}_{\lambda_k}, \boldsymbol{\mu}_t^i, \boldsymbol{\mu}_t^j) = \text{logit}^{-1} \left[\mu_{\delta_{k,t}^i} + \mu_{\delta_{k,t}^j} + \boldsymbol{\mu}_t^{i^T} \text{diag}(\boldsymbol{\mu}_{\lambda_k}) \boldsymbol{\mu}_t^j \right],$$

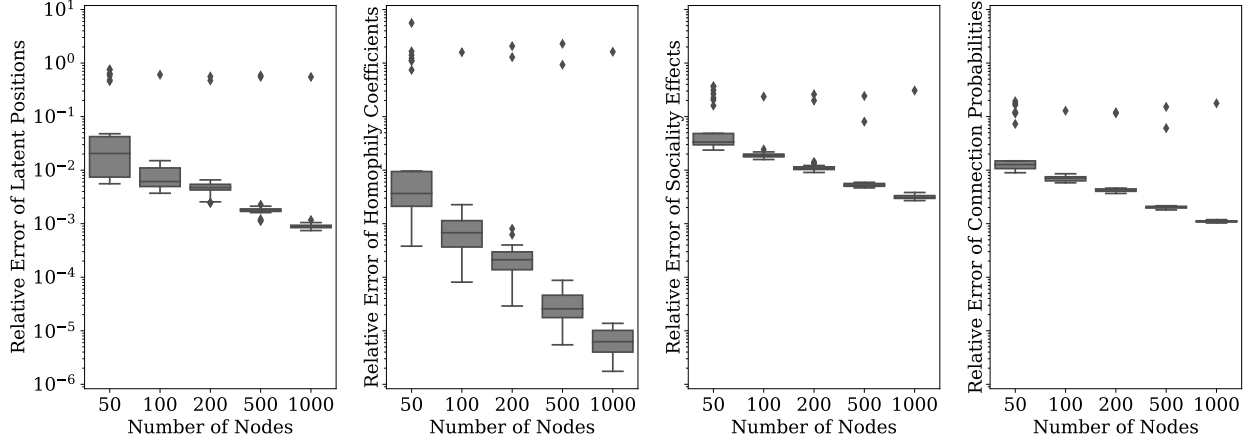


Figure 4.2: Relative estimation errors of the model’s parameters as the number of nodes n increases. Boxplots show the distribution over 30 simulations.

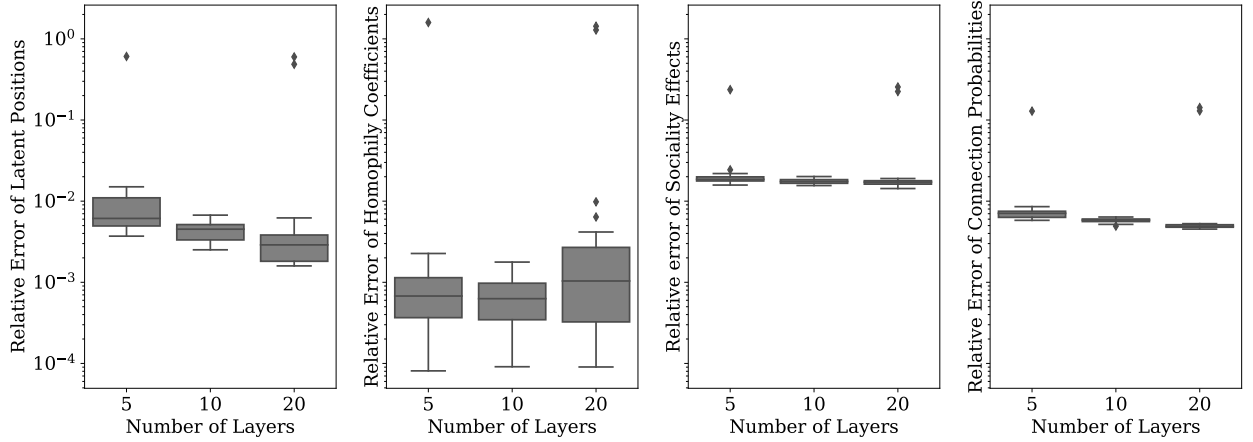


Figure 4.3: Relative estimation errors of the model’s parameters as the number of layers K increases. Boxplots show the distribution over 30 simulations.

which is an upper-bound on the approximate posterior mean of the dyad-wise probability. We can use Monte Carlo to estimate the dyad-wise probabilities’ approximate posterior mean by sampling from the approximate posterior if desired.

The estimation errors for varying n , K , T are displayed in the boxplots in Figure 4.2, Figure 4.3, and Figure 4.4, respectively. Overall, the CAVI algorithm recovers the model’s parameters with high accuracy. The starkest improvement in estimation accuracy occurs as the number of nodes increases. This improvement is partly due to the more accurate estimation of the homophily coefficients. Due to the model’s ability to pool information across layers, the latent positions’ relative error decreases as K increases. Such an improvement is not observed for the social trajectories because the number of social trajectories grows with the number of layers. Surprisingly, the homophily coefficients’ estimation error does not improve as T increases, although the estimation error is already low at roughly 10^{-3} . Since the relative error of the latent positions and social trajectories is on the order of 10^{-2} , we conclude that algorithm’s ability to estimate the latent positions and social trajectories accurately dominates the error. Furthermore, the latent positions’ estimation error slightly degrades as the number of time steps increases. Such deterioration is typical in smoothing problems.

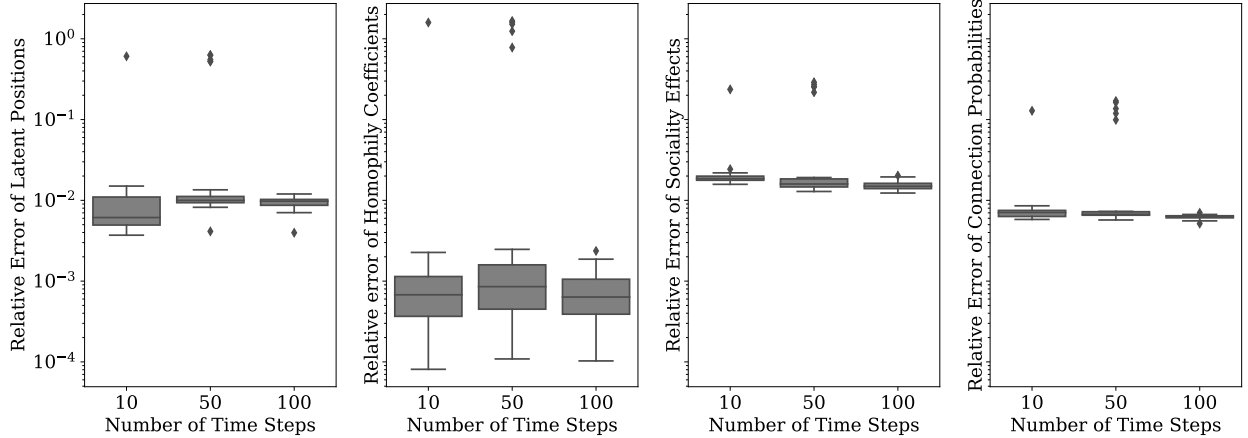


Figure 4.4: Relative estimation errors of the model’s parameters as the number of time steps T increases. Boxplots show the distribution over 30 simulations.

Next, we evaluated the model’s predictive performance by calculating the area under the receiver operating characteristic curve (AUC) for in-sample and held-out dyads. To evaluate held-out predictions, we removed 20% of the dyads randomly from each layer and time step during estimation. Figure 4.5’s boxplots summarize the prediction errors for increasing n , K , and T . The in-sample and holdout AUC are close to the maximum value of one for all scenarios. Overall, the simulations demonstrate that the CAVI algorithm is scalable and accurate.

4.5 Real Data Applications

In this section, we demonstrate how to use the proposed model to analyze real-world data sets. We consider networks from political science and epidemiology. The first example studies a time series of different international relations between 100 countries over eight years. The second example applies the model to a contact network of 242 individuals at a primary school measured over two days to quantify heterogeneities in infectious disease spread throughout the school day.

4.5.1 International Relations

This application explores the temporal evolution of different relations between socio-political actors. The raw data consists of (source actor, target actor, event type, time-stamp) tuples collected by the Integrated Crisis Early Warning System (ICEWS) project (Boschee et al., 2015), which automatically identifies and extracts international events from news articles. The event types are labeled according to the CAMEO taxonomy (Gerner et al., 2008). The CAMEO scheme includes twenty labels ranging from the most neutral “1 — make public statement” to the most negative “20 — engage in unconventional mass violence.”

Our sample consists of monthly event data between countries during the eight years of the Obama administration (2009 - 2017). We grouped the event types into four categories known as “QuadClass” (Duval & Thompson, 1980). These classes split events along four dimensions: (1) *verbal cooperation* (labels 2 to 5), (2) *material cooperation* (labels 6 to 7), (3) *verbal conflict* (labels 8 to 16), and (4) *material conflict* (labels 17 to 20). At a high-level, the first two classes represent friendly relations such as “5 — engage in

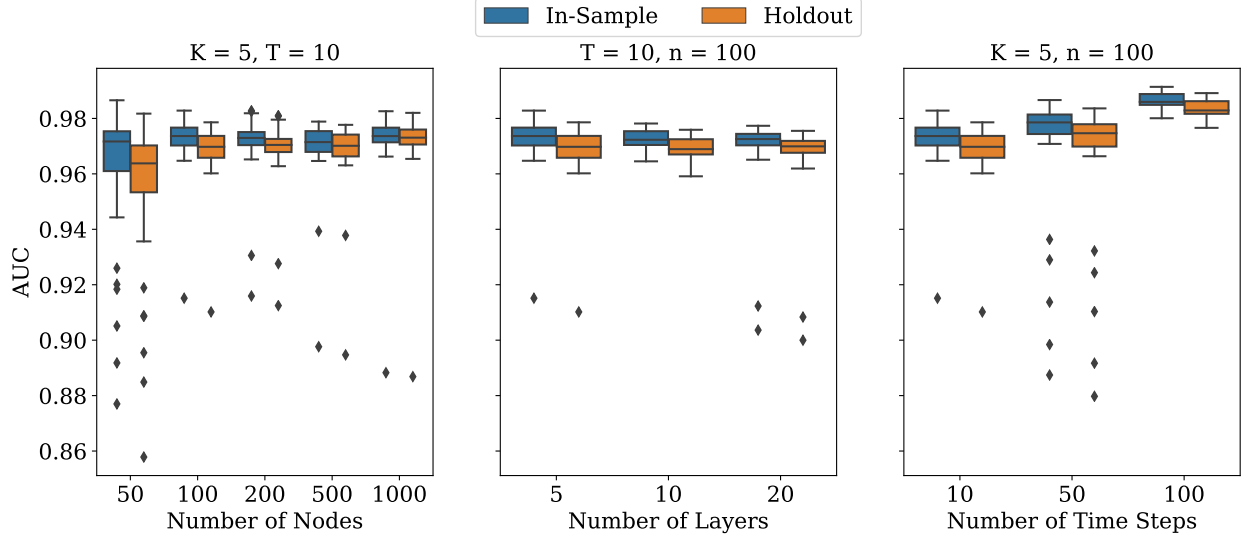


Figure 4.5: Dyad-wise predictive performance measured by the in-sample and holdout AUC as the number of nodes n (left), the number of layers K (center), and the number time steps T (right) increases. Boxplots show the distribution over 30 simulations.

diplomatic cooperation” and “7 — provide aid”, while the last two classes reflect hostile relations such as “13 — threaten” and “19 — assault”.

Statistical Network Analysis of the ICEWS Data

We structured the ICEWS data as a dynamic multilayer network recording which four event types occurred between nations each month from 2009 until the end of 2016. Each event type is a layer in the multilayer networks. We chose verbal cooperation as the reference layer because it contains the densest networks. We limited the actors to the 100 most active countries during this period. This preprocessing resulted in a dynamic multilayer network with $K = 4$ layers, $T = 96$ time steps, and $n = 100$ actors. An edge ($Y_{ijt}^k = 1$) means that country i and country j had at least one event of type k during the t th month, where $t = 1$ corresponds to January 2009. We fit the model using the procedure described in Section 4.3. The model’s in-sample AUC was 0.90, which indicates a good fit to the data.

Detection of Historical Events During the Obama Administration

We validate the model by demonstrating that the inferred social trajectories and latent space dynamics reflect major international events. We focus on three events: the Arab Spring, the American-led intervention in Iraq, and the Crimea Crisis. Specifically, we concentrate on interpreting the latent parameters for Libya, Syria, Iraq, the United States, Russia, and Ukraine since they played a large role in these events.

Because these events involve conflict, we start by analyzing each country’s material conflict social trajectory, i.e., $\delta_{4,1:T}^i$ for $1 \leq i \leq n$. Figure 4.6 plots these social trajectories’ posterior means with a few select countries highlighted. Section 4.7.5 contains the same plot for the remaining three layers. Most social trajectories are relatively flat. Indeed, the 95% credible interval for the step size standard deviation σ_δ is (0.0619, 0.0628), which is much smaller than that of the initial standard deviation τ_δ , which equals (2.10, 2.42). However, the social trajectories of Iraq, Syria, and Libya demonstrate dramatic changes. Specifically, Libya

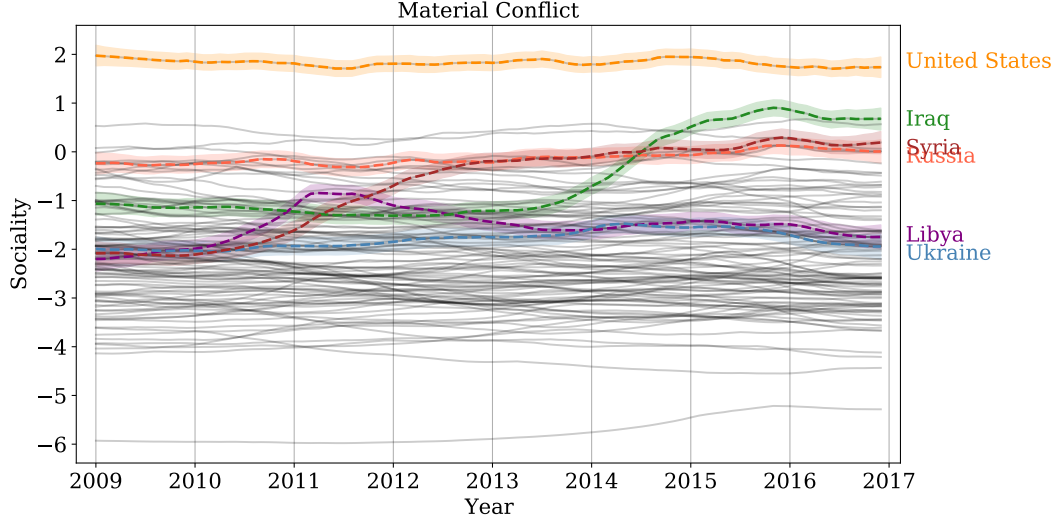


Figure 4.6: Posterior means of the material conflict social trajectories. Select countries are highlighted in color with bands that represent 95% credible intervals. The remaining countries' social trajectories are displayed with gray curves.

and Syria both increase their material conflict sociality at the start of the Arab Spring in 2011. In particular, Libya's sociality spikes during the Libyan Civil War in 2011 that saw Muammar Gaddafi's regime overthrown. Iraq's sociality increases leading up to and throughout the United States' escalated military presence in 2014. Note that the Crimea Crisis, which began with Russia annexing the Crimea Peninsula in February 2014, is not reflected in Ukraine's or Russia's social trajectory. This conflict is missing because an actor's social trajectory reflects their global standing in the network while the Crimea Crisis is primarily a regional conflict. In contrast, the latent space, which captures local transitive effects, should reflect this more localized conflict.

We begin analyzing the latent space by interpreting the estimated homophily coefficients, Λ_k (Figure 4.7). All layers exhibit assortativity along both latent dimensions. Interestingly, we notice similarities in how the cooperation and the conflict layers use the latent space. The homophily coefficients' 95% credible intervals overlap along the first dimension for the verbal conflict and the material conflict layers. Also, the credible intervals overlap along the second dimension for the verbal cooperation and the material cooperation layers. Furthermore, the conflict layers have larger homophily coefficients than the cooperation layers. To interpret this result, we visualize the latent space's layout.

Figure 4.8 displays the estimated latent space during February 2012 and February 2014. The latent space encodes the geographic locations of the countries. Due to the positive homophily of the relations, actors are more likely to connect when their latent positions share a common angle. Eastern European nations are on the top left, Latin American nations are on the top right, African nations are on the bottom right, and Middle Eastern nations are on the bottom left. Furthermore, highly sociable nations, such as the United States, are near the center of the latent space because their high sociality explains most of their interactions. Overall, we conclude that the higher values of the conflict homophily coefficients indicate that regional (geographic) effects play a more prominent role in predicting conflict than cooperation.

Finally, we demonstrate how the latent space reflects the regional Crimea Crisis between Russia and Ukraine in early 2014. Figure 4.9 displays the latent trajectories for the two nations. Unlike the actor's

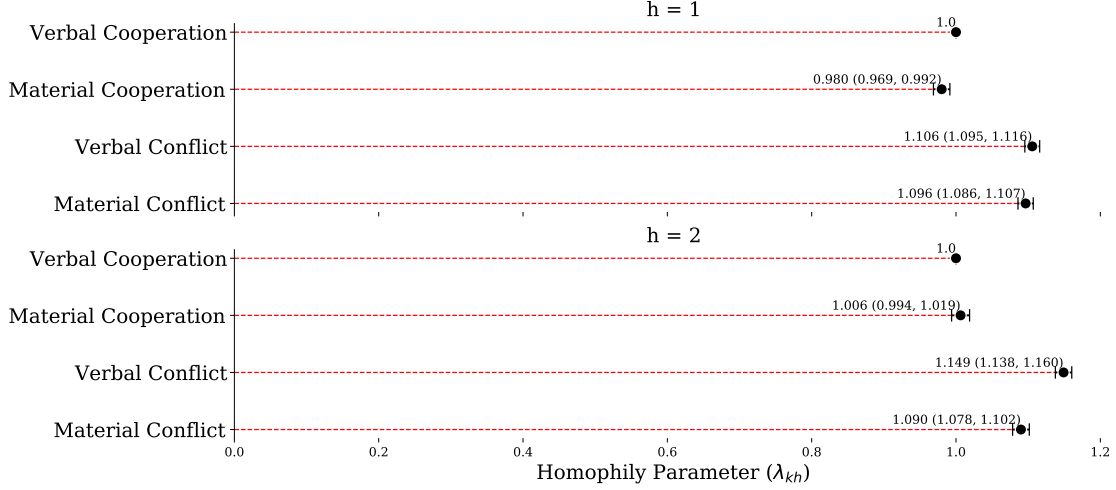


Figure 4.7: The homophily coefficients’ posterior means and 95% credible intervals for the ICEWS network’s four relations. The top and bottom plots give estimates for the degree of homophily along the first and second latent dimensions, respectively.

social trajectories, their latent trajectories are highly variable and encompass the Crimea Crisis. Around the second half of 2013, Ukraine’s latent feature along the second dimension increases significantly, reaching a maximum in early 2014. During this time, Russia’s second latent feature also increased. Comparing Ukraine and Russia’s latent positions in February 2012 to those in February 2014 in Figure 4.8, we see that they align themselves while moving toward the periphery of the latent space. These dynamics result in an increased connection probability between the two nations in all layers during the crisis, see Figure 4.10. Overall, we conclude that the latent trajectories reflect regional events in the ICEWS data.

4.5.2 Epidemiological Face-to-Face Contact Networks

This case study uses our proposed model to analyze longitudinal face-to-face contact networks drawn from an epidemiological survey of students at a primary school (grades 1 to 5) in Lyon, France. Such contact networks influence mathematical models of infectious disease spread in varying populations (Wallinga et al., 2006; Zagheni et al., 2008). Also, the analysis of these contact patterns allows school administrators to mitigate infectious disease spread in classrooms by determining the times during the day when spread is most prevalent. In the exploratory phase, these analyses often have difficulty visualizing the complicated dynamic networks. Furthermore, they often do not formally quantify the uncertainty in network statistics. In this section, we demonstrate how our model provides a meaningful network visualization and quantification of uncertainty.

The contact networks were collected by the SocioPatterns collaboration (<http://www.sociopatterns.org>) and initially analyzed in Stehlé et al. (2011). Contact data is available for 242 individuals (232 children and 10 teachers) belonging to grades 1 through 5. Each grade is split into two sections (A and B) so that there are ten classes overall. Each class has its own classroom and teacher. The school day runs from 8:30 am to 4:30 pm, with a lunch break from 12:00 pm to 2:00 pm and two breaks of 20 to 25 minutes around 10:30 am and 3:30 pm.

The face-to-face contacts occurred over two days: Thursday, October 1st, 2009, and Friday, October 2nd, 2009. Data was collected from 8:45 am to 5:20 pm on the first day and from 8:30 am to 5:05 pm on

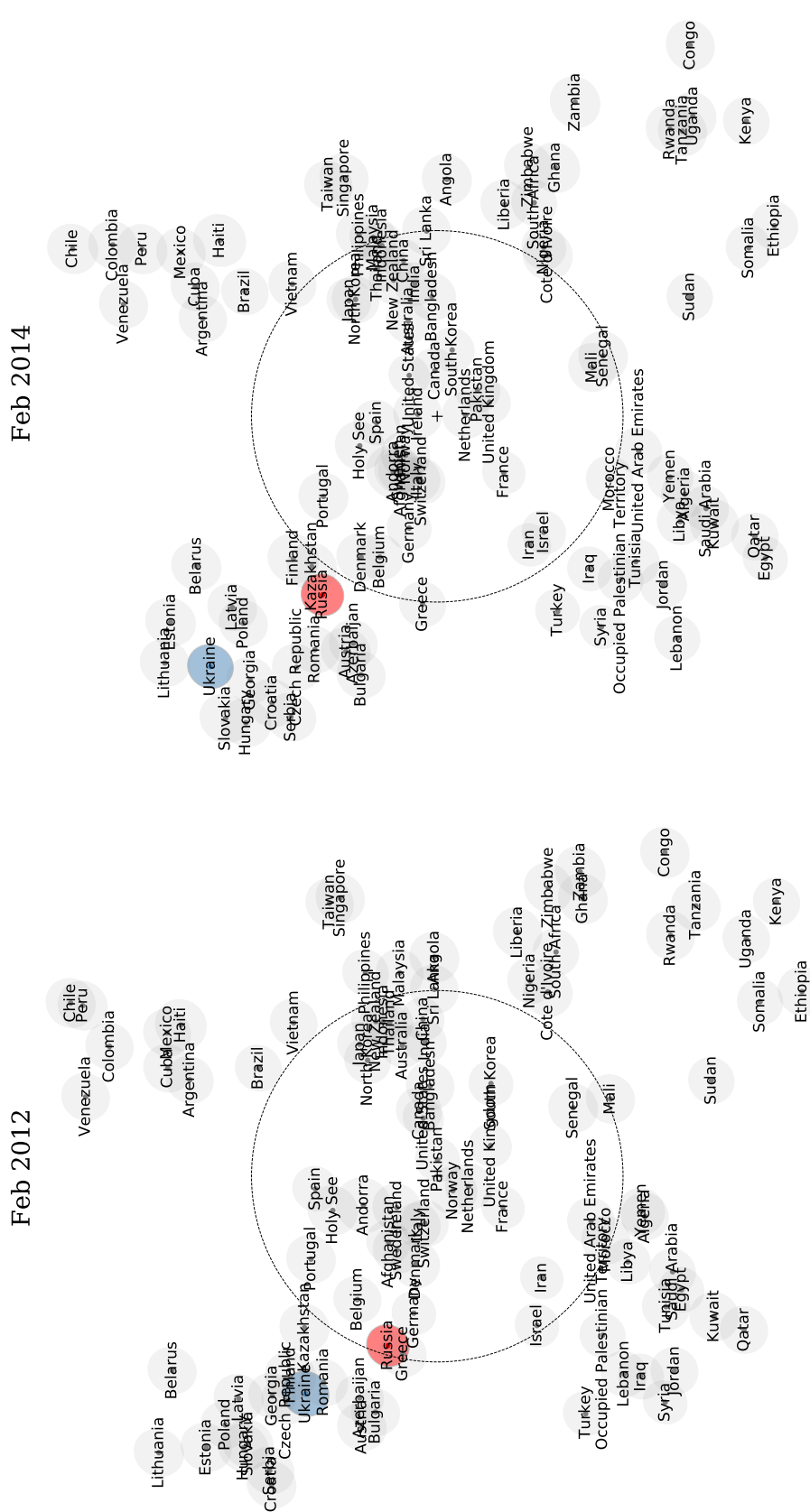


Figure 4.8: Estimated latent space for the ICEWS networks on February 2012 (left) and February 2014 (right). The origin of the latent space is denoted by a + and the initial variance τ^2 is displayed as a dotted one-standard deviation ellipse. The names of each nation are annotated. The ellipses are two standard deviation ($\sim 95\%$) credible ellipses for each actor's latent position. Ukraine and Russia are highlighted in blue and red, respectively.

the second day. Radio-frequency identification (RFID) devices measured the contacts between individuals. The RFID sensor registered a contact when two individuals were within 1 to 1.5 meters during a 20-second interval. This distance range was chosen to correspond to the range over which a communicable infectious disease could spread. For a detailed description of the data collection technology, see Cattuto et al. (2010).

Statistical Network Analysis of the School Contact Network

We structured the face-to-face contact data as a dynamic multilayer network recording face-to-face interactions each day. We treated each day as a layer so that the layers correspond to Thursday and Friday. We set Thursday as the reference layer. In concordance with the analysis in Stehlé et al. (2011), we divided the daily contact networks into 20-minute time intervals between 9:00 am and 5:00 pm and extended the first and last time intervals to accommodate the different starting and ending times of the experiment on the two days. This preprocessing resulted in a dynamic multilayer network with $K = 2$ layers, $T = 24$ time steps, and $n = 242$ actors. Specifically, an edge ($Y_{ijt}^k = 1$) means that actor i and actor j had at least one registered interaction during the t th 20-minute interval on day k . We fit the model using the procedure detailed in Section 4.3. The model’s in-sample AUC was 0.96, which indicates a good fit to the data.

Dynamics of the Epidemic Branching Factor

Here, we demonstrate how to use our model to (1) determine periods in the school day most susceptible to the spread of infectious disease and (2) identify differences in the contact patterns between the two days.

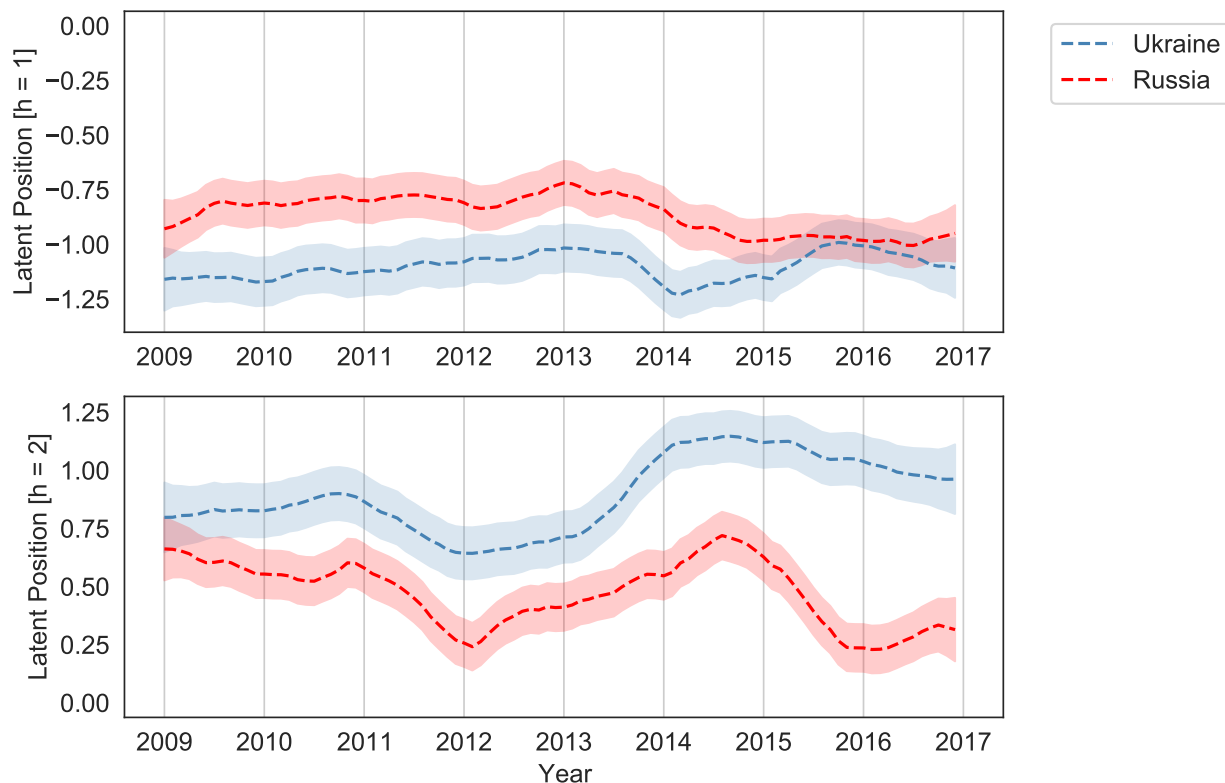


Figure 4.9: Posterior means and 95% credible intervals of Ukraine and Russia’s latent trajectories. The top and bottom plots give estimates for the first and second latent dimensions, respectively.

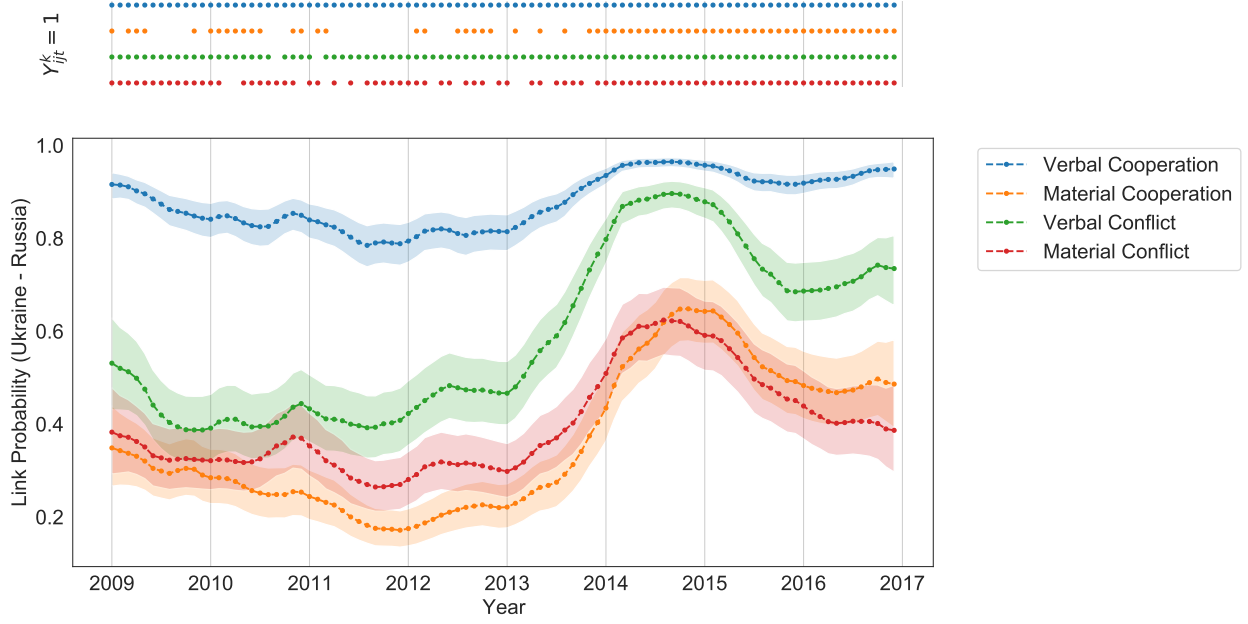


Figure 4.10: Raw values of the adjacency matrices where a dot indicates that Ukraine and Russia had a particular relation during that month (top). The posterior means and 95% credible intervals for the monthly link probability between the two nations across the four international relations (bottom). Estimates are calculated using 1,000 samples from the approximate posterior.

To quantify a network’s contribution to the spread of infectious disease, we use the epidemic branching factor (Andersson, 1998), defined as

$$\kappa = \frac{\sum_{i=1}^n d_i^2/n}{\sum_{i=1}^n d_i/n},$$

where d_i is the i th node’s degree. The epidemic branching factor is related to the basic reproduction number, R_0 , which is (loosely) equal to the number of secondary infections caused by a typical infectious individual during an epidemic’s early stages (Anderson & May, 1991). In network-based susceptible-infected-recovered (SIR) models, R_0 equals $\tau(\kappa - 1)/(\tau + \gamma)$, where τ and γ are infection and recovery rates, respectively (Andersson, 1997). This relation implies that larger branching factors lead to more massive epidemics.

Figure 4.11 depicts the posterior distribution of the epidemic branching factor. The boxplots contain 250 networks, each sampled from a different set of latent variables drawn from the model’s approximate posterior. The model matches the observed branching factor for most time steps; however, it underestimates the most dramatic changes at 10:40 am to 11:00 am, 12:00 pm to 12:20 pm, 1:00 pm to 1:20 pm, and 4:00 pm to 4:20 pm. Regardless, the model still captures these four spikes in the branching factor. Intuitively, the timings of these spikes occur during lunchtime (12:00 pm to 2:00 pm) and the two short breaks (around 10:30 am and 3:30 pm). We expect such events to lead to increased disease spread because they allow students from different classrooms to mix. More surprisingly, the branching factor’s dynamics differ between the two days. The most apparent difference is the spike from 10:40 am to 11:00 am on Thursday that is not present on Friday. The difference in branching factors between Thursday and Friday from 10:40 am to 11:00 am is significantly greater than zero, with the difference’s 95% credible interval equaling (3.11, 5.78). To understand what caused this difference, we analyzed the shared latent space. We defer a discussion of the actor’s social trajectories to Section 4.7.5.

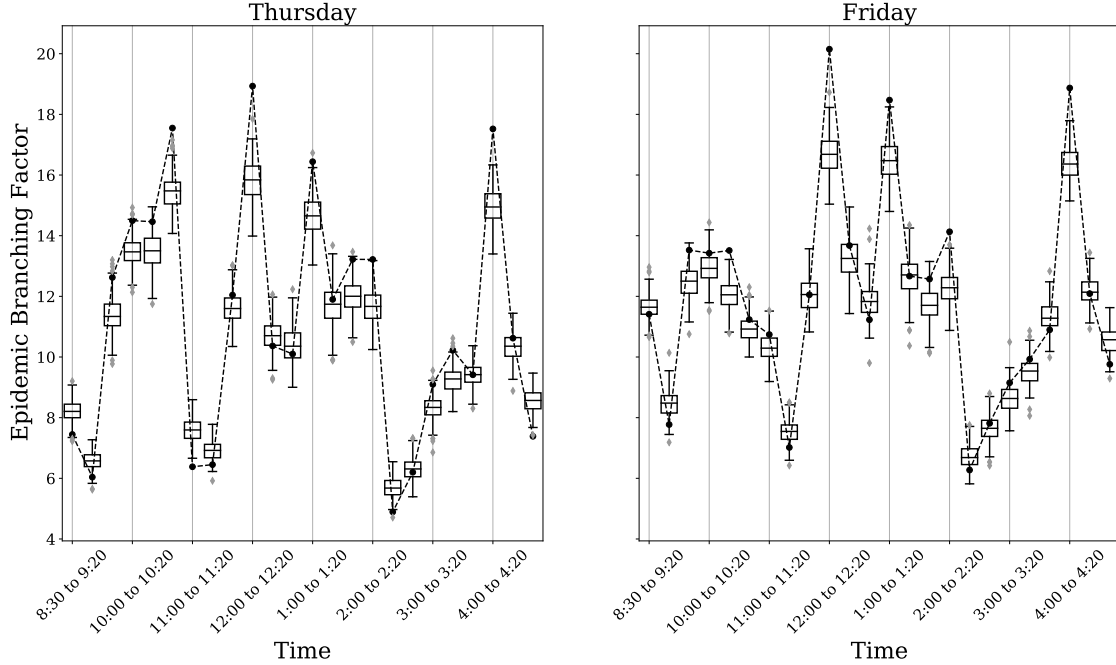


Figure 4.11: Epidemic branching factors for the face-to-face contact networks on Thursday (left) and Friday (right) at different times throughout the school day. The dashed black curves depict the observed network’s branching factor. Boxplots show the range of the branch factor’s posterior distribution.

Figure 4.12 depicts the latent positions’ posterior means and the observed edges on Thursday and Friday during the first short break from 10:40 am to 11:00 am. The inferred homophily coefficients are all positive and significantly different between layers (see Figure 4.18 in Section 4.7.5). The latent space accurately clusters the students into their ten classrooms. The two layers share the same classroom structure, which affirms our choice of a shared latent space. The difference in branching factors is due to the varying mixing patterns between the classrooms on the two days. Specifically, the classrooms that interact on the two days are different. On Thursday, there are many contacts between students in classes 1A, 1B, 2A, 3A, 3B, and 4B. In contrast, on Friday, classes 1A, 2A, 2B, 4B, and 5B interact. Furthermore, the number of edges between classrooms is much lower on Friday than on Thursday. This observation implies a simple intervention to mitigate disease spread: stagger each classroom’s break time in order to limit contacts between students of different classes, which will lower the epidemic branching factor.

4.6 Discussion

This chapter proposed a flexible, interpretable, and computationally efficient latent space model for dynamic multilayer networks. Our eigenmodel for dynamic multilayer networks decomposes the dyadic data into a common time-varying latent space used differently by the layers through layer-specific homophily levels and additive node-specific social trajectories that account for further degree heterogeneity. Also, we determined and corrected for various identifiability issues. This accomplishment allows for an intuitive interpretation of the latent space, unlike previous nonparametric models (Durante et al., 2017). Next, we developed an efficient variational inference algorithm for parameter estimation. Unlike previous variational approaches, we maintain the essential temporal dependencies in the posterior approximation. Furthermore, our varia-

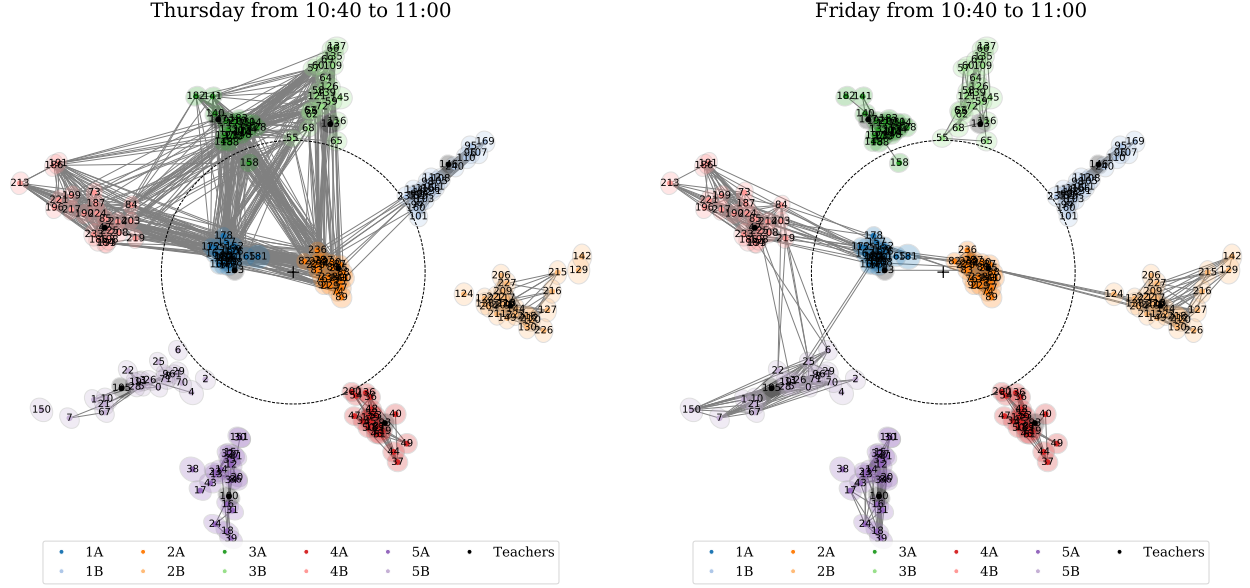


Figure 4.12: Latent positions’ 95% credible ellipses for the primary school face-to-face contact networks from 10:40 am to 11:00 am. The gray lines indicate observed edges on Thursday (left) and Friday (right). The students are colored by their classroom and section, while teachers are displayed in black. The ‘+’ denotes the origin of the latent space and the dotted circle indicates a one standard deviation ellipse with variance τ^2 .

tional algorithm is widely applicable to general dynamic bilinear latent space models. A simulation study established the effectiveness of our estimation procedure to scale to various network sizes. Finally, we demonstrated how to use our model to analyze international relations from 2009 to 2017 and understand the spread of an infectious disease in a primary school contact network.

In this work, we always set the latent space dimension $d = 2$, which allows for visualization; however, one may want a data-driven choice of d . One possibility is to use information criteria such as the Akaike information criteria (AIC), deviance information criteria (DIC), or Bayesian information criteria (BIC) to perform model selection. When the purpose of the model is to predict unobserved dyads, cross-validation procedures are a reasonable solution. In this case, we can either perform dyad-wise V -fold cross-validation (P. D. Hoff, 2005) or network cross-validation (K. Chen & Lei, 2018). Lastly, one can examine the posterior predictive distribution of statistics of interest, $T(\mathbf{Y}_{1:T}^1, \dots, \mathbf{Y}_{1:T}^K)$, and select the smallest d such that there is no substantial lack of fit. Such posterior predictive checks are standard in the social network literature (Hunter et al., 2008). A theoretically sound and easy-to-compute model selection criteria would be beneficial for bilinear LSMs.

Many real-world networks contain non-binary relations. One can adopt the proposed model to networks with non-binary edges with minor changes. For example, replacing the Bernoulli likelihood in Equation (4.1) with a Gaussian likelihood can model real-valued networks with minimal changes to the variational algorithm. However, extending the variational algorithm to general exponential family likelihoods, such as Poisson or negative binomial, is a direction for future research.

Relations are often directed in nature; therefore, it is natural to generalize the model to directed networks. Such a model needs to allow for varying levels of reciprocity in the directed relations. A simple extension of

our model to directed networks is

$$Y_{ijt}^k \stackrel{\text{ind.}}{\sim} \text{Bernoulli}(\text{logit}^{-1}[\Theta_{ijt}^k]), \quad \text{with} \quad \Theta_{ijt}^k = \delta_{k,t}^i + \gamma_{k,t}^j + \mathbf{X}_t^{i\top} \Lambda_k \mathbf{Z}_t^j,$$

where $Y_{ijt}^k = 1$ ($Y_{ijt}^k = 0$) denotes the presence (absence) of a directed edge from i to j in layer k at time t . The latent variables' distributions are

$$\gamma_{k,1}^i \stackrel{\text{iid}}{\sim} N(0, \tau_\gamma^2), \quad \gamma_{k,t}^i \sim N(\gamma_{k,t-1}^i, \sigma_\gamma^2), \quad \mathbf{Z}_1^i \stackrel{\text{iid}}{\sim} N(0, \tau_z^2 I_d), \quad \mathbf{Z}_t^i \sim N(\mathbf{Z}_{t-1}^i, \sigma_z^2 I_d),$$

and the priors on the remaining parameters are left unchanged from the undirected case. In this case, $\delta_{k,t}^i, \gamma_{k,t}^i \in \mathbb{R}$ model degree heterogeneity in outgoing and incoming edges, respectively. The asymmetric latent positions $\mathbf{X}_t^i, \mathbf{Z}_t^i \in \mathbb{R}^d$ allow an actor's features to differ depending on whether they are receiving or initiating the relation. The variational inference algorithm for this model remains mostly unchanged. However, a model that does not drastically increase the number of parameters compared to the undirected case, such as the one in Sewell and Chen (2015b), is an area of research interest.

Further research directions include increasing the algorithm's scalability through stochastic variational inference (Aliverti & Russo, 2022; Hoffman et al., 2013) and exploring the variational estimates' asymptotics. Overall, our proposed eigenmodel for dynamic multilayer networks is an interpretable statistical network model with applications to various real-world scientific problems.

4.7 Supplementary Material

Python-package multidynet: Python-package `multidynet` contains code to run the CAVI algorithm to approximate the eigenmodel for dynamic multilayer network's posterior and all datasets and scripts used to produce the results in this chapter. This package is available at <https://github.com/joshloyal/multidynet>.

4.7.1 Proofs of Propositions 4.1, 4.2, and 4.3

This section demonstrates the identifiability of our model under the conditions proposed in Propositions 4.1, 4.2, and 4.3. Before stating the proofs, we need the following lemma.

Lemma 4.1. *For any $\mathbf{v} = (v_1, \dots, v_n)^\top \in \mathbb{R}^n$, if $\mathbf{v} \mathbf{1}_n^\top \mathbf{1}_n + \mathbf{1}_n \mathbf{v}^\top \mathbf{1}_n = 0$, then $\mathbf{v} = 0$.*

Proof. The condition can be written as

$$n \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} + \begin{pmatrix} \sum_{i=1}^n v_i \\ \vdots \\ \sum_{i=1}^n v_i \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix},$$

which implies $v_1 = \dots = v_n = -(1/n) \sum_{i=1}^n v_i$. Thus, we have $\mathbf{v} = 0$. $\square$

Proof of Proposition 4.1. We begin by showing that under Assumption A4.1, the social trajectories $\delta_{k,t}$ are identifiable. Under Assumption A4.1, $J_n \mathcal{X}_t = \mathcal{X}_t$ and $J_n \tilde{\mathcal{X}}_t = \tilde{\mathcal{X}}_t$, which implies that $\mathcal{X}_t \Lambda_k \mathcal{X}_t^\top \mathbf{1}_n = \tilde{\mathcal{X}}_t \Lambda_k \tilde{\mathcal{X}}_t^\top \mathbf{1}_n = 0$. Now assume two sets of parameters satisfy

$$\delta_{k,t} \mathbf{1}_n^\top + \mathbf{1}_n \delta_{k,t}^\top + \mathcal{X}_t \Lambda_k \mathcal{X}_t^\top = \tilde{\delta}_{k,t} \mathbf{1}_n^\top + \mathbf{1}_n \tilde{\delta}_{k,t}^\top + \tilde{\mathcal{X}}_t \tilde{\Lambda}_k \tilde{\mathcal{X}}_t^\top \quad (4.7)$$

for $k = 1, \dots, K$ and $t = 1, \dots, T$. Right multiplying $\mathbf{1}_n$ on both sides of the above equation gives

$$\delta_{k,t} \mathbf{1}_n^T \mathbf{1}_n + \mathbf{1}_n \delta_{k,t}^T \mathbf{1}_n = \tilde{\delta}_{k,t} \mathbf{1}_n^T \mathbf{1}_n + \mathbf{1}_n \tilde{\delta}_{k,t}^T \mathbf{1}_n,$$

or

$$(\delta_{k,t} - \tilde{\delta}_{k,t}) \mathbf{1}_n^T \mathbf{1}_n + \mathbf{1}_n (\delta_{k,t} - \tilde{\delta}_{k,t})^T \mathbf{1}_n = 0.$$

Applying Lemma 4.1, we conclude that

$$\delta_{k,t} = \tilde{\delta}_{k,t} \quad (4.8)$$

for all k and t .

Now, we focus on the identifiability of the latent space and the homophily coefficients. By Assumption A4.3, for the reference layer r , Equation (4.7) and Equation (4.8) imply

$$\mathcal{X}_t I_{p,q} \mathcal{X}_t^T = \tilde{\mathcal{X}}_t I_{p',q'} \tilde{\mathcal{X}}_t^T. \quad (4.9)$$

By Assumption A4.2, $\mathcal{X}_t$ and $\tilde{\mathcal{X}}_t$ are full rank so have left inverses B and $\tilde{B}$, respectively. In other words, $B\mathcal{X}_t = \tilde{B}\tilde{\mathcal{X}}_t = I_d$. Multiplying Equation (4.9) on the right by $\tilde{B}^T I_{p',q'}$, we have that

$$\tilde{\mathcal{X}}_t = \mathcal{X}_t I_{p,q} \mathcal{X}_t^T \tilde{B}^T I_{p',q'} = \mathcal{X}_t M_t, \quad (4.10)$$

where $M_t = I_{p,q} \mathcal{X}_t^T \tilde{B}^T I_{p',q'} \in \mathbb{R}^{d \times d}$. More generally, for all layers $k \in \{1, \dots, K\}$, we have

$$\mathcal{X}_t \Lambda_k \mathcal{X}_t^T = \tilde{\mathcal{X}}_t \tilde{\Lambda}_k \tilde{\mathcal{X}}_t^T = \mathcal{X}_t M_t \tilde{\Lambda}_k M_t^T \mathcal{X}_t^T,$$

where the last equality used the identity in Equation (4.10). Multiplying each side of the previous identity on the left by B and on the right by B^T , we conclude that

$$\Lambda_k = M_t \tilde{\Lambda}_k M_t^T. \quad (4.11)$$

However, we want a transformation that takes Λ_k to $\tilde{\Lambda}_k$. To proceed, we note that M_t is invertable. Indeed, since Equation (4.11) holds for the reference layer, we conclude that $M_t I_{p',q'} M_t^T = I_{p,q}$. It is then easy to check that $M_t^{-1} = I_{p',q'} M_t^T I_{p,q}$. Therefore, $(M_t^T)^{-1} = (M_t^{-1})^T = I_{p,q} M_t I_{p',q'}$. Multiplying Equation (4.11) on the left by M_t^{-1} and on the right by $(M_t^T)^{-1}$, we find that

$$\tilde{\Lambda}_k = [I_{p',q'} M_t^T I_{p,q}] \Lambda_k [I_{p,q} M_t I_{p',q'}].$$

Lastly, multiplying on the left and the right by $I_{p',q'}$ and noting that $I_{p',q'} \tilde{\Lambda}_k I_{p',q'} = \tilde{\Lambda}_k$ and $I_{p,q} \Lambda_k I_{p,q} = \Lambda_k$, we find that

$$\tilde{\Lambda}_k = M_t^T \Lambda_k M_t,$$

which completes the proof. $\square$

Proof of Proposition 4.2. From Proposition 4.1, we have that each matrix M_t satisfies $M_t I_{p',q'} M_t^T = I_{p,q}$ for $1 \leq t \leq T$. Consider a single matrix $M \in \{M_t\}_{t=1}^T$. Taking the determinant of both sides of $M I_{p',q'} M^T = I_{p,q}$, we conclude that $\det(M)^2 (-1)^{q'} = (-1)^q$, so that $q - q'$ is an even number. Without loss of generality, assume that $q \geq q'$. We proceed case by case:

- (i) $d = 1$. Since $q - q'$ can only equal zero, the result is immediate.
- (ii) $d = 2$. In this case, the only non-trivial case is $q - q' = 2$, which corresponds to $q = 2, q' = 0$. Now, we show that this combination leads to a contradiction. In this case, M satisfies $MM^T = -I_2$, which is a contradiction because MM^T is a positive-definite matrix while $-I_2$ is not. Thus, $q = q'$.
- (iii) $d = 3$. Once again, the only non-trivial case is $q - q' = 2$, where we have the following two cases: $q = 3, q' = 1$ and $q = 2, q' = 0$. We proceed by showing that both scenarios lead to a contradiction.

For the case $q = 3, q' = 1$, we have $MI_{2,1}M^T = -I_3$ which implies $-M^T M = I_{2,1}$, where we used the fact that $M^T I_{p,q} M = I_{p',q'}$ shown during the proof of Proposition 4.1. Letting $\mathbf{m}_j$ be the j th column of M , we have that

$$-\|\mathbf{m}_1\|_2^2 = 1,$$

which cannot be satisfied by a real vector.

Similarly for $q = 2, q' = 0$, we have $MM^T = I_{1,2}$. Letting $\tilde{\mathbf{m}}_j$ be the j th row vector of M , we have that

$$\|\tilde{\mathbf{m}}_2\|_2^2 = -1.$$

which is impossible for a real vector.

Therefore, $I_{p,q} = I_{p',q'}$ when $1 \leq d \leq 3$, which completes the proof. $\square$

Proof of Proposition 4.3. Without loss of generality, consider a single matrix $M \in \{M_t\}_{t=1}^T$. Further let $\Lambda_k = \text{diag}(\boldsymbol{\lambda}_k)$ and $\tilde{\Lambda}_k = \text{diag}(\tilde{\boldsymbol{\lambda}}_k)$, so that by Proposition 4.1 we have that

$$\text{diag}(\tilde{\boldsymbol{\lambda}}_k) = M^T \text{diag}(\boldsymbol{\lambda}_k) M. \quad (4.12)$$

Now, left multiplying $MI_{p',q'}$ on both sides of Equation (4.12) and apply the identity $MI_{p',q'}M^T = I_{p,q}$, we have that

$$M \text{diag}(\tilde{\boldsymbol{\lambda}}_k) I_{p',q'} = \text{diag}(\boldsymbol{\lambda}_k) I_{p,q} M, \quad (4.13)$$

Denoting the j th columns of M by $\mathbf{m}_j \in \mathbb{R}^d$, we can re-express the linear system in Equation (4.13) as

$$\left(\tilde{\lambda}_{k,j} (I_{p',q'})_{jj} I_d - \text{diag}(\boldsymbol{\lambda}_k) I_{p,q} \right) \mathbf{m}_j = \mathbf{0}_d \quad \text{for } j = 1, \dots, d, \quad (4.14)$$

where $\mathbf{0}_d$ is a d -dimensional vector of zeros.

Now, we determine what relationship Equation (4.14) imposes on $\text{diag}(\boldsymbol{\lambda}_k)$ and $\text{diag}(\tilde{\boldsymbol{\lambda}}_k)$. Let $A_j = \tilde{\lambda}_{j,k} (I_{p',q'})_{jj} I_d - \text{diag}(\boldsymbol{\lambda}_k) I_{p,q}$ for $j = 1, \dots, d$. Since M is full rank, A_j must be a singular matrix for $j = 1, \dots, d$. Combining the facts that $\text{diag}(\tilde{\boldsymbol{\lambda}}_k) I_{p',q'}$ and $\text{diag}(\boldsymbol{\lambda}_k) I_{p,q}$ are both full rank with d distinct elements and that there are d singular diagonal matrices A_j , it is easy to see that $\text{diag}(\boldsymbol{\lambda}_k) I_{p,q} = P \text{diag}(\tilde{\boldsymbol{\lambda}}_k) I_{p',q'} P^T$ for some permutation matrix P . In other words, $\text{diag}(\boldsymbol{\lambda}_k) I_{p,q}$ equals $\text{diag}(\tilde{\boldsymbol{\lambda}}_k) I_{p',q'}$ with permuted diagonal entries.

Now we focus on the consequences for M . As a result of the argument in the previous paragraph, each A_j is a rank $d - 1$ diagonal matrix. This means that each A_j is a diagonal matrix with $d - 1$ non-zero entries and a single zero entry on the diagonal. Therefore, Equation (4.14) holds if and only if $\{\mathbf{m}_j\}_{j=1}^d$ are d -dimensional vectors with a single non-zero entry where A_j is zero on the diagonal. Also, since M is

full rank, $\{\mathbf{m}_1, \dots, \mathbf{m}_d\}$ are linearly independent. This implies that M is a generalized permutation matrix: $M = P \text{diag}(\mathbf{s})$ where $\text{diag}(\mathbf{s})$ is a full-rank diagonal matrix and P is a permutation matrix.

To complete the proof, we focus on the diagonal entries of $MI_{p',q'}M^\top = I_{p,q}$. From the previous paragraph, we have that $MI_{p',q'}M^\top = P \text{diag}(\mathbf{s})I_{p',q'} \text{diag}(\mathbf{s})P^\top$. Let $\sigma : \{1, \dots, d\} \rightarrow \{1, \dots, d\}$, denote the permutation encoded by P , so that

$$P = \begin{pmatrix} \mathbf{e}_{\sigma(1)}^\top \\ \vdots \\ \mathbf{e}_{\sigma(d)}^\top \end{pmatrix},$$

where $\mathbf{e}_1, \dots, \mathbf{e}_d$ are the standard basis vectors. Thus, the diagonal entries must satisfy

$$(I_{p',q'})_{\sigma(j)\sigma(j)} s_{\sigma(j)}^2 = (I_{p,q})_{jj} \text{ for } j = 1, \dots, d,$$

which holds if and only if $\mathbf{s} = \{\pm 1\}^d$, $I_{p,q} = I_{p',q'}$, and σ only permutes the first p and last q diagonal elements of $I_{p,q}$. $\square$

4.7.2 Derivation of Variational Updates

This section contains detailed derivations of the variational updates presented in Section 4.3 of the main text. For notational simplicity, we use the shorthand $\mathbb{E}_{q(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega})}[\cdot] = \langle \cdot \rangle$, where $q(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega})$ is defined in Equation (4.2) of the main text, to denote expectations with respect to the full variational posterior throughout this section. For a definition of the notation used in this section, see Algorithm 4.1.

Throughout this section, we encounter the following Gaussian state space model

$$\mathbf{x}_1 \sim N(0, \tau^2 I_d), \quad (4.15)$$

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{w}_t, \quad \mathbf{w}_t \sim N(0, \sigma^2 I_d), \quad (4.16)$$

$$\mathbf{y}_t = A_t \mathbf{x}_t + \mathbf{b}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim N(0, C_t), \quad (4.17)$$

where $\mathbf{x}_t \in \mathbb{R}^d$, $\mathbf{y}_t \in \mathbb{R}^n$, $A_t \in \mathbb{R}^{n \times d}$, $\mathbf{b}_t \in \mathbb{R}^n$, $C_t \in \mathbb{R}^{n \times n}$. In this context, d is not necessarily the dimension of the latent space and n is not necessarily the number of nodes in the network. Specifically, the full conditional distributions of the social and latent trajectories are of this form. Before proceeding, we state a lemma used throughout Section 4.7.2 and Section 4.7.3.

Lemma 4.2. *For the Gaussian state space model specified by Equations (4.15) – (4.17), the conditional distribution $p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T})$ is in the exponential family with natural parameters*

$$\psi = (-1/2\tau^2, -1/2\sigma^2, \Gamma_{1:T}^1, \Gamma_{1:T}^2/2), \quad (4.18)$$

where $\Gamma_t^1 = A_t^\top C_t^{-1} \mathbf{y}_t - A_t C_t^{-1} \mathbf{b}_t$ and $\Gamma_t^2 = A_t^\top C_t^{-1} A_t$ for $1 \leq t \leq T$.

Proof. We have

$$\begin{aligned} \log p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T}) &\propto -\frac{1}{2\tau^2} \|\mathbf{x}_1\|_2^2 - \frac{1}{2\sigma^2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2 - \\ &\quad \frac{1}{2} \sum_{t=1}^T (\mathbf{y}_t - A_t \mathbf{x}_t - \mathbf{b}_t)^\top C_t^{-1} (\mathbf{y}_t - A_t \mathbf{x}_t - \mathbf{b}_t), \end{aligned}$$

Update $q(\omega_{ijt}^k) = \text{PG}(1, c_{ijt}^k)$:

For each $k \in \{1, \dots, K\}$, $t \in \{1, \dots, T\}$, and $(i, j) \in \{(i, j) : 1 \leq i \leq n, j < i\}$:

$$c_{ijt}^k = (\sigma_{\delta_{k,t}^i}^2 + \mu_{\delta_{k,t}^i}^2 + \sigma_{\delta_{k,t}^j}^2 + \mu_{\delta_{k,t}^j}^2 + 2\mu_{\delta_{k,t}^i} \mu_{\delta_{k,t}^j} + 2(\mu_{\delta_{k,t}^i} + \mu_{\delta_{k,t}^j}) \boldsymbol{\mu}_t^{i\top} \text{diag}(\boldsymbol{\mu}_{\lambda_k}) \boldsymbol{\mu}_t^j + \|\boldsymbol{\Sigma}_{\lambda_k} + \boldsymbol{\mu}_{\lambda_k} \boldsymbol{\mu}_{\lambda_k}^\top \odot (\boldsymbol{\Sigma}_t^i + \boldsymbol{\mu}_t^i \boldsymbol{\mu}_t^{i\top}) \odot (\boldsymbol{\Sigma}_t^j + \boldsymbol{\mu}_t^j \boldsymbol{\mu}_t^{j\top})\|)^{1/2}, \quad (4.19)$$

$$\mu_{\omega_{ijt}^k} = \frac{1}{2c_{ijt}^k} \left(\frac{e^{c_{ijt}^k} - 1}{1 + e^{c_{ijt}^k}} \right). \quad (4.20)$$

Algorithm 4.2: Coordinate ascent updates for the auxiliary Pólya-gamma variables. Here, $\odot$ is the Hadamard product between two matrices, i.e., $(A \odot B)_{ij} = A_{ij} B_{ij}$, and $\|A\| = \sum_i \sum_j A_{ij}$.

$$\begin{aligned} & \propto -\frac{1}{2\tau^2} \|\mathbf{x}_1\|_2^2 - \frac{1}{2\sigma^2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2 + \\ & (A_t^\top C_t^{-1} \mathbf{y}_t - A_t C_t^{-1} \mathbf{b}_t)^\top \mathbf{x}_t - \frac{1}{2} \sum_{t=1}^T \text{tr}(A_t^\top C_t^{-1} A_t \mathbf{x}_t \mathbf{x}_t^\top), \end{aligned}$$

which is in exponential family form with natural parameters given in Equation (4.18). $\square$

The variational distributions of the social and latent trajectories—Equation (4.3) and Equation (4.4) in the main text—are GSSMs that are in the form assumed by Lemma 4.2. This observation implies that the expected natural parameters, $\mathbb{E}_{q(\mathbf{x}_{1:T})} [\psi]$, are sufficient for calculating the variational distribution’s moments, i.e., $\mathbb{E}_{q(\mathbf{x}_{1:T})} [\mathbf{x}_t]$, $\mathbb{E}_{q(\mathbf{x}_{1:T})} [\mathbf{x}_t \mathbf{x}_t^\top]$, and $\mathbb{E}_{q(\mathbf{x}_{1:T})} [\mathbf{x}_t \mathbf{x}_{t+1}^\top]$. In Section 4.7.3, we derive a variational Kalman smoother that calculates these moments recursively.

Derivation of Algorithm 4.2

The coordinate updates for $q(\omega_{ijt}^k)$ are given in Algorithm 4.2, which we formally derive in the remainder of this section.

Proposition 4.4. *Under the eigenmodel for dynamic multilayer networks with the same prior distributions and variational factorization defined in the main text, $q(\omega_{ijt}^k) = \text{PG}(1, c_{ijt}^k)$ where c_{ijt}^k is given in Equation (4.19). Furthermore, the mean of this distribution is given by Equation (4.20).*

Proof. From the exponential tilting property of the Pólya-gamma distribution, we have that

$$\omega_{ijt}^k \mid \cdot \sim \text{PG}(1, \psi_{ijt}^k), \quad (4.21)$$

where $\psi_{ijt}^k = \delta_{k,t}^i + \delta_{k,t}^j + \mathbf{X}_t^{i\top} \Lambda_k \mathbf{X}_t^j$. This distribution is in the exponential family with natural parameter $-(\psi_{ijt}^k)^2/2$. The variational distribution is then a $\text{PG}(1, c_{ijt}^k)$ where $(c_{ijt}^k)^2 = \mathbb{E}_{-q(\omega_{ijt}^k)} [(\psi_{ijt}^k)^2]$.

It remains to calculate the natural parameter c_{ijt}^k . We have that

$$\begin{aligned} (c_{ijt}^k)^2 &= \mathbb{E}_{-q(\omega_{ijt}^k)} [(\psi_{ijt}^k)^2] = \langle (\delta_{k,t}^i + \delta_{k,t}^j + \mathbf{X}_t^{i\top} \Lambda_k \mathbf{X}_t^j)^2 \rangle, \\ &= \langle (\delta_{k,t}^i + \delta_{k,t}^j)^2 \rangle + 2\langle \delta_{k,t}^i + \delta_{k,t}^j \rangle \langle \mathbf{X}_t^i \rangle^\top \langle \Lambda_k \rangle \langle \mathbf{X}_t^j \rangle + \langle (\mathbf{X}_t^{i\top} \Lambda_k \mathbf{X}_t^j)^2 \rangle, \end{aligned}$$

$$\begin{aligned}
&= \sigma_{\delta_{k,t}^i}^2 + \mu_{\delta_{k,t}^i}^2 + \sigma_{\delta_{k,t}^j}^2 + \mu_{\delta_{k,t}^j}^2 + 2\mu_{\delta_{k,t}^i} \mu_{\delta_{k,t}^j} + \\
&\quad 2(\mu_{\delta_{k,t}^i} + \mu_{\delta_{k,t}^j}) \boldsymbol{\mu}_t^{i\top} \text{diag}(\boldsymbol{\mu}_{\boldsymbol{\lambda}_k}) \boldsymbol{\mu}_t^j + \mathbb{E}_{-q(\omega_{ijt}^k)} \left[(\mathbf{X}_t^{i\top} \Lambda_k \mathbf{X}_t^j)^2 \right].
\end{aligned}$$

Note that the last term is equal to

$$\begin{aligned}
\mathbb{E}_{-q(\omega_{ijt}^k)} \left[(\mathbf{X}_t^{i\top} \Lambda_k \mathbf{X}_t^j)^2 \right] &= \mathbb{E}_{-q(\omega_{ijt}^k)} \left[\sum_{g=1}^d \sum_{h=1}^d \lambda_g^k \lambda_h^k X_{tg}^i X_{th}^i X_{tg}^j X_{th}^j \right], \\
&= \sum_{g=1}^d \sum_{h=1}^d \mathbb{E}_{q(\boldsymbol{\lambda}_k)} [\lambda_g^k \lambda_h^k] \mathbb{E}_{q(\mathbf{X}_t^i)} [X_{tg}^i X_{th}^i] \mathbb{E}_{q(\mathbf{X}_t^j)} [X_{tg}^j X_{th}^j], \\
&= \|\mathbb{E}_{q(\boldsymbol{\lambda}_k)} [\boldsymbol{\lambda}_k \boldsymbol{\lambda}_k^\top] \odot \mathbb{E}_{q(\mathbf{X}_t^i)} [\mathbf{X}_t^i \mathbf{X}_t^{i\top}] \odot \mathbb{E}_{q(\mathbf{X}_t^j)} [\mathbf{X}_t^j \mathbf{X}_t^{j\top}]\|, \\
&= \|(\Sigma_{\boldsymbol{\lambda}_k} + \boldsymbol{\mu}_{\boldsymbol{\lambda}_k} \boldsymbol{\mu}_{\boldsymbol{\lambda}_k}^\top) \odot (\Sigma_t^i + \boldsymbol{\mu}_t^i \boldsymbol{\mu}_t^{i\top}) \odot (\Sigma_t^j + \boldsymbol{\mu}_t^j \boldsymbol{\mu}_t^{j\top})\|,
\end{aligned}$$

where $\odot$ is the Hadamard product, i.e., $(A \odot B)_{ij} = A_{ij} B_{ij}$, and $\|A\| = \sum_i \sum_j A_{ij}$.

Lastly, the moments of the Pólya-gamma distribution are available in closed form. In particular, we have that

$$\mu_{\omega_{ijt}^k} = \mathbb{E}_{q(\omega_{ijt}^k)} [\omega_{ijt}^k] = \frac{1}{2c_{ijt}^k} \left(\frac{e^{c_{ijt}^k} - 1}{1 + e^{c_{ijt}^k}} \right).$$

□

Derivation of Algorithm 4.3

The coordinate updates for $q(\delta_{1:T}^i)$, $q(\tau_\delta^2)$, and $q(\sigma_\delta^2)$ are given in Algorithm 4.3, which we formally derive in the remainder of this section.

Proposition 4.5. *Under the eigenmodel for dynamic multilayer networks with the same prior distributions and variational factorization defined in the main text, $q(\tau_\delta^2) = \Gamma^{-1}(\bar{a}_{\tau_\delta^2}/2, \bar{b}_{\tau_\delta^2}/2)$ where $\bar{a}_{\tau_\delta^2}$ and $\bar{b}_{\tau_\delta^2}$ are defined in Equation (4.26) and Equation (4.27), respectively.*

Proof. Standard calculations show that

$$\begin{aligned}
p(\tau_\delta^2 \mid \cdot) &\propto \left(\frac{1}{\tau_\delta^2} \right)^{(a_{\tau_\delta^2} + nK)/2} \exp \left(-\frac{1}{2\tau_\delta^2} \sum_{k=1}^K \sum_{i=1}^n (\delta_{k,1}^i)^2 - \frac{b_{\tau_\delta^2}}{2\tau_\delta^2} \right), \\
&\propto \Gamma^{-1} \left(\frac{a_{\tau_\delta^2} + nK}{2}, \frac{1}{2} \left\{ \sum_{k=1}^K \sum_{i=1}^n (\delta_{k,1}^i)^2 + b_{\tau_\delta^2} \right\} \right).
\end{aligned}$$

Note that $\Gamma^{-1}(a/2, b/2)$ is in the exponential family with natural parameters a and b . Thus, the variational distribution is also an inverse-gamma distribution with natural parameters

$$\begin{aligned}
\bar{a}_{\tau_\delta^2} &= a_{\tau_\delta^2} + nK, \\
\bar{b}_{\tau_\delta^2} &= b_{\tau_\delta^2} + \sum_{k=1}^K \sum_{i=1}^n \mathbb{E}_{q(\delta_{k,1:T}^i)} [(\delta_{k,1}^i)^2], \\
&= b_{\tau_\delta^2} + \sum_{k=1}^K \sum_{i=1}^n \left(\sigma_{\delta_{k,1}^i}^2 + \mu_{\delta_{k,1}^i}^2 \right).
\end{aligned}$$

1. Update $q(\delta_{k,1:T}^i)$, a linear Gaussian state space model (GSSM):

For each $k \in \{1, \dots, K\}$ and $i \in \{1, \dots, n\}$:

- (a) For $t \in \{1, \dots, T\}$, update the natural parameters of the GSSM:

$$\Gamma_t^1 = \sum_{j \neq i} [Y_{ijt}^k - 1/2 - \mu_{\omega_{ijt}^k} (\mu_{\delta_{k,t}^j} + \boldsymbol{\mu}_t^{i^T} \text{diag}(\boldsymbol{\mu}_{\boldsymbol{\lambda}_k}) \boldsymbol{\mu}_t^j)], \quad (4.22)$$

$$\Gamma_t^2 = \sum_{j \neq i} \mu_{\omega_{ijt}^k}, \quad (4.23)$$

$$\langle 1/\tau_\delta^2 \rangle = \bar{a}_{\tau_\delta^2} / \bar{b}_{\tau_\delta^2}, \quad (4.24)$$

$$\langle 1/\sigma_\delta^2 \rangle = \bar{c}_{\sigma_\delta^2} / \bar{d}_{\sigma_\delta^2}. \quad (4.25)$$

- (b) Update marginal distributions and cross-covariances as in Algorithm 4.7:

$$\mu_{\delta_{k,1:T}^i}, \sigma_{\delta_{k,1:T}^i}^2, \{\sigma_{\delta_{k,t,t+1}^i}^2\}_{t=1}^{T-1} = \text{KalmanSmoother}(\Gamma_{1:T}^1, \Gamma_{1:T}^2, \bar{a}_{\tau_\delta^2} / \bar{b}_{\tau_\delta^2}, \bar{c}_{\sigma_\delta^2} / \bar{d}_{\sigma_\delta^2}).$$

2. Update $q(\tau_\delta^2) = \Gamma^{-1}(\bar{a}_{\tau_\delta^2}/2, \bar{b}_{\tau_\delta^2}/2)$:

$$\bar{a}_{\tau_\delta^2} = a_{\tau_\delta^2} + nK, \quad (4.26)$$

$$\bar{b}_{\tau_\delta^2} = b_{\tau_\delta^2} + \sum_{k=1}^K \sum_{i=1}^n \left(\sigma_{\delta_{k,1}^i}^2 + \mu_{\delta_{k,1}^i}^2 \right). \quad (4.27)$$

3. Update $q(\sigma_\delta^2) = \Gamma^{-1}(\bar{c}_{\sigma_\delta^2}/2, \bar{d}_{\sigma_\delta^2}/2)$:

$$\bar{c}_{\sigma_\delta^2} = c_{\sigma_\delta^2} + nK(T-1), \quad (4.28)$$

$$\bar{d}_{\sigma_\delta^2} = d_{\sigma_\delta^2} + \sum_{k=1}^K \sum_{t=2}^T \sum_{i=1}^n \left\{ \sigma_{\delta_{k,t}^i}^2 + \mu_{\delta_{k,t}^i}^2 + \sigma_{\delta_{k,t-1}^i}^2 + \mu_{\delta_{k,t-1}^i}^2 - 2(\sigma_{\delta_{k,t-1,t}^i}^2 + \mu_{\delta_{k,t-1}^i} \mu_{\delta_{k,t}^i}) \right\}. \quad (4.29)$$

Algorithm 4.3: Coordinate ascent updates for the social trajectories. `KalmanSmoother` is the variational Kalman smoother defined in Algorithm 4.7 of Section 4.7.3.

□

Proposition 4.6. *Under the eigenmodel for dynamic multilayer networks with the same prior distributions and variational factorization defined in the main text, $q(\sigma_\delta^2) = \Gamma^{-1}(\bar{c}_{\tau_\delta^2}/2, \bar{d}_{\tau_\delta^2}/2)$ where $\bar{c}_{\tau_\delta^2}$ and $\bar{d}_{\tau_\delta^2}$ are defined in Equation (4.28) and Equation (4.29), respectively.*

Proof. Standard calculations show that

$$\begin{aligned} p(\sigma_\delta^2 | \cdot) &\propto \left(\frac{1}{\sigma_\delta^2}\right)^{(c_{\sigma_\delta^2} + nK(T-1))/2} \exp\left(-\frac{1}{2\sigma_\delta^2} \sum_{k=1}^K \sum_{t=2}^T \sum_{i=1}^n (\delta_{k,t}^i - \delta_{k,t-1}^i)^2 - \frac{d_{\sigma_\delta^2}}{2\sigma_\delta^2}\right), \\ &\propto \Gamma^{-1}\left(\frac{c_{\sigma_\delta^2} + nK(T-1)}{2}, \frac{1}{2} \left\{ \sum_{k=1}^K \sum_{t=2}^T \sum_{i=1}^n (\delta_{k,t}^i - \delta_{k,t-1}^i)^2 + d_{\sigma_\delta^2} \right\}\right). \end{aligned}$$

Note that $\Gamma^{-1}(a/2, b/2)$ is in the exponential family with natural parameters a and b . Thus, the variational distribution is also an inverse-gamma distribution with natural parameters

$$\begin{aligned} \bar{c}_{\sigma_\delta^2} &= c_{\sigma_\delta^2} + nK(T-1), \\ \bar{d}_{\sigma_\delta^2} &= d_{\sigma_\delta^2} + \sum_{k=1}^K \sum_{t=2}^T \sum_{i=1}^n \mathbb{E}_{q(\delta_{k,1:T}^i)} [(\delta_{k,t}^i - \delta_{k,t-1}^i)^2], \\ &= d_{\sigma_\delta^2} + \sum_{k=1}^K \sum_{t=2}^T \sum_{i=1}^n \left\{ \sigma_{\delta_{k,t}^i}^2 + \mu_{\delta_{k,t}^i}^2 + \sigma_{\delta_{k,t-1}^i}^2 + \mu_{\delta_{k,t-1}^i}^2 - 2(\sigma_{\delta_{k,t-1,t}^i}^2 + \mu_{\delta_{k,t}^i} \mu_{\delta_{k,t-1}^i}) \right\}. \end{aligned}$$

□

Proposition 4.7. *Under the eigenmodel for dynamic multilayer networks with the same prior distributions and variational factorization defined in the main text, $q(\delta_{k,1:T}^i)$ is a Gaussian state space model with natural parameters given by Equations (4.22) – (4.25).*

Proof. From Equation (4.3), we associate $p(\delta_{k,1:T}^i | \cdot)$ with a GSSM parameterized by

$$\begin{aligned} \mathbf{x}_t &= \delta_{k,t}^i \in \mathbb{R}, \\ \mathbf{y}_t &= \mathbf{z}_{k,t}^i \in \mathbb{R}^{n-1}, \\ A_t &= (\omega_{i1t}^k, \dots, \omega_{i(i-1)t}^k, \omega_{i(i+1)t}^k, \dots, \omega_{int}^k) \in \mathbb{R}^{n-1}, \\ (\mathbf{b}_t)_j &= \omega_{ijt}^k (\delta_{k,t}^j + \mathbf{X}_t^{jT} \Lambda_k \mathbf{X}_t^i), \quad \text{for } j \neq i, \\ C_t &= \text{diag}(\omega_{i1t}^k, \dots, \omega_{i(i-1)t}^k, \omega_{i(i+1)t}^k, \dots, \omega_{int}^k). \end{aligned}$$

We then apply Lemma 4.2 in Section 4.7.3 to identify the natural parameters. Note that $A_t^T C_t^{-1} = \mathbf{1}_{n-1}$, so that expected natural parameters are

$$\begin{aligned} \Gamma_t^1 &= \langle \mathbf{1}_{n-1}^T (\mathbf{y}_t - \mathbf{b}_t) \rangle = \sum_{j \neq i} (Y_{ijt}^k - 1/2 - \mu_{\omega_{ijt}^k} (\mu_{\delta_{k,t}^j} + \boldsymbol{\mu}_t^{jT} \text{diag}(\boldsymbol{\mu}_{\lambda_k}) \boldsymbol{\mu}_t^i)), \\ \Gamma_t^2 &= \langle \mathbf{1}_{n-1}^T A_t \rangle = \sum_{j \neq i} \mu_{\omega_{ijt}^k}. \end{aligned}$$

Furthermore, from Proposition 4.5 and Proposition 4.6, we have that $1/\tau_\delta^2$ and $1/\sigma_\delta^2$ are gamma distributed with means $\bar{a}_{\tau_\delta^2}/\bar{b}_{\tau_\delta^2}$ and $\bar{c}_{\sigma_\delta^2}/\bar{d}_{\sigma_\delta^2}$, respectively. Finally, we can apply the variational Kalman smoothing

equations derived in Section 4.7.3 to calculate the moments of $q(\delta_{k,1:T}^i)$. $\square$

Derivation of Algorithm 4.4

The coordinate updates for $q(\mathbf{X}_{1:T}^i)$, $q(\tau^2)$, and $q(\sigma^2)$ are given in Algorithm 4.4, which we formally derive in the remainder of this section.

Proposition 4.8. *Under the eigenmodel for dynamic multilayer networks with the same prior distributions and variational factorization defined in the main text, $q(\tau^2) = \Gamma^{-1}(\bar{a}_{\tau^2}/2, \bar{b}_{\tau^2}/2)$ where $\bar{a}_{\tau^2}$ and $\bar{b}_{\tau^2}$ are defined in Equation (4.34) and Equation (4.35), respectively.*

Proof. Standard calculations show that

$$\begin{aligned} p(\tau^2 \mid \cdot) &\propto \left(\frac{1}{\tau^2}\right)^{(a_{\tau^2} + nd)/2} \exp\left(-\frac{1}{2\tau^2} \sum_{i=1}^n \|\mathbf{X}_1^i\|_2^2 - \frac{b_{\tau^2}}{2\tau^2}\right), \\ &\propto \Gamma^{-1}\left(\frac{a_{\tau^2} + nd}{2}, \frac{1}{2} \left\{ \sum_{i=1}^n \|\mathbf{X}_1^i\|_2^2 + b_{\tau^2} \right\}\right). \end{aligned}$$

Note that $\Gamma^{-1}(a/2, b/2)$ is in the exponential family with natural parameters a and b . Thus, the variational distribution is also an inverse-gamma distribution with natural parameters

$$\begin{aligned} \bar{a}_{\tau^2} &= a_{\tau^2} + nd, \\ \bar{b}_{\tau^2} &= b_{\tau^2} + \sum_{i=1}^n \mathbb{E}_{q(\mathbf{X}_{1:T}^i)} [\|\mathbf{X}_1^i\|_2^2], \\ &= b_{\tau^2} + \sum_{k=1}^K \sum_{i=1}^n (\text{tr}(\Sigma_1^i) + \boldsymbol{\mu}_1^{i\top} \boldsymbol{\mu}_1^i). \end{aligned}$$

$\square$

Proposition 4.9. *Under the eigenmodel for dynamic multilayer networks with the same prior distributions and variational factorization defined in the main text, $q(\sigma^2) = \Gamma^{-1}(\bar{c}_{\sigma^2}/2, \bar{d}_{\sigma^2}/2)$ where $\bar{c}_{\sigma^2}$ and $\bar{d}_{\sigma^2}$ are defined in Equation (4.36) and Equation (4.37), respectively.*

Proof. Standard calculations show that

$$\begin{aligned} p(\sigma^2 \mid \cdot) &\propto \left(\frac{1}{\sigma^2}\right)^{(c_{\sigma^2} + nd(T-1))/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{t=2}^T \sum_{i=1}^n \|\mathbf{X}_t^i - \mathbf{X}_{t-1}^i\|_2^2 - \frac{d_{\sigma^2}}{2\sigma^2}\right), \\ &\propto \Gamma^{-1}\left(\frac{c_{\sigma^2} + nd(T-1)}{2}, \frac{1}{2} \left\{ \sum_{t=2}^T \sum_{i=1}^n \|\mathbf{X}_t^i - \mathbf{X}_{t-1}^i\|_2^2 + d_{\sigma^2} \right\}\right). \end{aligned}$$

Note that $\Gamma^{-1}(a/2, b/2)$ is in the exponential family with natural parameters a and b . Thus, the variational distribution is also an inverse-gamma distribution with natural parameters

$$\begin{aligned} \bar{c}_{\sigma^2} &= c_{\sigma^2} + nd(T-1), \\ \bar{d}_{\sigma^2} &= d_{\sigma^2} + \sum_{t=2}^T \sum_{i=1}^n \mathbb{E}_{q(\mathbf{X}_{1:T}^i)} [\|\mathbf{X}_t^i - \mathbf{X}_{t-1}^i\|_2^2], \end{aligned}$$

1. Update $q(\mathbf{X}_{1:T}^i)$, a linear Gaussian state space model (GSSM):

For $i \in \{1, \dots, n\}$:

- (a) For $t \in \{1, \dots, T\}$, update the natural parameters of the GSSM:

$$\Gamma_t^1 = \begin{pmatrix} \sum_{k=1}^K \sum_{j \neq i} \mu_{\lambda_{k1}} \mu_{t1}^j [Y_{ij}^k - 1/2 - \mu_{\omega_{ij}^k} (\mu_{\delta_{k,t}^i} + \mu_{\delta_{k,t}^j})] \\ \vdots \\ \sum_{k=1}^K \sum_{j \neq i} \mu_{\lambda_{kd}} \mu_{td}^j [Y_{ij}^k - 1/2 - \mu_{\omega_{ij}^k} (\mu_{\delta_{k,t}^i} + \mu_{\delta_{k,t}^j})] \end{pmatrix}, \quad (4.30)$$

$$\Gamma_t^2 = \sum_{k=1}^K \sum_{j \neq i} \mu_{\omega_{ij}^k} (\Sigma_{\lambda_k} + \boldsymbol{\mu}_{\lambda_k} \boldsymbol{\mu}_{\lambda_k}^T) \odot (\Sigma_t^j + \boldsymbol{\mu}_t^j \boldsymbol{\mu}_t^{jT}), \quad (4.31)$$

$$\langle 1/\tau^2 \rangle = a_{\tau^2}/b_{\tau^2}, \quad (4.32)$$

$$\langle 1/\sigma^2 \rangle = \bar{c}_{\tau^2}/\bar{d}_{\tau^2}. \quad (4.33)$$

- (b) Update marginal distributions and cross-covariances as in Algorithm 4.7:

$$\boldsymbol{\mu}_{1:T}^i, \Sigma_{1:T}^i, \{\Sigma_{t,t+1}^i\}_{t=1}^{T-1} = \text{KalmanSmoother}(\Gamma_{1:T}^1, \Gamma_{1:T}^2, \bar{a}_{\tau^2}/\bar{b}_{\tau^2}, \bar{c}_{\sigma^2}/\bar{d}_{\sigma^2}).$$

2. Update $q(\tau^2) = \Gamma^{-1}(\bar{a}_{\tau^2}/2, \bar{b}_{\tau^2}/2)$:

$$\bar{a}_{\tau^2} = a_{\tau^2} + nd, \quad (4.34)$$

$$\bar{b}_{\tau^2} = b_{\tau^2} + \sum_{i=1}^n (\text{tr}(\Sigma_1^i) + \boldsymbol{\mu}_1^{iT} \boldsymbol{\mu}_1^i). \quad (4.35)$$

3. Update $q(\sigma^2) = \Gamma^{-1}(\bar{c}_{\sigma^2}/2, \bar{d}_{\sigma^2}/2)$:

$$\bar{c}_{\sigma^2} = c_{\sigma^2} + nd(T-1), \quad (4.36)$$

$$\begin{aligned} \bar{d}_{\sigma^2} = d_{\sigma^2} + \sum_{t=2}^T \sum_{i=1}^n \Big\{ & \text{tr}(\Sigma_t^i) + \boldsymbol{\mu}_t^{iT} \boldsymbol{\mu}_t^i + \text{tr}(\Sigma_{t-1}^i) + \boldsymbol{\mu}_{t-1}^{iT} \boldsymbol{\mu}_{t-1}^i \\ & - 2(\text{tr}(\Sigma_{t-1,t}^i) + \boldsymbol{\mu}_{t-1}^{iT} \boldsymbol{\mu}_t^i) \Big\}. \end{aligned} \quad (4.37)$$

Algorithm 4.4: Coordinate ascent updates for the latent trajectories. `KalmanSmoother` is the variational Kalman smoother defined in Algorithm 4.7 of Section 4.7.3.

$$\begin{aligned}
&= d_{\sigma^2} + \sum_{t=2}^T \sum_{i=1}^n \left\{ \text{tr}(\Sigma_t^i) + \boldsymbol{\mu}_t^{i\top} \boldsymbol{\mu}_t^i + \text{tr}(\Sigma_{t-1}^i) + \boldsymbol{\mu}_{t-1}^{i\top} \boldsymbol{\mu}_{t-1}^i \right. \\
&\quad \left. - 2(\text{tr}(\Sigma_{t-1,t}^i) + \boldsymbol{\mu}_{t-1}^{i\top} \boldsymbol{\mu}_t^i) \right\}.
\end{aligned}$$

□

Proposition 4.10. *Under the eigenmodel for dynamic multilayer networks with the same prior distributions and variational factorization defined in the main text, $q(\mathbf{X}_{1:T}^i)$ is a Gaussian state space model with natural parameters given in Equations (4.30) – (4.33).*

Proof. First we layout some notation. Let

$$\begin{aligned}
\boldsymbol{\theta}_{k,t}^i &= \delta_{k,t}^i \mathbf{1}_{n-1} + (\delta_{k,t}^1, \dots, \delta_{k,t}^{i-1}, \delta_{k,t}^{i+1}, \dots, \delta_{k,t}^n)^\top \in \mathbb{R}^{n-1}, \\
\boldsymbol{\omega}_{it}^k &= (\omega_{i1t}^k, \dots, \omega_{i(i-1)t}^k, \omega_{i(i+1)t}^k, \dots, \omega_{int}^k)^\top \in \mathbb{R}^{n-1}, \\
X_{k,t}^i &= (\mathbf{X}_t^1 \Lambda_k, \dots, \mathbf{X}_t^{i-1} \Lambda_k, \mathbf{X}_t^{i+1} \Lambda_k, \dots, \mathbf{X}_t^n \Lambda_k)^\top \in \mathbb{R}^{(n-1) \times d}.
\end{aligned}$$

We then define the concatenated version of these quantities:

$$\begin{aligned}
\Omega_t^i &= \text{diag}(\boldsymbol{\omega}_{it}^{1\top}, \dots, \boldsymbol{\omega}_{it}^{K\top}) \in \mathbb{R}^{K(n-1) \times K(n-1)}, \\
\boldsymbol{\theta}_t^i &= (\boldsymbol{\delta}_{1,t}^{i\top}, \dots, \boldsymbol{\delta}_{K,t}^{i\top})^\top \in \mathbb{R}^{K(n-1)},
\end{aligned}$$

and $X_t^i \in \mathbb{R}^{K(n-1) \times d}$ formed by stacking the matrices $X_{k,t}^i$ row-wise for $k = 1, \dots, K$.

From Equation (4.4), we associate $p(\mathbf{X}_{1:T}^i \mid \cdot)$ with a GSSM parameterized by

$$\begin{aligned}
\mathbf{x}_t &= \mathbf{X}_t^i, \\
\mathbf{y}_t &= \mathbf{z}_t^i, \\
A_t &= \Omega_t^i X_t^i, \\
\mathbf{b}_t &= \Omega_t^i \boldsymbol{\theta}_t^i, \\
C_t &= \Omega_t^i.
\end{aligned}$$

We then apply Lemma 4.2 in Section 4.7.3 to identify the natural parameters. Note that $A_t^\top C_t^{-1} = X_t^i$. Taking into account the independence assumptions contained in the approximate posterior, we have

$$\begin{aligned}
\Gamma_t^1 &= \langle X_t^i \rangle^\top (\mathbf{z}_t^i - \langle \Omega_t^i \rangle \langle \boldsymbol{\theta}_t^i \rangle), \\
&= \begin{pmatrix} \sum_{k=1}^K \sum_{j \neq i} \mu_{\lambda_{k1}} \mu_{t1}^j [Y_{ijt}^k - 1/2 - \mu_{\omega_{ijt}^k} (\mu_{\delta_{k,t}^i} + \mu_{\delta_{k,t}^j})] \\ \vdots \\ \sum_{k=1}^K \sum_{j \neq i} \mu_{\lambda_{kd}} \mu_{td}^j [Y_{ijt}^k - 1/2 - \mu_{\omega_{ijt}^k} (\mu_{\delta_{k,t}^i} + \mu_{\delta_{k,t}^j})] \end{pmatrix}.
\end{aligned}$$

Next, the individual elements of $\Gamma_t^2 \in \mathbb{R}^{d \times d}$ are

$$(\Gamma_t^2)_{gh} = \langle X_t^{i\top} \Omega_t^i X_t^i \rangle_{gh} = \sum_{k=1}^K \sum_{j \neq i} \langle \omega_{ijt}^k \rangle \langle \lambda_g^k \lambda_h^k \rangle \langle X_{tg}^j X_{th}^j \rangle,$$

1. Update $q(\lambda_{1h}) = p_{\lambda_{1h}}^{\mathbb{1}_{\{\lambda_{1h}=1\}}} (1 - p_{\lambda_{1h}})^{\mathbb{1}_{\{\lambda_{1h}=-1\}}}$:

For $h \in \{1, \dots, d\}$:

$$\eta_{\lambda_{1h}} = \log \left[\frac{\rho}{1 - \rho} \right] + 2 \sum_{t=1} \sum_{j < i} \left\{ (Y_{ijt}^1 - 1/2 - \mu_{\omega_{ijt}^i} (\mu_{\delta_{1,t}^i} + \mu_{\delta_{1,t}^j}) \mu_{th}^i \mu_{th}^j - \right. \\ \left. \mu_{\omega_{ijt}^1} \sum_{g \neq h} \mu_{\lambda_{1g}} ((\Sigma_t^i)_{gh} + \mu_{tg}^i \mu_{th}^i) ((\Sigma_t^j)_{gh} + \mu_{tg}^j \mu_{th}^j) \right\}, \quad (4.38)$$

$$p_{\lambda_{1h}} = e^{\eta_{\lambda_{1h}}} / (1 + e^{\eta_{\lambda_{1h}}}), \quad \mu_{\lambda_{1h}} = 2p_{\lambda_{1h}} - 1, \quad \sigma_{\lambda_{1h}}^2 = 1 - (2p_{\lambda_{1h}} - 1)^2. \quad (4.39)$$

2. Update $q(\boldsymbol{\lambda}_k) = N(\boldsymbol{\mu}_{\boldsymbol{\lambda}_k}, \Sigma_{\boldsymbol{\lambda}_k})$:

For $k \in \{2, \dots, K\}$:

$$\Sigma_{\boldsymbol{\lambda}_k} = \left[\sum_{t=1}^T \sum_{j < i} \mu_{\omega_{ijt}^k} (\Sigma_t^i + \boldsymbol{\mu}_t^i \boldsymbol{\mu}_t^{i\top}) \odot (\Sigma_t^j + \boldsymbol{\mu}_t^j \boldsymbol{\mu}_t^{j\top}) + \frac{1}{\sigma_{\boldsymbol{\lambda}}^2} I_p \right]^{-1}, \quad (4.40)$$

$$\boldsymbol{\mu}_{\boldsymbol{\lambda}_k} = \Sigma_{\boldsymbol{\lambda}_k} \begin{pmatrix} \sum_{t=1}^T \sum_{j < i} [Y_{ijt}^k - 1/2 - \mu_{\omega_{ijt}^k} (\mu_{\delta_{k,t}^i} + \mu_{\delta_{k,t}^j})] \mu_{t1}^i \mu_{t1}^j \\ \vdots \\ \sum_{t=1}^T \sum_{j < i} [Y_{ijt}^k - 1/2 - \mu_{\omega_{ijt}^k} (\mu_{\delta_{k,t}^i} + \mu_{\delta_{k,t}^j})] \mu_{td}^i \mu_{td}^j \end{pmatrix}. \quad (4.41)$$

Algorithm 4.5: Coordinate ascent updates for the homophily coefficients

or

$$\Gamma_t^2 = \sum_{k=1} \sum_{j \neq i} \mu_{\omega_{ijt}^k} \mathbb{E}_{q(\boldsymbol{\lambda}_k)} [\boldsymbol{\lambda}_k \boldsymbol{\lambda}_k^\top] \odot \mathbb{E}_{q(\mathbf{X}_{1:T}^j)} [\mathbf{X}_t^j \mathbf{X}_t^{j\top}], \\ = \sum_{k=1} \sum_{j \neq i} \mu_{\omega_{ijt}^k} (\Sigma_{\boldsymbol{\lambda}_k} + \boldsymbol{\mu}_{\boldsymbol{\lambda}_k} \boldsymbol{\mu}_{\boldsymbol{\lambda}_k}^\top) \odot (\Sigma_t^j + \boldsymbol{\mu}_t^j \boldsymbol{\mu}_t^{j\top}).$$

Furthermore, from Proposition 4.8 and Proposition 4.9, we have that $1/\tau^2$ and $1/\sigma^2$ are gamma distributed with means $\bar{a}_{\tau^2}/\bar{b}_{\tau^2}$ and $\bar{c}_{\sigma^2}/\bar{d}_{\sigma^2}$, respectively. Finally, we can apply the variational Kalman smoothing equations derived in Section 4.7.3 to calculate the moments of $q(\mathbf{X}_{1:T}^i)$. $\square$

Derivation of Algorithm 4.5

The coordinate updates for $q(\boldsymbol{\lambda}_k)$ are given in Algorithm 4.5, which we formally derive in the remainder of this section.

Proposition 4.11. *Consider the eigenmodel for dynamic multilayer networks with the same prior distributions and variational factorization defined in the main text. For $k \in \{2, \dots, K\}$ (non-reference layers), $q(\boldsymbol{\lambda}_k) = N(\boldsymbol{\mu}_{\boldsymbol{\lambda}_k}, \Sigma_{\boldsymbol{\lambda}_k})$ with parameters given in Equation (4.40) and Equation (4.41).*

Proof. First we define some notation. Let $\mathcal{D} = \{(i, j) : j < i, 1 \leq i \leq n\}$ denote the set of dyads. Define $\mathbf{X}_t^{i \odot j} = \mathbf{X}_t^i \odot \mathbf{X}_t^j$ and $\theta_{k,t}^{ij} = \delta_{k,t}^i + \delta_{k,t}^j$. Let $\boldsymbol{\omega}_t^k = (\omega_{ijt}^k)_{((i,j) \in \mathcal{D})} \in \mathbb{R}^{|\mathcal{D}|}$ be a vector formed by stacking the

ω_{ijt}^k by dyads. Also, let $\Omega_k = \text{diag}(\omega_1^k, \dots, \omega_T^k) \in \mathbb{R}^{|\mathcal{D}|T \times |\mathcal{D}|T}$. Finally, let $\mathbf{z}_k \in \mathbb{R}^{|\mathcal{D}|T}$, $X \in \mathbb{R}^{|\mathcal{D}|T \times d}$ and $\boldsymbol{\theta}_k \in \mathbb{R}^{|\mathcal{D}|T}$ be formed by stacking $z_{ijt}^k = Y_{ijt}^k - 1/2$, $\mathbf{X}_t^{i \odot j}$ and $\theta_{k,t}^{ij}$ first by dyads and then the result by time, respectively. Standard manipulations show that

$$p(\boldsymbol{\lambda}_k | \cdot) \propto p(\boldsymbol{\lambda}_k) N(\mathbf{z}_k | \Omega_k X \boldsymbol{\lambda}_k + \Omega_k \boldsymbol{\theta}_k, \Omega_k).$$

Since $p(\boldsymbol{\lambda}_k) = N(0, \sigma_\lambda^2 I_d)$, the full conditional distribution is Gaussian with the following natural parameters:

$$\Lambda = \left[X^T \Omega_k X + \frac{1}{\sigma_\lambda^2} I_d \right],$$

$$\boldsymbol{\eta} = X^T (\mathbf{z}_k - \Omega_k \boldsymbol{\theta}_k).$$

Taking expectations with respect to the approximate posterior and converting back to the mean and covariance parameters, we have

$$\Sigma_\lambda = \left[\langle X^T \Omega_k X \rangle + \frac{1}{\sigma_\lambda^2} I_d \right]^{-1},$$

$$\boldsymbol{\mu}_{\lambda_k} = \Sigma_{\lambda_k} \langle X \rangle^T (\mathbf{z} - \langle \Omega_k \rangle \langle \boldsymbol{\theta}_k \rangle),$$

which is equivalent to the parameters in Equation (4.40) and Equation (4.41). $\square$

Proposition 4.12. *Consider the eigenmodel for dynamic multilayer networks with the same prior distributions and variational factorization defined in the main text. For $h \in \{1, \dots, d\}$, $q(\lambda_{1h}) = p_{\lambda_{1h}}^{\mathbb{1}_{\{\lambda_{1h}=1\}}}$ $(1 - p_{\lambda_{1h}})^{\mathbb{1}_{\{\lambda_{1h}=-1\}}}$ where $p_{\lambda_{1h}}$ is given in Equation (4.39).*

Proof. The full conditional distributions are

$$p(\lambda_{1h} | \cdot) \propto p(\lambda_{1h}) \exp \left\{ \sum_{t=1}^T \sum_{j < i} \left[(Y_{ijt}^1 - 1/2 - \omega_{ijt}^k (\delta_{1,t}^i + \delta_{1,t}^j)) \lambda_{1h} X_{th}^i X_{th}^j - \right. \right.$$

$$\left. \frac{1}{2} \omega_{ijt}^1 \lambda_{1q}^2 (X_{th}^i)^2 (X_{th}^j)^2 - \omega_{ijt}^1 \sum_{g \neq h} \lambda_{1g} \lambda_{1h} X_{tg}^i X_{th}^i X_{th}^j X_{tg}^j \right] \Big\}. \quad (4.42)$$

The natural parameter is then

$$\eta_{\lambda_{1h}} = \mathbb{E}_{-q(\lambda_{1h})} [\log p(\lambda_{1h} = 1 | \cdot)] - \mathbb{E}_{-q(\lambda_{1h})} [\log p(\lambda_{1h} = -1 | \cdot)],$$

$$= \log \left[\frac{\rho}{1 - \rho} \right] +$$

$$2 \sum_{t=1}^T \sum_{j < i} \left\{ (Y_{ijt}^1 - 1/2 \mu_{\omega_{ijt}^1} (\mu_{\delta_{1,t}^i} + \mu_{\delta_{1,t}^j})) \mu_{th}^i \mu_{th}^j - \right.$$

$$\left. \mu_{\omega_{ijt}^1} \sum_{g \neq h} \mu_{\lambda_{1g}} ((\Sigma_t^i)_{gh} + \mu_{tg}^i \mu_{th}^i) ((\Sigma_t^j)_{gh} + \mu_{tg}^j \mu_{th}^j) \right\}.$$

Converting back to the standard parameterization, we have

$$p_{\lambda_{1h}} = e^{\eta_{\lambda_{1h}}} / (1 + e^{\eta_{\lambda_{1h}}}),$$

$$\mu_{\lambda_{1h}} = 2p_{\lambda_{1h}} - 1,$$

$$\sigma_{\lambda_{1h}}^2 = 1 - (2p_{\lambda_{1h}} - 1)^2.$$

□

4.7.3 Derivation of the Variational Kalman Smoother

In this section, we derive the variational Kalman smoother used for inference in our model. Many of our results are based on the work in Beal (2003). The major difference in the two formulations is that we incorporate time-varying state space parameters and non-identity covariance matrices.

Consider the Gaussian state space model specified by Equations (4.15) – (4.17). Our goal is to perform variational inference based on the factorization $q(\mathbf{x}_{1:T})q(\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ contains the parameters of the GSSM. Crucially, the variational distributions of the hidden states are also GSSMs. Indeed,

$$\begin{aligned} q(\mathbf{x}_{1:T}) &= c \exp(\langle \log p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}) \rangle) \triangleq h(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}) \\ &= h(\mathbf{x}_1) h(\mathbf{y}_1 | \mathbf{x}_1) \prod_{t=2}^T h(\mathbf{x}_t | \mathbf{x}_{t-1}) h(\mathbf{y}_t | \mathbf{x}_t), \end{aligned}$$

where

$$\begin{aligned} h(\mathbf{x}_1) &= c_1 \exp(\langle \log N(\mathbf{x}_1 | 0, \tau^2 I_d) \rangle), \\ h(\mathbf{x}_t | \mathbf{x}_{t-1}) &= c_2 \exp(\langle \log N(\mathbf{x}_t | \mathbf{x}_{t-1}, \sigma^2 I_d) \rangle), \\ h(\mathbf{y}_t | \mathbf{x}_t) &= c_3 \exp(\langle \log N(\mathbf{y}_t | A_t \mathbf{x}_t + \mathbf{b}_t, C_t) \rangle), \end{aligned}$$

are Gaussian distributions and $\langle \cdot \rangle$ denotes an expectation with respect to $q(\boldsymbol{\theta})$. When calculating the distributions in $q(\mathbf{x}_{1:T})$, we use the Gaussian distribution's natural parameter form so that the expectations have an analytical form. The trade-off in using this parameterization is that when calculating the moments of these variational GSSMs, the standard Kalman smoothing equations are no longer applicable because we only have access to the natural parameters. Therefore, we derive Kalman smoothing equations that are expressed in terms of the variational distribution's natural parameters.

Variational Kalman Filter

Property 4.1. *For the Gaussian state space model specified in Equations (4.15) – (4.17), the variational filtering distributions $h(\mathbf{x}_t | \mathbf{y}_{1:t}) = N(\mathbf{x}_t | \boldsymbol{\mu}_t, \Sigma_t)$ with parameters that can be calculated recursively via Algorithm 4.6.*

Proof. Define the forwards message variables as $\alpha_t(\mathbf{x}_t) = h(\mathbf{x}_t | \mathbf{y}_{1:t})$. The filter proceeds recursively starting at $t = 1$:

$$\begin{aligned} \alpha_1(\mathbf{x}_1) &\propto h(\mathbf{x}_1) h(\mathbf{y}_1 | \mathbf{x}_1), \\ &\propto \exp \left(-\frac{1}{2} \left\langle \mathbf{x}_1^T \frac{1}{\tau^2} I_d \mathbf{x}_1 + (\mathbf{y}_1 - A_1 \mathbf{x}_1 - \mathbf{b}_1)^T C_1^{-1} (\mathbf{y}_1 - A_1 \mathbf{x}_1 - \mathbf{b}_1) \right\rangle \right), \\ &\propto \exp \left(-\frac{1}{2} \mathbf{x}_1^T \left(\left\langle \frac{1}{\tau^2} \right\rangle I_d + \langle A_1^T C_1^{-1} A_1 \rangle \right) \mathbf{x}_1 + \right. \end{aligned}$$

Given the natural parameters $\langle \Gamma_{1:T}^1 \rangle = \langle A_{1:T}^T C_{1:T}^{-1} \rangle \mathbf{y}_{1:T} - \langle A_{1:T}^T C_{1:T}^{-1} \mathbf{b}_{1:T} \rangle$, $\langle \Gamma_{1:T}^2 \rangle = \langle A_{1:T}^T C_{1:T}^{-1} A_{1:T} \rangle$, $\langle 1/\tau^2 \rangle$, and $\langle 1/\sigma^2 \rangle$, calculate the filtering distribution's moments as follows:

1. For $t = 1$:

$$\begin{aligned}\Sigma_1 &= \left[\left\langle \frac{1}{\tau^2} \right\rangle I_d + \langle A_1^T C_1^{-1} A_1 \rangle \right]^{-1}, \\ \boldsymbol{\mu}_1 &= \Sigma_1 [\langle A_1^T C_1^{-1} \rangle \mathbf{y}_1 - \langle A_1^T C_1^{-1} \mathbf{b}_1 \rangle].\end{aligned}$$

2. For $t = 2, \dots, T$:

$$\begin{aligned}\Sigma_{t-1}^* &= \left[\left\langle \frac{1}{\sigma^2} \right\rangle I_d + \Sigma_{t-1}^{-1} \right]^{-1}, \\ \Sigma_t &= \left[\left\langle \frac{1}{\sigma^2} \right\rangle I_d + \langle A_t^T C_t^{-1} A_t \rangle - \left\langle \frac{1}{\sigma^2} \right\rangle^2 \Sigma_{t-1}^* \right]^{-1}, \\ \boldsymbol{\mu}_t &= \Sigma_t \left[\langle A_t^T C_t^{-1} \rangle \mathbf{y}_t - \langle A_t^T C_t^{-1} \mathbf{b}_t \rangle + \left\langle \frac{1}{\sigma^2} \right\rangle \Sigma_{t-1}^* \Sigma_{t-1}^{-1} \boldsymbol{\mu}_{t-1} \right].\end{aligned}$$

Output $\{\boldsymbol{\mu}_{1:T}, \Sigma_{1:T}, \Sigma_{1:(T-1)}^*\}$.

Algorithm 4.6: Variational Kalman filter.

$$\begin{aligned}& (\langle A_1^T C_1^{-1} \rangle \mathbf{y}_1 - \langle A_1^T C_1^{-1} \mathbf{b}_1 \rangle)^T \mathbf{x}_1), \\ & \propto N(\mathbf{x}_1 \mid \boldsymbol{\mu}_1, \Sigma_1),\end{aligned}$$

where

$$\begin{aligned}\Sigma_1 &= \left[\left\langle \frac{1}{\tau^2} \right\rangle I_d + \langle A_1^T C_1^{-1} A_1 \rangle \right]^{-1}, \\ \boldsymbol{\mu}_1 &= \Sigma_1 [\langle A_1^T C_1^{-1} \rangle \mathbf{y}_1 - \langle A_1^T C_1^{-1} \mathbf{b}_1 \rangle].\end{aligned}$$

The last line follows from matching the natural parameters of a Gaussian density. This shows that $\alpha_1(\mathbf{x}_1)$ is Gaussian; therefore, we can proceed by induction. For $t > 1$, we have

$$\begin{aligned}\alpha_t(\mathbf{x}_t) &\propto \int d\mathbf{x}_{t-1} h(\mathbf{x}_t, \mathbf{y}_{1:t}, \mathbf{x}_{t-1}), \\ &= \int d\mathbf{x}_{t-1} h(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1}) h(\mathbf{x}_t \mid \mathbf{x}_{t-1}) h(\mathbf{y}_t \mid \mathbf{x}_t), \\ &\propto \int d\mathbf{x}_{t-1} \alpha_{t-1}(\mathbf{x}_{t-1}) \times \\ &\quad \exp \left(-\frac{1}{2} \left[\mathbf{x}_{t-1}^T \left\langle \frac{1}{\sigma^2} \right\rangle I_d \mathbf{x}_{t-1} - 2 \mathbf{x}_{t-1}^T \left\langle \frac{1}{\sigma^2} \right\rangle I_d \mathbf{x}_t + \right. \right. \\ &\quad \left. \left. \mathbf{x}_t^T \left(\left\langle \frac{1}{\sigma^2} \right\rangle I_d + \langle A_t^T C_t^{-1} A_t \rangle \right) \mathbf{x}_t - 2 (\langle A_t^T C_t^{-1} \rangle \mathbf{y}_t - \langle A_t^T C_t^{-1} \mathbf{b}_t \rangle)^T \mathbf{x}_t \right] \right).\end{aligned}$$

Inside the integral is a $N(\mathbf{x}_{t-1} \mid \boldsymbol{\mu}_{t-1}^*, \Sigma_{t-1}^*)$ with

$$\begin{aligned}\Sigma_{t-1}^* &= \left[\left\langle \frac{1}{\sigma^2} \right\rangle I_d + \Sigma_{t-1}^{-1} \right]^{-1}, \\ \boldsymbol{\mu}_{t-1}^* &= \Sigma_{t-1}^* \left[\left\langle \frac{1}{\sigma^2} \right\rangle \mathbf{x}_t + \Sigma_{t-1}^{-1} \boldsymbol{\mu}_{t-1} \right].\end{aligned}$$

Marginalizing over this distribution leaves the following terms

$$\begin{aligned}\alpha_t(\mathbf{x}_t) &\propto \exp\left(-\frac{1}{2}\mathbf{x}_t^T \left(\left\langle \frac{1}{\sigma^2} \right\rangle I_d + \langle A_t^T C_t^{-1} A_t \rangle - \left\langle \frac{1}{\sigma^2} \right\rangle \Sigma_{t-1}^* \left\langle \frac{1}{\sigma^2} \right\rangle \right) \mathbf{x}_t + \right. \\ &\quad \left. (\langle A_t^T C_t^{-1} \rangle \mathbf{y}_t - \langle A_t^T C_t^{-1} \mathbf{b}_t \rangle)^T \mathbf{x}_t + \frac{1}{2} \boldsymbol{\mu}_{t-1}^{*T} \Sigma_{t-1}^{*-1} \boldsymbol{\mu}_{t-1}^* \right), \\ &\propto \exp\left(-\frac{1}{2}\mathbf{x}_t^T \left(\left\langle \frac{1}{\sigma^2} \right\rangle I_d + \langle A_t^T C_t^{-1} A_t \rangle - \left\langle \frac{1}{\sigma^2} \right\rangle \Sigma_{t-1}^* \left\langle \frac{1}{\sigma^2} \right\rangle \right) \mathbf{x}_t + \right. \\ &\quad \left. (\langle A_t^T C_t^{-1} \rangle \mathbf{y}_t - \langle A_t^T C_t^{-1} \mathbf{b}_t \rangle + \left\langle \frac{1}{\sigma^2} \right\rangle \Sigma_{t-1}^* \Sigma_{t-1}^{-1} \boldsymbol{\mu}_{t-1})^T \mathbf{x}_t \right), \\ &\propto N(\mathbf{x}_t \mid \boldsymbol{\mu}_t, \Sigma_t),\end{aligned}$$

where

$$\begin{aligned}\Sigma_t &= \left[\left\langle \frac{1}{\sigma^2} \right\rangle I_d + \langle A_t^T C_t^{-1} A_t \rangle - \left\langle \frac{1}{\sigma^2} \right\rangle^2 \Sigma_{t-1}^* \right]^{-1}, \\ \boldsymbol{\mu}_t &= \Sigma_t \left[\langle A_t^T C_t^{-1} \rangle \mathbf{y}_t - \langle A_t^T C_t^{-1} \mathbf{b}_t \rangle + \left\langle \frac{1}{\sigma^2} \right\rangle \Sigma_{t-1}^* \Sigma_{t-1}^{-1} \boldsymbol{\mu}_{t-1} \right].\end{aligned}$$

□

Variational Kalman Smoother

Property 4.2. *For the Gaussian state space model specified in Equations (4.15) – (4.17), the backwards message variables $h(\mathbf{y}_{(t+1):T} \mid \mathbf{x}_t) = N(\mathbf{x}_t \mid \boldsymbol{\eta}_t, \Psi_t)$ with parameters that can be calculated recursively via Algorithm 4.7.*

Proof. Define the backwards message variables $\beta_t(\mathbf{x}_t) = h(\mathbf{y}_{(t+1):T} \mid \mathbf{x}_t)$ and set $\beta_T(\mathbf{x}_T) = 1$. Note that

$$\begin{aligned}\beta_{t-1}(\mathbf{x}_{t-1}) &= \int d\mathbf{x}_t h(\mathbf{y}_{t:T}, \mathbf{x}_t \mid \mathbf{x}_{t-1}), \\ &= \int d\mathbf{x}_t h(\mathbf{y}_{(t+1):T} \mid \mathbf{x}_t) h(\mathbf{y}_t \mid \mathbf{x}_t) h(\mathbf{x}_t \mid \mathbf{x}_{t-1}), \\ &= \int d\mathbf{x}_t \beta_t(\mathbf{x}_t) h(\mathbf{y}_t \mid \mathbf{x}_t) h(\mathbf{x}_t \mid \mathbf{x}_{t-1}).\end{aligned}$$

The derivation proceeds sequentially backwards in time. For $t = T - 1$, we have

$$\begin{aligned}\beta_{T-1}(\mathbf{x}_{T-1}) &\propto \int d\mathbf{x}_T \times \\ &\quad \exp\left(-\frac{1}{2} \left\langle (\mathbf{y}_T - A_T \mathbf{x}_T - \mathbf{b}_T)^T C_T^{-1} (\mathbf{y}_T - A_T \mathbf{x}_T - \mathbf{b}_T) + \right.\right.\end{aligned}$$

Given the natural parameters $\langle \Gamma_{1:T}^1 \rangle = \langle A_{1:T}^T C_{1:T}^{-1} \rangle \mathbf{y}_{1:T} - \langle A_{1:T}^T C_{1:T}^{-1} \mathbf{b}_{1:T} \rangle$, $\langle \Gamma_{1:T}^2 \rangle = \langle A_{1:T}^T C_{1:T}^{-1} A_{1:T} \rangle$, $\langle 1/\tau^2 \rangle$, and $\langle 1/\sigma^2 \rangle$, calculate the smoothing distribution's moments and cross-covariances as follows:

1. Calculate $\{\boldsymbol{\mu}_{1:T}, \Sigma_{1:T}, \Sigma_{1:(T-1)}^*\}$ as in Algorithm 4.6.

2. Set $\boldsymbol{\nu}_T = \boldsymbol{\mu}_T$ and $\Upsilon_T = \Sigma_T$.

3. Initialize $\Psi_T^{-1} = 0$ to satisfy $\beta_T(\mathbf{x}_T) = 1$.

4. For $t = T, \dots, 2$.

▷ Calculate backwards message variables

$$\begin{aligned}\Psi_t^* &= \left[\left\langle \frac{1}{\sigma^2} \right\rangle I_d + \langle A_t^T C_t^{-1} A_t \rangle + \Psi_t^{-1} \right]^{-1}, \\ \Psi_{t-1} &= \left[\left\langle \frac{1}{\sigma^2} \right\rangle I_d - \left\langle \frac{1}{\sigma^2} \right\rangle^2 \Psi_t^* \right]^{-1}, \\ \boldsymbol{\eta}_{t-1} &= \Psi_{t-1} \left[\left\langle \frac{1}{\sigma^2} \right\rangle \Psi_t^* (\langle A_t^T C_t^{-1} \rangle \mathbf{y}_t - \langle A_t^T C_t^{-1} \mathbf{b}_t \rangle) + \Psi_t^{-1} \boldsymbol{\eta}_t \right].\end{aligned}$$

▷ Calculate smoothing distributions and cross-covariances

$$\begin{aligned}\Upsilon_{t-1} &= [\Sigma_{t-1}^{-1} + \Psi_{t-1}^{-1}]^{-1}, \\ \Upsilon_{t-1,t} &= \left\langle \frac{1}{\sigma^2} \right\rangle \Sigma_{t-1}^* \left[\Psi_t^{*-1} - \left\langle \frac{1}{\sigma^2} \right\rangle^2 \Sigma_{t-1}^* \right]^{-1}, \\ \boldsymbol{\nu}_{t-1} &= \Upsilon_{t-1} [\Sigma_{t-1}^{-1} \boldsymbol{\mu}_{t-1} + \Psi_{t-1}^{-1} \boldsymbol{\eta}_{t-1}].\end{aligned}$$

Output smoothing distributions $\{\boldsymbol{\nu}_{1:T}, \Upsilon_{1:T}\}$ and cross-covariances $\{\Upsilon_{t,t+1}\}_{t=1}^{T-1}$.

Algorithm 4.7: Variational Kalman smoother. Throughout the text, we refer to this algorithm as `KalmanSmoother`($\langle \Gamma_{1:T}^1 \rangle, \langle \Gamma_{1:T}^2 \rangle, \langle 1/\tau^2 \rangle, \langle 1/\sigma^2 \rangle$).

$$\begin{aligned}
& (\mathbf{x}_T - \mathbf{x}_{T-1})^T \frac{1}{\sigma^2} I_d (\mathbf{x}_T - \mathbf{x}_{T-1}) \Big\rangle \Bigg), \\
& \propto \int d\mathbf{x}_T \exp \left(-\frac{1}{2} \left[\mathbf{x}_{T-1} \left\langle \frac{1}{\sigma^2} \right\rangle I_d \mathbf{x}_{T-1} + \mathbf{x}_T^T \left(\left\langle \frac{1}{\sigma^2} \right\rangle I_d + \langle A_T^T C_T^{-1} A_T \rangle \right) \mathbf{x}_T - \right. \right. \\
& \quad \left. \left. 2 \langle \langle A_T^T C_T^{-1} \rangle \mathbf{y}_T - \langle A_T^T C_T^{-1} \mathbf{b}_T \rangle + \left\langle \frac{1}{\sigma^2} \right\rangle \mathbf{x}_{T-1} \right)^T \mathbf{x}_T \right] \right).
\end{aligned}$$

Inside the integral is a $N(\mathbf{x}_T \mid \boldsymbol{\eta}_T^*, \Psi_T^*)$ with

$$\begin{aligned}
\Psi_T^* &= \left[\left\langle \frac{1}{\sigma^2} \right\rangle I_d + \langle A_T^T C_T^{-1} A_T \rangle \right]^{-1}, \\
\boldsymbol{\eta}_T^* &= \Psi_T^* \left[\langle A_T^T C_T^{-1} \rangle \mathbf{y}_T - \langle A_T^T C_T^{-1} \mathbf{b}_T \rangle + \left\langle \frac{1}{\sigma^2} \right\rangle \mathbf{x}_{T-1} \right].
\end{aligned}$$

Marginalizing over this density, we are left with

$$\begin{aligned}
\beta_{T-1}(\mathbf{x}_{T-1}) &\propto \exp \left(-\frac{1}{2} \mathbf{x}_{T-1}^T \left\langle \frac{1}{\sigma^2} \right\rangle \mathbf{x}_{T-1} + \frac{1}{2} \boldsymbol{\eta}_T^{*T} \Psi_T^{*-1} \boldsymbol{\eta}_T^* \right), \\
&\propto \exp \left(-\frac{1}{2} \mathbf{x}_{T-1}^T \left(\left\langle \frac{1}{\sigma^2} \right\rangle I_d - \left\langle \frac{1}{\sigma^2} \right\rangle^2 \Psi_T^* \right) \mathbf{x}_{T-1} + \right. \\
&\quad \left. \left(\langle A_T^T C_T^{-1} \rangle \mathbf{y}_T - \langle A_T^T C_T^{-1} \mathbf{b}_T \rangle \right)^T \left\langle \frac{1}{\sigma^2} \right\rangle \Psi_T^* \mathbf{x}_{T-1} - 1 \right), \\
&\propto N(\mathbf{x}_{T-1} \mid \boldsymbol{\eta}_{T-1}, \Psi_{T-1}),
\end{aligned}$$

where

$$\begin{aligned}
\Psi_{T-1} &= \left[\left\langle \frac{1}{\sigma^2} \right\rangle I_d - \left\langle \frac{1}{\sigma^2} \right\rangle^2 \Psi_T^* \right]^{-1}, \\
\boldsymbol{\eta}_{T-1} &= \Psi_{T-1} \left[\left\langle \frac{1}{\sigma^2} \right\rangle \Psi_T^* \left(\langle A_T^T C_T^{-1} \rangle \mathbf{y}_T - \langle A_T^T C_T^{-1} \mathbf{b}_T \rangle \right) \right].
\end{aligned}$$

Now we proceed inductively. For $2 < t < T-1$, we have

$$\begin{aligned}
\beta_{t-1}(\mathbf{x}_{t-1}) &\propto \int d\mathbf{x}_t N(\mathbf{x}_t \mid \boldsymbol{\eta}_t, \Psi_t) \times \\
&\quad \exp \left(-\frac{1}{2} \left(\mathbf{y}_t - A_t \mathbf{x}_t - \mathbf{b}_t \right)^T C_t^{-1} (\mathbf{y}_t - A_t \mathbf{x}_t - \mathbf{b}_t) + \right. \\
&\quad \left. (\mathbf{x}_t - \mathbf{x}_{t-1})^T \frac{1}{\sigma^2} I_d (\mathbf{x}_t - \mathbf{x}_{t-1}) \right).
\end{aligned}$$

In fact, the remaining derivation is exactly the same as before, except

$$\begin{aligned}
\Psi_t^* &= \left[\left\langle \frac{1}{\sigma^2} \right\rangle I_d + \langle A_t^T C_t^{-1} A_t \rangle + \Psi_t^{-1} \right]^{-1}, \\
\boldsymbol{\eta}_t^* &= \Psi_t^* \left[\langle A_t^T C_t^{-1} \rangle \mathbf{y}_t - \langle A_t^T C_t^{-1} \mathbf{b}_t \rangle + \Psi_t^{-1} \boldsymbol{\eta}_t + \left\langle \frac{1}{\sigma^2} \right\rangle \mathbf{x}_{t-1} \right],
\end{aligned}$$

so that $\beta_{t-1}(\mathbf{x}_{t-1}) = N(\mathbf{x}_{t-1} \mid \boldsymbol{\eta}_{t-1}, \Psi_{t-1})$ with

$$\begin{aligned}\Psi_{t-1} &= \left[\left\langle \frac{1}{\sigma^2} \right\rangle I_d - \left\langle \frac{1}{\sigma^2} \right\rangle^2 \Psi_T^* \right]^{-1}, \\ \boldsymbol{\eta}_{t-1} &= \Psi_{t-1} \left[\left\langle \frac{1}{\sigma^2} \right\rangle \Psi_T^* (\langle A_t^T C_t^{-1} \rangle \mathbf{y}_t - \langle A_t^T C_t^{-1} \mathbf{b}_t \rangle) + \Psi_t^{-1} \boldsymbol{\eta}_t \right].\end{aligned}$$

□

Property 4.3. *For the Gaussian state space model specified in Equations (4.15) – (4.17), the variational smoothing distributions $h(\mathbf{x}_t \mid \mathbf{y}_{1:T}) = N(\mathbf{x}_t \mid \boldsymbol{\nu}_t, \Upsilon_t)$ with parameters that can be calculated recursively via Algorithm 4.7.*

Proof. We define forwards and backwards message variables, $\alpha_t(\mathbf{x}_t)$ and $\beta_t(\mathbf{x}_t)$, as in the previous proofs. For $t = T$, $h(\mathbf{x}_T \mid \mathbf{y}_{1:T}) = \alpha_T(\mathbf{x}_T) = N(\mathbf{x}_T \mid \boldsymbol{\mu}_T, \Sigma_T)$. For $t < T - 1$, we have

$$\begin{aligned}h(\mathbf{x}_t \mid \mathbf{y}_{1:T}) &\propto \alpha_t(\mathbf{x}_t) \beta_t(\mathbf{x}_t), \\ &\propto N(\mathbf{x}_t \mid \boldsymbol{\mu}_t, \Sigma_t) N(\mathbf{x}_t \mid \boldsymbol{\eta}_t, \Psi_t), \\ &\propto N(\mathbf{x}_t \mid \boldsymbol{\nu}_t, \Upsilon_t),\end{aligned}$$

where

$$\begin{aligned}\Upsilon_t &= [\Sigma_t^{-1} + \Psi_t^{-1}]^{-1}, \\ \boldsymbol{\nu}_t &= \Upsilon_t [\Sigma_t^{-1} \boldsymbol{\mu}_t + \Psi_t^{-1} \boldsymbol{\eta}_t].\end{aligned}$$

□

Cross-Covariance Matrices

Property 4.4. *For the Gaussian state space model specified in Equations (4.15) – (4.17), the variational joint distributions $h(\mathbf{x}_t, \mathbf{x}_{t+1} \mid \mathbf{y}_{1:T})$ are Gaussian with cross-covariance matrices $\Upsilon_{t,t+1}$ that can be calculated recursively via Algorithm 4.7.*

Proof. We have that

$$\begin{aligned}h(\mathbf{x}_t, \mathbf{x}_{t+1} \mid \mathbf{y}_{1:T}) &\propto h(\mathbf{x}_t \mid \mathbf{x}_{1:t}) h(\mathbf{x}_{t+1} \mid \mathbf{x}_t) h(\mathbf{y}_{t+1} \mid \mathbf{x}_{t+1}) h(\mathbf{y}_{(t+2):T} \mid \mathbf{x}_{t+1}), \\ &\propto \alpha_t(\mathbf{x}_t) h(\mathbf{x}_{t+1} \mid \mathbf{x}_t) h(\mathbf{y}_{t+1} \mid \mathbf{x}_{t+1}) \beta_{t+1}(\mathbf{x}_{t+1}).\end{aligned}$$

To determine $\Upsilon_{t,t+1}$, we identify the cross-terms in the above product. This product is proportional to

$$\begin{aligned}&\alpha_t(\mathbf{x}_t) \times \\ &\exp \left(-\frac{1}{2} (-\mathbf{x}_t^T \left\langle \frac{1}{\sigma^2} \right\rangle I_d \mathbf{x}_{t+1} + \mathbf{x}_t^T \left\langle \frac{1}{\sigma^2} \right\rangle I_d \mathbf{x}_t + \mathbf{x}_{t+1}^T \left\langle \frac{1}{\sigma^2} \right\rangle I_d \mathbf{x}_{t+1} \right. \\ &\quad \left. \mathbf{x}_{t+1}^T \langle A_{t+1}^T C_{t+1}^{-1} A_{t+1} \rangle \mathbf{x}_{t+1} - \right. \\ &\quad \left. 2 \langle \langle A_{t+1}^T C_{t+1}^{-1} \rangle \mathbf{y}_{t+1} + \langle A_{t+1}^T C_{t+1}^{-1} \mathbf{b}_{t+1} \rangle \rangle^T \mathbf{x}_{t+1} \right) \times\end{aligned}$$

$$\beta_{t+1}(\mathbf{x}_{t+1}),$$

$$\propto \exp \left\{ -\frac{1}{2} \begin{pmatrix} \mathbf{x}_t \\ \mathbf{x}_{t+1} \end{pmatrix}^T \begin{pmatrix} \Gamma_t & \Gamma_{t,t+1} \\ \Gamma_{t,t+1}^T & \Gamma_{t+1} \end{pmatrix} \begin{pmatrix} \mathbf{x}_t \\ \mathbf{x}_{t+1} \end{pmatrix} \right\},$$

with

$$\begin{aligned} \Gamma_t &= \left\langle \frac{1}{\sigma^2} \right\rangle I_d + \Sigma_t^{-1} = \Sigma_t^{*-1}, \\ \Gamma_{t+1} &= \left\langle \frac{1}{\sigma^2} \right\rangle I_d + \langle A_{t+1}^T C_{t+1}^{-1} A_{t+1} \rangle + \Psi_{t+1}^{-1} = \Psi_{t+1}^{*-1}, \\ \Gamma_{t,t+1} &= -\left\langle \frac{1}{\sigma^2} \right\rangle I_d, \end{aligned}$$

where in the last line we only kept terms quadratic in $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$. Applying Schur's compliment, we obtain the cross-covariance matrix

$$\Upsilon_{t,t+1} = \Sigma_t^* \left\langle \frac{1}{\sigma^2} \right\rangle \left[\Psi_{t+1}^{*-1} - \left\langle \frac{1}{\sigma^2} \right\rangle^2 \Sigma_t^* \right]^{-1}.$$

□

4.7.4 Parameter Initialization and Prior Settings

Due to variational inference algorithm's non-convex objective, appropriate initial values of the approximate posterior parameters can greatly improve convergence. We used the same initialization scheme and prior settings for all experiments and real-data applications.

We initialized the social trajectories $\delta_{1:K,1:T}$ by estimating independent two-way logistic regression models

$$Y_{ijt}^k \stackrel{\text{iid}}{\sim} \text{Bernoulli} \left(\text{logit}^{-1} \left[\delta_{k,t}^i + \delta_{k,t}^j \right] \right),$$

for each $k = 1, \dots, K$ and $t = 1, \dots, T$. Furthermore, we set $\sigma_{\delta_{k,t}^i}^2 = 1$ and $\sigma_{\delta_{k,t,t+1}^i}^2 = 1$. We placed broad priors on the state space parameters. To make the prior on τ_δ^2 flat, one can set the shape and scale parameters of the inverse gamma priors to $a_{\tau_\delta^2} = 2(2 + \epsilon)$ and $b_{\tau_\delta^2} = 2(1 + \epsilon)\mathbb{E}[\tau_\delta^2]$ for some small $\epsilon > 0$, respectively. We set $\epsilon = 0.05$ and $\mathbb{E}[\tau_\delta^2] = 10$. For σ_δ^2 , we set $c_{\sigma_\delta^2} = 2$ and $d_{\sigma_\delta^2} = 2$.

Next, we initialized the latent trajectories $\mathcal{X}_{1:T}$ by sampling the entries independently from a $N(0, 1)$. We set the variances and cross-covariances to the identity matrix I_d . As for the social trajectories, we can make the prior on τ^2 flat by setting $a_{\tau^2} = 2(2 + \epsilon)$ and $b_{\tau^2} = 2(1 + \epsilon)\mathbb{E}[\tau^2]$. We set $\epsilon = 0.05$ and $\mathbb{E}[\tau^2] = 10$. For σ^2 , we set $c_{\sigma^2} = 2$ and $d_{\sigma^2} = 2$.

For the homophily matrix, we initialized the reference layer differently from the rest. For the reference layer, we set $\mu_{\lambda_{1h}} = 1$ for $h = 1, \dots, d$ and $\rho = 1/2$. We sampled $\boldsymbol{\mu}_{\lambda_k} \sim N(0, 4I_d)$ and set $\Sigma_{\lambda_k} = 10I_d$. In addition, we set the prior variance to $\sigma_\lambda^2 = 10$.

Lastly, we initialized the mean of the Pólya-gamma random variables, $\mu_{\omega_{ijt}^k}$, to zero.

4.7.5 Additional Figures

In this section, we include the remaining figures for the data analyzed in the main text's real data applications (Section 4.5). These figures include the remaining social trajectories for the international relations and

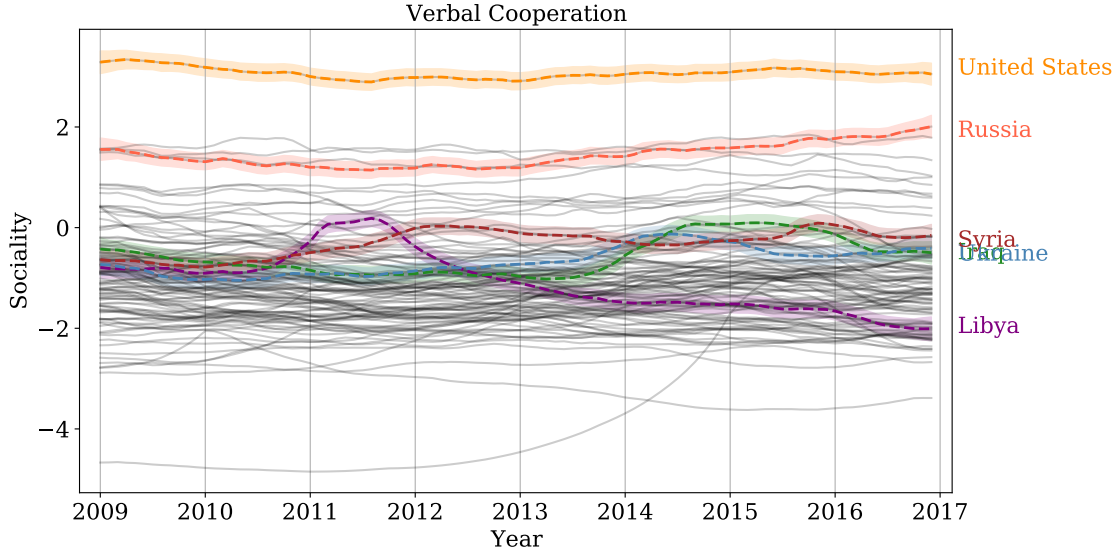


Figure 4.13: Posterior means of the verbal cooperation social trajectories for the ICEWS network. Select countries are highlighted in color with bands that represent 95% credible intervals. The remaining countries' social trajectories are displayed with gray curves.

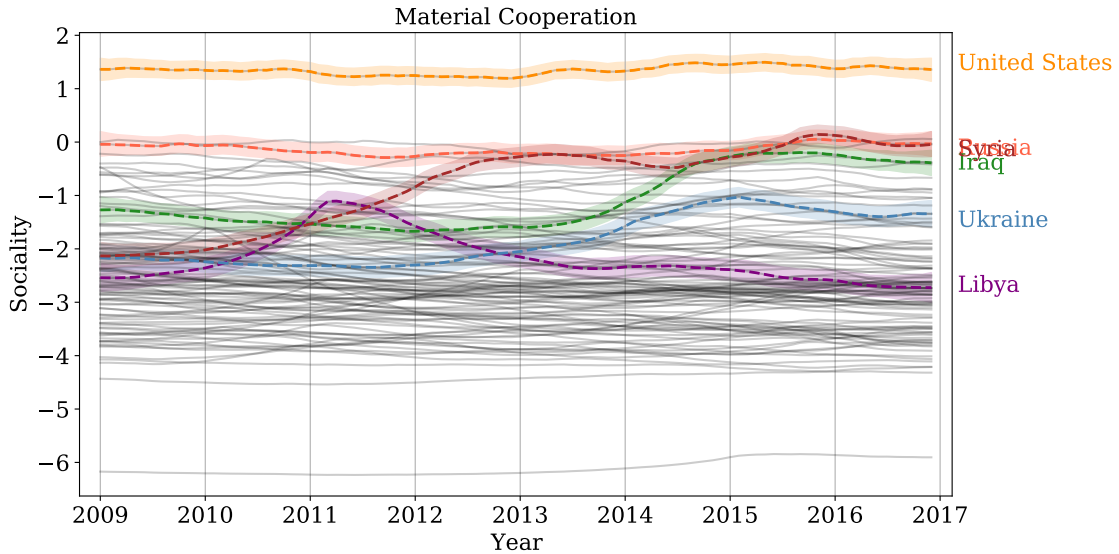


Figure 4.14: Posterior means of the material cooperation social trajectories for the ICEWS network. Select countries are highlighted in color with bands that represent 95% credible intervals. The remaining countries' social trajectories are displayed with gray curves.

primary school networks and the homophily coefficients for the primary school networks.

We start with the remaining figures for the ICEWS data set. Figure 4.13 contains the social trajectories for the verbal cooperation networks, Figure 4.14 contains the social trajectories for the material cooperation networks, and Figure 4.15 contains the social trajectories for the verbal conflict networks. The shapes of the social trajectories are similar to the material conflict relation. A notable difference is that Ukraine increases its material cooperation and verbal conflict sociality dramatically leading up to the Crimea Crisis.

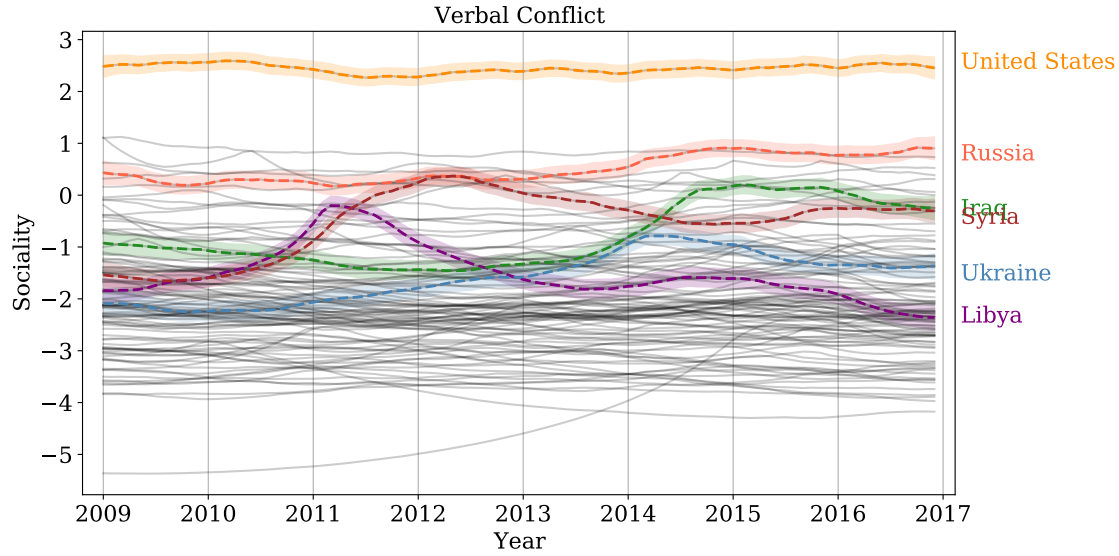


Figure 4.15: Posterior means of the verbal conflict social trajectories for the ICEWS network. Select countries are highlighted in color with bands that represent 95% credible intervals. The remaining countries' social trajectories are displayed with gray curves.

Next, we present the social trajectories and homophily coefficients for the primary school network. Figure 4.16 and Figure 4.17 contain the actors' social trajectories on Thursday and Friday, respectively. We highlighted the trajectories of three actors. Actor 148 is a teacher, actor 195 is a student in class 3A, and actor 5 is a student in class 5B. Their social trajectories demonstrate three interesting longitudinal patterns. Actor 148, the teacher, is most socially active during class and least active during lunch. Conversely, actor 195 is most sociable during lunch and less sociable during class. Lastly, actor 5's social trajectory differs between the two days because he/she is absent on Friday. Next, Figure 4.18 contains the primary school network's homophily coefficients. All homophily coefficients are positive. Also, the magnitude of the homophily coefficients is larger on Friday than on Thursday.

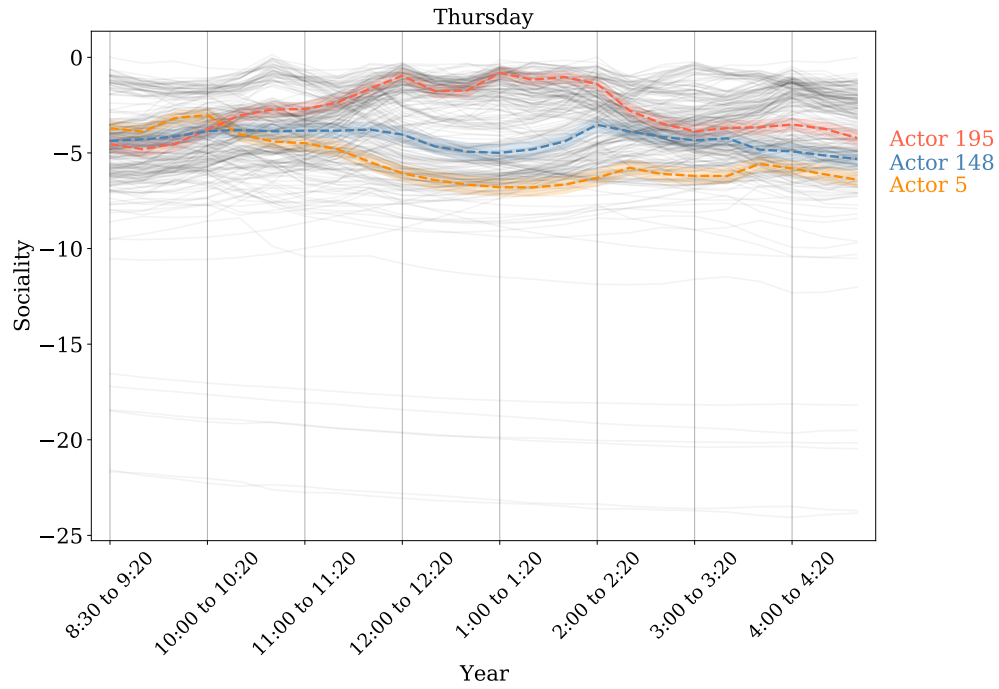


Figure 4.16: Posterior means of the social trajectories on Thursday for the primary school network. Select actors are highlighted in color with bands that represent 95% credible intervals. The remaining actors' social trajectories are displayed with gray curves.

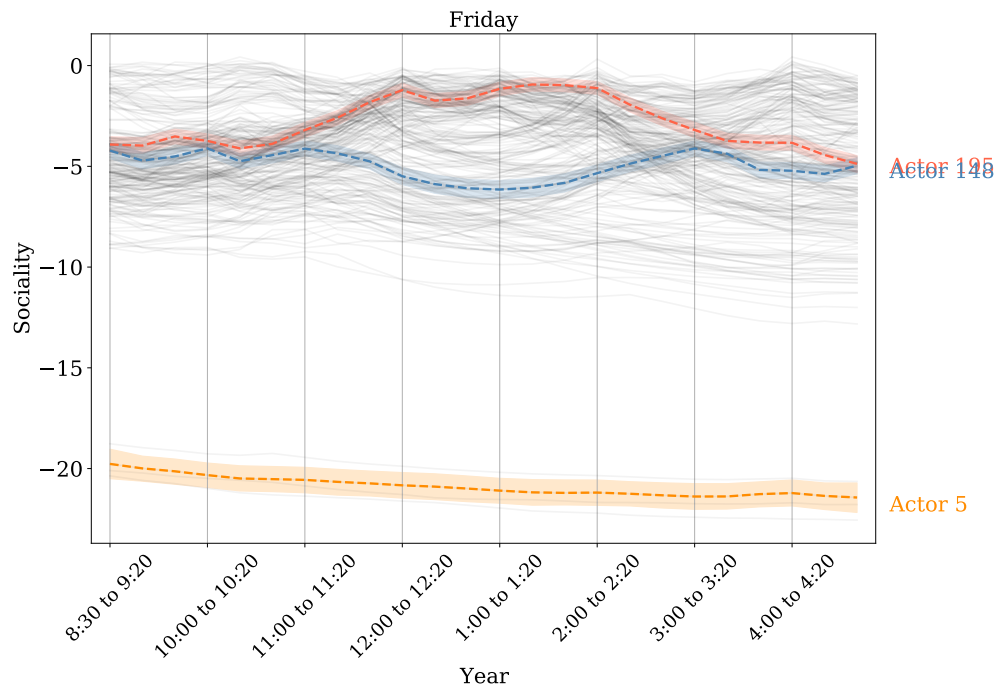


Figure 4.17: Posterior means of the social trajectories on Friday for the primary school network. Select actors are highlighted in color with bands that represent 95% credible intervals. The remaining actors' social trajectories are displayed with gray curves.

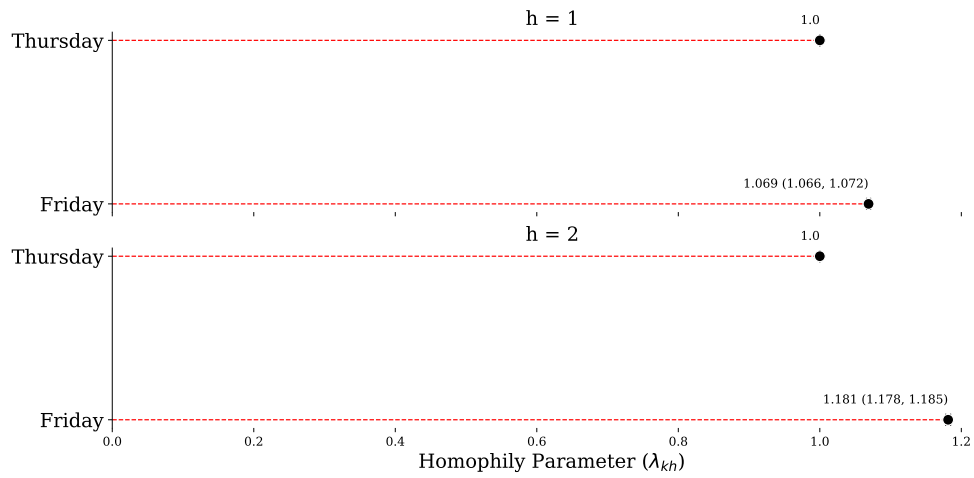


Figure 4.18: The homophily coefficients' posterior means and 95% credible intervals for the primary school face-to-face contact networks. The top and bottom plots give estimates for the degree of homophily along the first and second latent dimensions, respectively.

Chapter 5

Spike-and-Slab Priors for Dimension Selection in Network Eigenmodels

5.1 Introduction

Latent space models (LSMs) (P. D. Hoff et al., 2002) aim to infer low-dimensional structure in network data and have seen many applications in fields such as sociology, biology, political science, and economics. The popularity of latent space models stems from their ability to capture various network structures while remaining amiable to interpretation and visualization. LSMs assume that each node i in a network with n nodes can be represented by a vector $\mathbf{u}_i \in \mathbb{R}^d$ in a low-dimensional ($d \ll n$) latent space. This space is equipped with a scalar-valued similarity function, $g(\cdot, \cdot)$, such that nodes deemed “similar” by $g(\mathbf{u}_i, \mathbf{u}_j)$ are more likely to be connected in the network. LSMs have been used to model static networks (P. D. Hoff, 2005; Krivitsky et al., 2009; Z. Ma et al., 2020), dynamic or time-varying networks (Durante & Dunson, 2014; Sarkar & Moore, 2005; Sewell & Chen, 2015b), and multilayer networks (Gollini & Murphy, 2016; MacDonald et al., in press; Zhang et al., 2020). Furthermore, LSMs are flexible enough to describe networks with different edge types (P. Hoff, 2021; Sewell & Chen, 2015a, 2016) and to incorporating dyad-wise covariates (P. D. Hoff et al., 2002). For a comprehensive overview of recent developments in LSMs, see B. Kim et al. (2018) and Sosa and Buitrago (2021).

A persistent difficulty when using latent space models for statistical network analysis is selecting the latent space’s dimension d . An appropriate choice of d is crucial to ensuring that the model captures the essential properties of the network while balancing the bias-variance trade-off. Also, the dimension d is of empirical interest since the latent space could have a real-world interpretation. The traditional approach to dimension selection involves estimating LSMs for different values of d and selecting the model that minimizes some model selection criterion. These criteria include information criteria such as the deviance information criterion (DIC) (Spiegelhalter et al., 2002) and the Watanabe-Akaike information criterion (WAIC) (Watanabe, 2010), or metrics calculated on data held out by a data splitting strategy such as cross-validation (K. Chen & Lei, 2018; P. D. Hoff, 2008; L. Li et al., 2020). For Bayesian LSMs, significant issues with these approaches are that they either are too computationally intensive, lack theoretical support, or only apply to static networks without covariates. As such, many Bayesian analyses forgo dimension selection altogether.

The main thrust of this chapter is the development of a computationally efficient and theoretically grounded Bayesian approach that can automatically identify the dimension of the latent space for both

static and dynamic networks with undirected edges. To do so, we introduce a new model for dynamic networks that reduces to a model similar to the static network eigenmodel (P. D. Hoff, 2008, 2009) when we observe a single static network. In the case of a dynamic network observed over $t = 1, \dots, T$ time points, the network eigenmodel uses the similarity function $g(\mathbf{u}_{it}, \mathbf{u}_{jt}) = \mathbf{u}_{it}^T \mathbf{\Lambda}_t \mathbf{u}_{jt}$, where $\mathbf{\Lambda}_t = \text{diag}(\lambda_{1t}, \dots, \lambda_{dt})$ is a time-varying diagonal matrix and $\mathbf{u}_{it} \in \mathbb{R}^d$ is the latent position of node i at time t . We focus on the network eigenmodel because it includes popular LSMs as special cases. Specifically, P. D. Hoff (2009) showed that the static network eigenmodel generalizes the stochastic block model (Holland et al., 1983) and the latent distance model ($g(\mathbf{u}_i, \mathbf{u}_j) = -\|\mathbf{u}_i - \mathbf{u}_j\|_2$) in the sense that the eigenmodel can represent their network structure but not vice versa. In particular, our proposed dynamic network eigenmodel captures both time-varying positive and negative homophily patterns, while previous dynamic LSMs can only model positive homophily patterns.

Our approach to dimension selection begins with a prior on $\{\lambda_{ht}\}_{h=1}^d$ for $t = 1, \dots, T$ that induces posterior zeros with high probability. This entails some variant of a spike-and-slab prior commonly used in the regression setting to separate important and negligible coefficients (Ishwaran & Rao, 2005). However, unlike the typical regression setting, we expect additional dimensions to play a progressively less important role in describing the network structure. As such, we introduce a novel spike-and-slab prior coupled with a non-homogeneous Indian buffet process prior (Griffiths & Ghahramani, 2011) that increasingly pulls coefficients to the spike at zero. The advantages of this prior are two-fold: (1) we demonstrate that the prior induces a stochastic ordering on $\{\lambda_{ht}\}_{h=1}^d$ for $t = 1, \dots, T$ that alleviates issues arising from the likelihood’s invariance to the re-ordering of the latent features, and (2) in the static case, we provide a tail bound on the expected posterior latent space dimension that demonstrates concentration near the true number of dimensions. To the best of our knowledge, this is the first such theoretical guarantee for Bayesian LSMs. Recent applications of the static network eigenmodel (Guha & Rodriguez, 2021; Guhaniyogi & Rodriguez, 2020) have employed spike-and-slab priors for dimension selection that penalize increasing model complexity; however, these priors’ increasing penalization only holds in expectation and lack theoretical guarantees. In addition, Durante and Dunson (2014) used an ordered shrinkage prior (Battacharya & Dunson, 2011) to reduce sensitivity to the choice of d in a dynamic LSM; however, this prior often induces undesirable overshrinkage and has other practical disadvantages (Durante, 2017).

Our final contribution addresses the challenge of efficient Markov chain Monte Carlo (MCMC) sampling when the model parameters lie on the Stiefel manifold. We address this challenge by developing a sampler based on a strategy proposed by Jauch et al. (2021) that targets a posterior distribution on a parameter expanded space with marginal posterior densities for the model parameters of interest. This approach allows us to employ a simple Metropolis-within-Gibbs sampler with efficient Hamiltonian Monte Carlo proposals (Hoffman & Gelman, 2014; Neal, 2011).

The chapter is structured as follows. Section 5.2 introduces the proposed dynamic network eigenmodel. Section 5.3 develops our Bayesian approach to dimension selection and demonstrates some properties of the prior and posterior. Section 5.4 describes our parameter expansion strategy and MCMC sampler. Section 5.5 contains a simulation study that compares our approach to existing methods. In Section 5.6, we apply our methodology to model selection problems involving real static and dynamic networks. Section 5.7 contains a discussion. Finally, the supplementary material in Section 5.8 contains proofs, implementation details, and additional figures.

5.2 The Models

In this section, we describe a Bayesian modeling framework for static and dynamic networks that allows for dimension selection through a novel spike-and-slab prior. We start by fixing notation. Suppose we observe an undirected dynamic network on n nodes over $t = 1, \dots, T$ time points. We represent this dynamic network by a sequence of $n \times n$ symmetric adjacency matrices $\{\mathbf{Y}_t\}_{t=1}^T$ with binary entries $Y_{ijt} \in \{0, 1\}$. An entry indicates the presence ($Y_{ijt} = Y_{jit} = 1$) or absence ($Y_{ijt} = Y_{jit} = 0$) of an edge between nodes i and j at time t . Also, we may observe a set of dyad-wise covariates $\{\mathbf{X}_{ijt}\}_{i < j}$ with $\mathbf{X}_{ijt} = \mathbf{X}_{jit} \in \mathbb{R}^p$ for $t = 1, \dots, T$. These covariates measure time-varying characteristics of node pairs. We use the notation $A_{1:L}$ to refer to the set $\{A_1, \dots, A_L\}$ where A_ℓ is any indexed object. We denote the n -dimensional vector of ones and zeros by $\mathbf{1}_n$ and $\mathbf{0}_n$, respectively. We let I_d represent the $d \times d$ identity matrix. Given any vector $\mathbf{v} \in \mathbb{R}^d$, we use $\text{diag}(\mathbf{v})$ to denote the $d \times d$ diagonal matrix with the elements of $\mathbf{v}$ on its diagonal. Lastly, for a matrix $C \in \mathbb{R}^{n \times m}$, we denote the Frobenius norm as $\|C\|_F = (\sum_{ij} C_{ij}^2)^{1/2}$.

5.2.1 The Dynamic Network Eigenmodel

Conditioned on the model parameters and dyad-wise covariates, we assume that the adjacency matrices' temporal dependence has a Markov structure, i.e., $\mathbf{Y}_t \mid \mathbf{Y}_{t-1}, \dots, \mathbf{Y}_1 \stackrel{d}{=} \mathbf{Y}_t \mid \mathbf{Y}_{t-1}$ for $t = 2, \dots, T$. Under this dependency structure, we introduce the following model for any $i < j$ and $t = 1, \dots, T$:

$$Y_{ijt} \mid Y_{ij(t-1)} = Y_{jit} \mid Y_{ji(t-1)} \stackrel{\text{ind.}}{\sim} \text{Bernoulli} \left[\mathbb{P}(Y_{ijt} = 1 \mid Y_{ij(t-1)} = y_{ij(t-1)}) \right], \quad \text{with} \quad (5.1)$$

$$\text{logit} \left[\mathbb{P}(Y_{ijt} = 1 \mid Y_{ij(t-1)} = y_{ij(t-1)}) \right] = \beta^T \mathbf{X}_{ijt} + \gamma_t y_{ij(t-1)} + \alpha_t (1 - y_{ij(t-1)}) + [\mathbf{U}_t \boldsymbol{\Lambda}_t \mathbf{U}_t^T]_{ij},$$

where $\beta \in \mathbb{R}^p$, $\gamma_t \in \mathbb{R}$, $\alpha_t \in \mathbb{R}$, and $\boldsymbol{\Lambda}_t = \text{diag}(\lambda_{1t}, \dots, \lambda_{dt}) = \text{diag}(\boldsymbol{\lambda}_t) \in \mathbb{R}^{d \times d}$ is a real-valued diagonal matrix. Note that we set $\gamma_1 = \alpha_1 = 0$ for the initial observation. For the identifiability of $\boldsymbol{\Lambda}_{1:T}$, we require $\mathbf{U}_t \in \mathcal{V}_{d,n}$, where $\mathcal{V}_{d,n} = \{\mathbf{U} \in \mathbb{R}^{n \times d} \mid \mathbf{U}^T \mathbf{U} = I_d\}$ is the set of semi-orthogonal matrices with $d \leq n$ known as the Stiefel manifold. Furthermore, we require the columns of $\mathbf{U}_1$ to sum to zero, i.e., $\mathbf{U}_1^T \mathbf{1}_n = \mathbf{0}_d$, so that $\mathbf{U}_1$ is a centered semi-orthogonal matrix that lies in a submanifold of the Stiefel manifold. We denote the set of centered semi-orthogonal matrices by $\tilde{\mathcal{V}}_{d,n} = \{\mathbf{U} \in \mathbb{R}^{n \times d} \mid \mathbf{U}^T \mathbf{U} = I_d \text{ and } \mathbf{U}^T \mathbf{1}_n = \mathbf{0}_d\}$. As we elaborate at the end of this section, the sum-to-zero condition along with a full-rank condition on the covariates renders the covariate effects β identifiable. We place this model within the latent space framework presented earlier by re-expressing $[\mathbf{U}_t \boldsymbol{\Lambda}_t \mathbf{U}_t^T]_{ij}$ as a bilinear product $\mathbf{u}_{it}^T \boldsymbol{\Lambda}_t \mathbf{u}_{jt}$, where $\mathbf{u}_{it}$ is the i th row of $\mathbf{U}_t$, i.e., $\mathbf{U}_t = (\mathbf{u}_{1t}, \dots, \mathbf{u}_{nt})^T$. In this way, we interpret the vector $\mathbf{u}_{it} = (u_{it,1}, \dots, u_{it,d})^T$ as the latent position of node i at time t .

Our model includes both edge persistent parameters $\{\gamma_t\}_{t=2}^T$ and non-edge persistent parameters $\{\alpha_t\}_{t=2}^T$ introduced by Friel et al. (2016). The edge (non-edge) persistent parameters capture the phenomenon that once an edge (non-edge) forms at time t , it tends to persist. These parameters allow the model to capture additional temporal dyadic dependencies beyond those included in the latent space and covariates. Formally, γ_t (α_t) is the log-odds ratio of an edge persisting (forming) at time t , holding all other parameters fixed.

To complete the model, we place the following prior structure on the model parameters:

$$(\mathbf{U}_1, \dots, \mathbf{U}_T) \sim p(\mathbf{U}_1) \prod_{t=2}^T p(\mathbf{U}_t \mid \mathbf{U}_{t-1}) \quad (5.2)$$

$$(\mathbf{\Lambda}_1, \dots, \mathbf{\Lambda}_T) \sim p(\mathbf{\Lambda}_1) \prod_{t=2}^T p(\mathbf{\Lambda}_t \mid \mathbf{\Lambda}_{t-1}) \quad (5.3)$$

$$\boldsymbol{\beta} \sim N(\mathbf{0}_p, \sigma_\beta^2 I_p) \quad (5.4)$$

$$(\gamma_2, \dots, \gamma_T) \mid \sigma_\gamma \sim N(\gamma_2 \mid 0, \sigma_{0,\gamma}^2) \prod_{t=3}^T N(\gamma_t \mid \gamma_{t-1}, \sigma_\gamma^2) \quad (5.5)$$

$$(\alpha_2, \dots, \alpha_T) \mid \sigma_\alpha \sim N(\alpha_2 \mid 0, \sigma_{0,\alpha}^2) \prod_{t=3}^T N(\alpha_t \mid \alpha_{t-1}, \sigma_\alpha^2) \quad (5.6)$$

$$\sigma_\gamma \sim C^+(0, 1), \quad \sigma_\alpha \sim C^+(0, 1), \quad (5.7)$$

where $C^+(0, 1)$ denotes a half-Cauchy distribution with mean zero and scale one, and $N(x \mid \mu, \sigma^2)$ denotes a normal density with mean and variance μ and σ^2 , respectively. Note that the initial variances $\sigma_{0,\gamma}^2$ and $\sigma_{0,\alpha}^2$ are hyperparameters, which should be set to induce a broad prior. We specify the distributions for $\mathbf{\Lambda}_{1:T}$ and $\mathbf{U}_{1:T}$ later.

As previously mentioned, the sum-to-zero constraint on $\mathbf{U}_1$'s columns and a full-rank assumption on the covariates is sufficient to identify the covariate effects. We formalize this statement in the following proposition whose proof can be found in Section 5.8.4. Note that this proposition requires that the model in (5.1) extends to the diagonal elements of $\mathbf{Y}_t$, which is sometimes assumed in the LSM literature (MacDonald et al., [in press](#)).

Proposition 5.1 (Identifiability of Covariate Effects). *Suppose we observe a set of dyad-wise covariates $\{\mathbf{X}_{ijt}\}_{i \leq j}$ and define $\bar{\mathbf{X}}_t = (\sum_{j=1}^n \mathbf{X}_{1jt}, \dots, \sum_{j=1}^n \mathbf{X}_{njt})^T \in \mathbb{R}^{n \times p}$. Assume all dyads Y_{ijt} for $1 \leq i \leq j \leq n$ and $t = 1, \dots, T$ are drawn from model (5.1) with parameters $\{\boldsymbol{\beta}, \mathbf{U}_{1:T}, \mathbf{\Lambda}_{1:T}\}$ such that $\mathbf{U}_1 \in \tilde{\mathcal{V}}_{d,n}$ and $\text{rank}(\bar{\mathbf{X}}_1) = p$, e.g., $\bar{\mathbf{X}}_1$ is full column-rank, then $\boldsymbol{\beta}$ is identifiable. That is, if the probability distribution induced by two different parameterizations $\{\boldsymbol{\beta}, \mathbf{U}_{1:T}, \mathbf{\Lambda}_{1:T}\}$ and $\{\tilde{\boldsymbol{\beta}}, \tilde{\mathbf{U}}_{1:T}, \tilde{\mathbf{\Lambda}}_{1:T}\}$ coincide, then $\boldsymbol{\beta} = \tilde{\boldsymbol{\beta}}$.*

5.2.2 A Special Case: The Static Network Eigenmodel

In the special case that we observe a single static network, so that $t = 1$, the dynamic network eigenmodel reduces to a model similar to the static network eigenmodel introduced by P. D. Hoff (2008, 2009). Specifically, denote the single observed $n \times n$ binary adjacency matrix and the set of p -dimensional covariates as $\mathbf{Y}$ and $\{\mathbf{X}_{ij}\}_{i < j}$ with $\mathbf{X}_{ij} \in \mathbb{R}^p$, respectively. Conditioned on the model parameters and covariates, the static network eigenmodel assumes that for any $i < j$:

$$Y_{ij} = Y_{ji} \stackrel{\text{ind.}}{\sim} \text{Bernoulli}[\mathbb{P}(Y_{ij} = 1)], \quad \text{with} \quad (5.8)$$

$$\text{logit}[\mathbb{P}(Y_{ij} = 1)] = \boldsymbol{\beta}^T \mathbf{X}_{ij} + [\mathbf{U} \mathbf{\Lambda} \mathbf{U}^T]_{ij},$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$, $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_d) = \text{diag}(\boldsymbol{\lambda}) \in \mathbb{R}^{d \times d}$ is a diagonal matrix, and $\mathbf{U} \in \tilde{\mathcal{V}}_{d,n}$. Note that the original formulation of the static network eigenmodel in P. D. Hoff (2008, 2009) only includes an intercept, so $\mathbf{X}_{ij} = 1$, uses a probit link function, and does not require the columns of $\mathbf{U}$ to sum to zero.

In the static case, an additional benefit of the sum-to-zero constraint beyond parameter identifiability is that it gives a meaningful interpretation to the elements of $\boldsymbol{\beta}$. The existence of such an interpretation is not always possible in LSMs because the covariate effects are often confounded with the latent positions (Minhas et al., 2018). Under the sum-to-zero constraint, the elements of $\boldsymbol{\beta}$ are “average” log-odds

ratios, where the average is taken over all nodes in the network. To see this, we define $\bar{\mathbf{X}}_i = (1/n) \sum_{j=1}^n \mathbf{X}_{ij}$, the average of all dyad-wise covariates associated with node i . The sum-to-zero constraint implies that $(1/n) \sum_{j=1}^n \text{logit} [\mathbb{P}(Y_{ij} = 1)] = \boldsymbol{\beta}^T \bar{\mathbf{X}}_i$. Therefore, β_k is the average log-odds ratio associated with feature k , i.e., a unit increase in the average covariate $\bar{X}_{ik}$ associated with a node increases the average log-odds of that node forming an edge in the network by β_k .

5.2.3 A Prior Over $\tilde{\mathcal{V}}_{d,n}$ for $p(\mathbf{U}_1)$

An explicit prior density over centered semi-orthogonal matrices is difficult to construct due to the constraints placed on these matrices. In fact, we are unaware of any explicit prior density over $\tilde{\mathcal{V}}_{d,n}$ currently used for Bayesian inference. Instead, we use the following probabilistic construction based on the QR decomposition:

1. Sample $\mathbf{B}_1 \in \mathbb{R}^{n \times d}$ with entries independently drawn from a standard normal $N(0, 1)$.
2. Calculate the unique QR decomposition of $\tilde{\mathbf{B}}_1 = [\mathbf{1}_n, \mathbf{B}_1] \in \mathbb{R}^{n \times (d+1)}$, i.e., $\tilde{\mathbf{B}}_1 = \mathbf{Q}\mathbf{R}$, where $\mathbf{Q} \in \mathcal{V}_{d+1,n}$ and $\mathbf{R} \in \mathbb{R}^{(d+1) \times (d+1)}$ is an upper-triangular matrix with positive elements on its diagonal.
3. Set $\mathbf{U}_1$ to the last d columns of $\mathbf{Q}$, i.e., $\mathbf{U}_1 = \mathbf{Q}_{2:(d+1)}$.

This procedure generates a matrix $\mathbf{U}_1 \in \tilde{\mathcal{V}}_{d,n}$ since the columns of $\mathbf{Q}_{2:(d+1)}$ lie in the orthogonal complement of $\text{span}(\mathbf{Q}_1) = \text{span}(\mathbf{1}_n)$, where $\mathbf{Q}_1$ is the first column of $\mathbf{Q}$. To make the dependence of $\mathbf{U}_1$ on $\mathbf{B}_1$ explicit, we denote $\mathbf{Q}_{\mathbf{B}_1}$ as a matrix constructed in this fashion. Also, the following result holds.

Proposition 5.2. *The probability distribution of $\mathbf{Q}_{\mathbf{B}_1}$ has full support on $\tilde{\mathcal{V}}_{d,n}$.*

The proof is provided in the Section 5.8.4. This proposition states that the probability distribution induced by steps (1) – (3) assigns positive probability to all open neighborhoods of $\tilde{\mathcal{V}}_{d,n}$'s elements, which makes the distribution suitable as a prior when performing posterior inference over parameters in $\tilde{\mathcal{V}}_{d,n}$. Lastly, since this procedure is differentiable with respect to $\mathbf{B}_1$ (Walter et al., 2012), we can easily include it as a prior in gradient-based sampling methods such as the one we develop in Section 5.4.

5.2.4 A Matrix Langevin Distribution for $p(\mathbf{U}_t \mid \mathbf{U}_{t-1})$

To impose a smooth temporal dependence on the columns of $\mathbf{U}_t$ for $2 \leq t \leq T$, we assume the conditional densities in Equation (5.2) take the following form:

$$p(\mathbf{U}_t \mid \mathbf{U}_{t-1}, \phi) = Z(\phi)^{-1} \text{etr}(\phi \mathbf{U}_{t-1}^T \mathbf{U}_t), \quad (5.9)$$

where $\phi \in \mathbb{R}$, $\text{etr}(\mathbf{A}) = \exp(\text{tr}(\mathbf{A}))$, and the normalizing constant $Z(\phi) = {}_0F_1(\frac{1}{2}n, \frac{1}{4}\phi^2 I_d)$ is a type of hypergeometric function with matrix arguments. This distribution is known as the matrix Langevin distribution (Chikuse, 2003a) or the matrix von Mises-Fisher distribution (P. D. Hoff, 2009) and is defined with respect to the normalizing invariant measure on $\mathcal{V}_{d,n}$. Note that when $d = 1$, the matrix Langevin distribution reduces to the von Mises-Fisher distribution on the n -sphere, which is commonly used as a prior over unit-vectors. The following proposition outlines three properties of the matrix Langevin distribution that make it suitable as a prior for modeling Markov processes on the Stiefel manifold. The proof can be found in Section 5.8.4.

Proposition 5.3 (Properties of Conditional Matrix Langevin Priors). *Let $p(\mathbf{U}_t \mid \mathbf{U}_{t-1}, \phi)$ be the matrix Langevin distribution defined in Equation (5.9), then the following three properties hold: (1) If $\phi > 0$, then $p(\mathbf{U}_t \mid \mathbf{U}_{t-1}, \phi)$ has a unique mode at $\mathbf{U}_{t-1}$, (2) As $\phi \rightarrow \infty$, $\mathbf{U}_t$ converges to $\mathbf{U}_{t-1}$ in probability, (3) When $\phi = 0$, $p(\mathbf{U}_t \mid \mathbf{U}_{t-1}, \phi) = \text{Uniform}(\mathcal{V}_{d,n})$.*

Proposition 5.3 says that we should interpret ϕ as a concentration parameter in the sense that as ϕ increase, $\mathbf{U}_t$ tends to concentrate around the previous state $\mathbf{U}_{t-1}$. Conversely, when $\phi = 0$, $\mathbf{U}_t$ completely ignores the previous state, i.e., $\mathbf{U}_t \mid \mathbf{U}_{t-1} \stackrel{d}{=} \mathbf{U}_t$. To ensure $p(\mathbf{U}_t \mid \mathbf{U}_{t-1}, \phi)$ has a unique mode at $\mathbf{U}_{t-1}$, we place an $\text{Exponential}(b_\phi)$ prior on ϕ so that $\phi > 0$. Note that the evaluation of $Z(\phi)$ is problematic, which makes posterior inference for ϕ doubly-intractable. We address this issue in Section 5.4.2.

5.3 Dimension Selection

Now, we turn to the main contribution of this chapter which is the development of a Bayesian approach that can automatically identify the dimension of the latent space. Note that we will assume that the true dimension is fixed over time, which is an assumption made by previous Bayesian dynamic latent space models (Sewell & Chen, 2015b). In the context of network eigenmodels, the dimension selection problem is equivalent to estimating $\text{rank}(\mathbf{U}_t \mathbf{\Lambda}_t \mathbf{U}_t^T) = \|\mathbf{\Lambda}_t\|_0 \leq d$, i.e., the number of non-zero elements of $\mathbf{\Lambda}_t$. As such our approach begins with a spike-and-slab prior on the diagonal elements of $\mathbf{\Lambda}_t$ that induces posterior zeros with high-probability. Furthermore, we expect additional dimensions to play a progressively less important role in describing the network structure. As a solution, we introduce a novel ordered spike-and-slab prior that induces a stochastic ordering of $\mathbf{\Lambda}_t$'s elements and results in a posterior that asymptotically concentrates around the true dimension d_0 .

5.3.1 The Non-Homogeneous Spike-and-Slab Indian Buffet Process

To construct a prior with ordered shrinkage and dimension selection, we build upon the spike-and-slab priors (Ishwaran & Rao, 2005) and the Indian buffet process (Griffiths & Ghahramani, 2011). Specifically, let $\{\eta_h\}_{h=1}^d$ be a collection of d random variables. We consider the following spike-and-slab prior on $\{\eta_h\}_{h=1}^d$ with a shrinking sequence of slab probabilities $\theta_1 > \theta_2 > \dots > \theta_d$:

$$\begin{aligned} \eta_h \mid \theta_h &\stackrel{\text{ind.}}{\sim} \theta_h \mathbb{P}_{\text{slab}} + (1 - \theta_h) \mathbb{P}_{\text{spike}}, & \theta_h &= \prod_{\ell=1}^h \nu_\ell, & h &= 1, \dots, d, \\ \nu_1 &\stackrel{\text{ind.}}{\sim} \text{Beta}(\alpha, \kappa + 1), & \nu_\ell &\stackrel{\text{iid}}{\sim} \text{Beta}(\alpha, 1), & \ell &= 2, \dots, d, \end{aligned} \quad (5.10)$$

where $\alpha > 0$, $\kappa \geq 0$, and $\mathbb{P}_{\text{spike}}$ and $\mathbb{P}_{\text{slab}}$ are the distributions of the spike and slab components, respectively. Because our prior uses the stick-breaking construction of the Indian buffet process, we denote the prior in Equation (5.10) as the non-homogeneous spike-and-slab Indian buffet process prior truncated at d or $\text{SS-IBP}_d(\alpha, \kappa, \mathbb{P}_{\text{spike}}, \mathbb{P}_{\text{slab}})$. The hyperparameters of this prior are determined by $\mathbb{P}_{\text{slab}}$, $\mathbb{P}_{\text{spike}}$, and the scalars α and κ . We call this prior non-homogeneous because we draw ν_1 from a two-parameter Beta distribution and the remaining $\nu_{2:d}$ parameters from a one-parameter Beta distribution. We show this non-homogeneous IBP is sufficient to obtain posterior concentration around the true dimension when used as a prior for $\mathbf{\Lambda}$ for the static network eigenmodel.

The ordering of the slab probabilities $\{\theta_h\}_{h=1}^d$ induces a stochastic ordering of the parameters $\{\eta_h\}_{h=1}^d$ under the prior, which is formalized in the following proposition whose proof can be found in Section 5.8.4.

Proposition 5.4 (The SS-IBP Prior Induces a Stochastic Ordering). *For $\epsilon > 0$ and fixed $\eta_0 \in \mathbb{R}$, let $\mathbb{B}_\epsilon(\eta_0) = \{\eta_h : |\eta_h - \eta_0| \leq \epsilon\}$ denote the ϵ -ball centered at η_0 . Under the SS-IBP $_d(\alpha, \kappa, \mathbb{P}_{spike}, \mathbb{P}_{slab})$, $\mathbb{P}(|\eta_h - \eta_0| \leq \epsilon) < \mathbb{P}(|\eta_{h+1} - \eta_0| \leq \epsilon)$ whenever $\mathbb{P}_{slab}(\mathbb{B}_\epsilon(\eta_0)) < \mathbb{P}_{spike}(\mathbb{B}_\epsilon(\eta_0))$.*

Intuitively, for any η_0 that is favored by the spike distribution $\mathbb{P}_{spike}$ relative to the slab $\mathbb{P}_{slab}$, the SS-IBP $_d(\alpha, \kappa, \mathbb{P}_{spike}, \mathbb{P}_{slab})$ places more mass around η_0 as h increases. Note that we are most interested in $\eta_0 = 0$, which implies the prior places more mass around zero as h increases.

5.3.2 An SS-IBP Prior for $\Lambda_{1:T}$ in the Dynamic Network Eigenmodel

Now, we use the SS-IBP $_d(\alpha, \kappa, \mathbb{P}_{spike}, \mathbb{P}_{slab})$ to construct a prior for $\Lambda_{1:T}$ that promotes sparsity and induces a soft identifiability constraint that counteracts the permutation invariance of the network eigenmodel's likelihood. Let $\lambda_h = (\lambda_{h1}, \dots, \lambda_{hT})^T \in \mathbb{R}^T$ denote the time-series of each homophily coefficient. We assume each λ_h , $1 \leq h \leq d$, is drawn independently from the following hierarchy based on the hypervariance spike-and-slab specification of Ishwaran and Rao (2005):

$$\begin{aligned} Z_h &| \theta_h \sim \text{Bernoulli}(\theta_h), \\ \phi_h &| Z_h \sim Z_h \delta_1 + (1 - Z_h) \delta_{v_0}, \\ \sigma_h^2 &\sim \text{Exponential}(1/2b^2), \\ (\lambda_{h1}, \dots, \lambda_{hT}) &| \phi_h, \sigma_h^2, \rho \sim N\left(\lambda_{h1} | 0, \frac{\phi_h^2 \sigma_h^2}{1 - \rho^2}\right) \prod_{t=2}^T N(\lambda_{ht} | \rho \lambda_{ht-1}, \phi_h^2 \sigma_h^2), \end{aligned} \quad (5.11)$$

where $b > 0$ and δ_1 and δ_{v_0} denote point masses concentrated at 1 and v_0 with $0 \leq v_0 < 1$, respectively. Note that we require $|\rho| < 1$, which we enforce with a $(\rho + 1)/2 \sim \text{Beta}(2, 1)$ prior, so that each λ_h follows a stationary AR(1) process with variance $\phi_h^2 \sigma_h^2$. To enforce shrinkage, we draw the slab probabilities from a non-homogeneous IBP truncated at d :

$$\theta_h = \prod_{\ell=1}^h \nu_\ell, \quad \nu_1 \stackrel{\text{ind.}}{\sim} \text{Beta}(\alpha, \kappa + 1), \quad \nu_\ell \stackrel{\text{iid}}{\sim} \text{Beta}(\alpha, 1), \quad \ell = 2, \dots, d, \quad h = 1, \dots, d,$$

As the following corollary to Proposition 5.4 states, an important property of this prior is that it induces a stochastic ordering on the diagonal elements of Λ_t , which helps counteract the likelihood's invariance to column permutation.

Corollary 5.1. *For any $\epsilon > 0$, $\mathbb{P}(|\lambda_{ht}| \leq \epsilon) < \mathbb{P}(|\lambda_{(h+1)t}| \leq \epsilon)$ for $t = 1, \dots, T$, whenever $0 \leq v_0 < 1$.*

We provide the proof in Section 5.8.4. This stochastic ordering results from the fact that marginally $(\lambda_{1t}, \dots, \lambda_{dt}) | \rho \sim \text{SS-IBP}_d(\alpha, \kappa, \mathbb{P}_{spike}, \mathbb{P}_{slab})$ with $\mathbb{P}_{slab} = \text{Laplace}(b/\sqrt{1 - \rho^2})$ and $\mathbb{P}_{spike} = \text{Laplace}(v_0 b/\sqrt{1 - \rho^2})$, where $\text{Laplace}(b)$ denotes a Laplace distribution with mean zero and scale b .

5.3.3 A Special Case: A Prior for the Static Network Eigenmodel

For the static network eigenmodel, the prior for λ simplifies to

$$\begin{aligned} Z_h &| \theta_h \sim \text{Bernoulli}(\theta_h), \\ \phi_h &| Z_h \sim Z_h \delta_1 + (1 - Z_h) \delta_{v_0}, \end{aligned} \quad (5.12)$$

$$\sigma_h^2 \sim \text{Exponential}(1/2b^2),$$

$$\lambda_h \mid \phi_h, \sigma_h^2 \sim N(0, \phi_h^2 \sigma_h^2), \quad h = 1, \dots, d,$$

where we set $\rho = 0$ and we again draw the slab probabilities from a non-homogeneous IBP truncated at d

$$\theta_h = \prod_{\ell=1}^h \nu_\ell, \quad \nu_1 \stackrel{\text{ind.}}{\sim} \text{Beta}(\alpha, \kappa + 1), \quad \nu_\ell \stackrel{\text{iid}}{\sim} \text{Beta}(\alpha, 1), \quad \ell = 2, \dots, d, \quad h = 1, \dots, d.$$

Marginalizing over $Z_{1:d}$ and $\sigma_{1:d}^2$, $\boldsymbol{\lambda} \sim \text{SS-IBP}_d(\alpha, \kappa, \mathbb{P}_{\text{spike}}, \mathbb{P}_{\text{slab}})$ with $\mathbb{P}_{\text{slab}} = \text{Laplace}(b)$ and $\mathbb{P}_{\text{spike}} = \text{Laplace}(v_0 b)$. The following corollary, which is a special case of Corollary 5.1, demonstrates that this prior induces a stochastic ordering on the diagonal elements of $\boldsymbol{\Lambda}$ that helps counteract the permutation invariance of the likelihood.

Corollary 5.2. *For any $\epsilon > 0$, $\mathbb{P}(|\lambda_h| \leq \epsilon) < \mathbb{P}(|\lambda_{h+1}| \leq \epsilon)$, whenever $0 \leq v_0 < 1$.*

5.3.4 Theoretical Results for Dimension Selection using SS-IBPs

In this section, we investigate the posterior distribution's ability to select the dimension of the latent space for the static network eigenmodel, taking into account the permutation invariance of the likelihood. In particular, we develop theoretical results for the case that $v_0 = 0$ in Equation (5.12), which corresponds to using a discrete spike-and-slab prior with $\mathbb{P}_{\text{spike}} = \delta_0$, a point mass concentrated at 0, and $\mathbb{P}_{\text{slab}} = \text{Laplace}(b)$. This is also the choice we made for the static model in the simulation studies and real-data analysis in Section 5.5 and Section 5.6.

To begin, we show that the $\text{SS-IBP}_d(\alpha, \kappa, \mathbb{P}_{\text{spike}}, \mathbb{P}_{\text{slab}})$ prior induces an implicit prior on the number of non-zero elements of $\boldsymbol{\Lambda}$, i.e., $\|\boldsymbol{\Lambda}\|_0$, that favors low-dimensions. Formally, the next theorem shows that this induced prior has exponentially decaying tails when $\mathbb{P}_{\text{spike}} = \delta_0$ and we choose α and κ appropriately. The proof is provided in Section 5.8.4.

Theorem 5.1 (The Induced Prior on $\|\boldsymbol{\Lambda}\|_0$ has Exponentially Decaying Tails). *Let $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\lambda}) \in \mathbb{R}^{d \times d}$ with $\boldsymbol{\lambda} \sim \text{SS-IBP}_d(\alpha, d^{1+\delta}, \mathbb{P}_{\text{spike}}, \mathbb{P}_{\text{slab}})$ for $\alpha \in (0, 1)$, $\delta \geq 6/\log d$, and $\mathbb{P}_{\text{spike}} = \delta_0$, then for any $t \geq 1$,*

$$\mathbb{P}(\|\boldsymbol{\Lambda}\|_0 > t) \leq 2e^{-t(\delta/6) \log d}. \quad (5.13)$$

It has been widely observed in the literature on posterior asymptotics that an exponentially decaying tail is an important part of obtaining optimally behaving posteriors (Castillo & van der Vaart, 2012; Ročková & George, 2016). Note that Theorem 5.1 leaves the slab distribution unspecified; however, a sufficiently heavy-tailed slab is used in Theorem 5.2 to ensure property (5.13) transfers to the posterior.

Now, we proceed to demonstrate that the posterior concentrates on low-dimensions. First, we lay out some notation and assumptions. We assume that the adjacency matrix $\mathbf{Y}$ is generated from model (5.8) with non-zero latent space dimension d_0 and true parameters $\{\boldsymbol{\beta}_0, \mathbf{U}_0, \boldsymbol{\Lambda}_0\}$ where $\boldsymbol{\beta}_0 \in \mathbb{R}^p$, $\boldsymbol{\Lambda}_0 \in \mathbb{R}^{d_0 \times d_0}$, and $\mathbf{U}_0 \in \mathcal{V}_{d_0, n}$ with $\|\boldsymbol{\Lambda}_0\|_0 = d_0$. Also, we let $\mathbb{E}_0^{(n)}$ denote the expectation under this true data generating process. We consider the scenario that d_0 grows with n , but we assume p is fixed. As such, we must also increase the prior's truncation level d as n increases. In particular, we make the following assumptions:

$$(A5.1) \quad d = \lceil n^\gamma \rceil \text{ for some } \gamma \in (0, 1],$$

$$(A5.2) \quad d_0/\log d \rightarrow 0 \text{ as } n \rightarrow \infty,$$

(A5.3) For sufficiently large n , $b \leq Kd$ for a constant $K > 0$ independent of n ,

(A5.4) $\sup_{1 \leq h \leq d_0} |\lambda_{0h}| \leq M$ for some $M > 0$,

(A5.5) There exist positive constants c_1 , c_2 , and n_0 such that $c_1 n \leq \max_{1 \leq k \leq p} \|\mathbf{X}_k\|_F \leq c_2 n$ for all $n > n_0$, i.e., $\max_{1 \leq k \leq p} \|\mathbf{X}_k\|_F \asymp n$, where $\mathbf{X}_k \in \mathbb{R}^{n \times n}$ has elements $X_{ij,k}$.

(A5.1) forces d to grow with n . (A5.2) sets the growth rate of the truncation level d relative to the true latent space dimension d_0 , both of which hold under (A5.1) for the often assumed case that d_0 is bounded. Assumption (A5.3) constrains the scale of the Laplace(b) slab. This assumption holds for the case that b is a constant or grows as a fractional power of n , such as our choice of $b = \sqrt{n/2}$ in the following sections. Under model (5.8), (A5.4) is equivalent to bounding the values of the latent space term $\mathbf{U}_0 \mathbf{\Lambda}_0 \mathbf{U}_0^T$, which is common in the network latent space literature (Z. Ma et al., 2020). Lastly, (A5.5) holds for the traditional case that the model only includes an intercept or the case where the covariates are bounded.

The following theorem states that the average posterior probability that $\|\mathbf{\Lambda}\|_0$ “overshoots” a constant multiple of the true latent space dimension d_0 goes to zero as both the number of nodes n and prior truncation level d simultaneously increase. The proof is presented in Section 5.8.4.

Theorem 5.2 (The Posterior Concentrates on Low Dimensions in Expectation). *Assume model (5.8) with non-zero latent space dimension d_0 and true parameters $\{\beta_0, \mathbf{\Lambda}_0, \mathbf{U}_0\}$ such that $\|\mathbf{\Lambda}_0\|_0 = d_0$. Assume the following prior: $\boldsymbol{\lambda} \sim SS-IBP_d(1/d, d^{1+\delta}, \mathbb{P}_{spike}, \mathbb{P}_{slab})$ for $\delta > 0$, $\mathbb{P}_{slab} = N(0, \sigma_h^2)$, $\mathbb{P}_{spike} = \delta_0$, $\sigma_h^2 \stackrel{iid}{\sim} \text{Exponential}(1/2b^2)$ for $b \geq 1$, $\beta \sim N(\mathbf{0}_p, \sigma_\beta^2 I_p)$, and $\mathbf{U} \in \mathcal{V}_{d,n}$ with prior probability one. Assume (A5.1) - (A5.5) hold, then*

$$\lim_{n \rightarrow \infty} \mathbb{E}_0^{(n)} \mathbb{P} \left[\|\mathbf{\Lambda}\|_0 > Cd_0 \mid \mathbf{Y} \right] = 0,$$

for some $C > 1$ that only depends on δ and M .

Practically, Theorem 5.2 justifies the following strategy for estimating d_0 . Without knowledge of d_0 , one would set the prior truncation level d as large as possible with the hope that the shrinkage prior will select an appropriate dimension. An objection to this strategy is that employing too many latent dimensions may cause the posterior to over-estimate the dimension of the latent space due to over-fitting. Theorem 5.2 alleviates this concern by showing that the posterior will asymptotically concentrate on low-dimensional latent spaces near the true dimension d_0 .

5.4 Estimation

Here, we develop a Markov chain Monte Carlo (MCMC) algorithm to sample from the dynamic network eigenmodel’s posterior. Note that the MCMC algorithm developed here also applies to the static network eigenmodel by setting $t = 1$ in the following discussion. Due to the high-dimensionality of the latent variables, we use gradient-based sampling methods. Specifically, we propose a Metropolis-within-Gibbs sampler with Hamiltonian Monte Carlo (HMC) proposals. However, the requirement that $\mathbf{U}_{1:T}$ lies on the Stiefel manifold poses a challenge for gradient-based samplers because they often require the parameters to lie on an unconstrained space. As such, we first propose various parameter expansion strategies that overcome this limitation.

5.4.1 Polar Expansion

To facilitate efficient MCMC on the Stiefel manifold, we extend the polar expansion approach introduced in Jauch et al. (2021) to temporally dependent semi-orthogonal matrices. The idea is to introduce an additional set of unconstrained random matrices $\mathbf{B}_t \in \mathbb{R}^{n \times d}$ for $2 \leq t \leq T$ such that the orthogonal components of the polar decomposition of $\mathbf{B}_2, \dots, \mathbf{B}_T$ follow the distribution in Equation (5.9). The polar decomposition is the unique representation of a full rank matrix $\mathbf{B}_t$ as the product $\mathbf{B}_t = \mathbf{U}_{\mathbf{B}_t} \mathbf{S}_{\mathbf{B}_t}^{1/2}$, where $\mathbf{U}_{\mathbf{B}_t} = \mathbf{B}_t (\mathbf{B}_t^T \mathbf{B}_t)^{-1/2} \in \mathcal{V}_{d,n}$ and $\mathbf{S}_{\mathbf{B}_t} = \mathbf{B}_t^T \mathbf{B}_t$ is a $d \times d$ symmetric positive definite matrix. Note that the mapping from a full rank $\mathbf{B}_t$ to $(\mathbf{U}_{\mathbf{B}_t}, \mathbf{S}_{\mathbf{B}_t})$ is one-to-one, so that the density of $\mathbf{B}_{2:T}$ can be derived from the joint density of $(\mathbf{U}_{\mathbf{B}_{2:T}}, \mathbf{S}_{\mathbf{B}_{2:T}})$ via a change-of-variables. In the previous expression, we are using the notation $\mathbf{U}_{\mathbf{B}_{2:T}} = (\mathbf{U}_{\mathbf{B}_2}, \dots, \mathbf{U}_{\mathbf{B}_T})$ and $\mathbf{S}_{\mathbf{B}_{2:T}} = (\mathbf{S}_{\mathbf{B}_2}, \dots, \mathbf{S}_{\mathbf{B}_T})$. In particular, we have that

$$\begin{aligned} p(\mathbf{B}_{2:T}) &= p(\mathbf{U}_{\mathbf{B}_{2:T}}, \mathbf{S}_{\mathbf{B}_{2:T}}) \times \prod_{t=2}^T |J(\mathbf{B}_t \rightarrow (\mathbf{U}_{\mathbf{B}_t}, \mathbf{S}_{\mathbf{B}_t}))| \\ &= p(\mathbf{S}_{\mathbf{B}_{2:T}} \mid \mathbf{U}_{\mathbf{B}_{2:T}}) p(\mathbf{U}_{\mathbf{B}_{2:T}}) \times \prod_{t=2}^T |J(\mathbf{B}_t \rightarrow (\mathbf{U}_{\mathbf{B}_t}, \mathbf{S}_{\mathbf{B}_t}))|, \end{aligned}$$

where $|J(\mathbf{B}_t \rightarrow (\mathbf{U}_{\mathbf{B}_t}, \mathbf{S}_{\mathbf{B}_t}))|$ denotes the determinant of the Jacobian of the transformation from $\mathbf{B}_t$ to the components of its polar decomposition:

$$|J(\mathbf{B}_t \rightarrow (\mathbf{U}_{\mathbf{B}_t}, \mathbf{S}_{\mathbf{B}_t}))| = \frac{\Gamma_n(d/2)}{\pi^{dn/2}} |\mathbf{S}_{\mathbf{B}_t}|^{-\frac{n-d-1}{2}},$$

which is derived in Chikuse (2003a). The product over $T - 1$ Jacobian terms is due to the block-diagonal nature of the Jacobian transformation across time.

To complete the expansion, we set $p(\mathbf{U}_{\mathbf{B}_{2:T}})$ to the joint distribution described in Equation (5.9). However, we have freedom in the choice of $p(\mathbf{S}_{\mathbf{B}_{2:T}} \mid \mathbf{U}_{\mathbf{B}_{2:T}})$. Similar to Jauch et al. (2021), we chose it to be a collection of independent Wishart distributions with n degrees of freedom and scale matrix I_d , i.e., $W_d(n, I_d)$, which has the added benefit of canceling the Jacobian terms. The result of this choice is that

$$p(\mathbf{B}_{2:T}) = \prod_{t=2}^T (2\pi)^{-dn/2} \text{etr}(-\mathbf{B}_t^T \mathbf{B}_t / 2) \times p(\mathbf{U}_{\mathbf{B}_2} \mid \mathbf{Q}_{\mathbf{B}_1}, \phi) \prod_{t=3}^T p(\mathbf{U}_{\mathbf{B}_t} \mid \mathbf{U}_{\mathbf{B}_{t-1}}, \phi),$$

where $\mathbf{Q}_{\mathbf{B}_1}$ is the centered semi-orthogonal matrix described in Section 5.2.3.

In summary, we construct a Markov chain targeting a posterior distribution whose $\mathbf{U}_{\mathbf{B}_{2:T}}$ marginals are the posterior distribution of $\mathbf{U}_{2:T}$ by construction. Note that MCMC on the unconstrained space typically has a higher effective sample size per iteration than constrained samplers such as geodesic Monte Carlo (Byrne & Girolami, 2013).

5.4.2 An Asymptotic Approximation to $Z(\phi)$

As discussed in Section 5.2.4, for general d , the exact calculation of the normalizing constant $Z(\phi) = {}_0F_1(\frac{1}{2}n, \frac{1}{4}\phi^2 I_d)$ is unavailable or too slow for use in MCMC algorithms. However, various asymptotic ap-

proximations exist (Chikuse, 2006; Muirhead, 1978). We approximate ${}_0F_1(\frac{1}{2}n, \frac{1}{4}\Phi^2)$ by the first order term

$${}_0\tilde{F}_1(\frac{1}{2}n, \frac{1}{4}\Phi^2) = \frac{\Gamma_d(\frac{1}{2}n)}{2^d \pi^{nd/2}} \exp\left(\sum_{r=1}^d \Phi_{rr}\right) \prod_{r=1}^d \left(\frac{2\pi}{\Phi_{rr}}\right)^{\frac{1}{2}(n-d)} \prod_{s < r}^d \left(\frac{2\pi}{\Phi_{ss} + \Phi_{rr}}\right)^{\frac{1}{2}},$$

where $\Phi \in \mathbb{R}^{d \times d}$ is a diagonal matrix and $\Gamma_d(a)$ is the multivariate gamma function. We use the above approximation in place of $Z(\phi)$ for the MCMC computations. As such, our MCMC algorithm targets a posterior with a new prior for ϕ given by $\tilde{p}(\phi) \propto p(\phi) \left[{}_0\tilde{F}_1(n/2, \phi^2/4I_d)/{}_0F_1(n/2, \phi^2/4I_d)\right]^{(T-1)}$, where $p(\phi)$ is an Exponential(b_ϕ) density. In other words, our inference remains valid under a re-weighted prior. Results from Muirhead (1978) show that

$$\frac{{}_0\tilde{F}_1(\frac{1}{2}n, \frac{1}{4}\phi^2 I_d)}{{}_0F_1(\frac{1}{2}n, \frac{1}{4}\phi^2 I_d)} = 1 + \left[\frac{4d(d-2) - (n-d)(n-d-2)d}{32} \right] \frac{1}{\phi} + O(\phi^{-2}).$$

As such, we could correct this bias using importance sampling based on higher-order terms or exact calculations; however, we will not explore such post-processing in this work.

5.4.3 Non-Centered Parameterizations and Change-of-Variables

To improve the mixing of our MCMC algorithm, we used non-centered parameterizations (Papaspiliopoulos et al., 2007) for $\Lambda_{1:T}$, $\gamma_{3:T}$, and $\alpha_{3:T}$. Starting with Equation (5.11), we marginalize over $\phi_{1:d}$ and re-express λ_h in terms of a new set of standard normal random variables $\tilde{\lambda}_{h1}, \dots, \tilde{\lambda}_{hT} \stackrel{\text{iid}}{\sim} N(0, 1)$ for $1 \leq h \leq d$ as follows:

$$\lambda_{h1} \mid Z_h, \sigma_h, \rho = \frac{[Z_h + (1 - Z_h)v_0] \sigma_h}{\sqrt{1 - \rho^2}} \tilde{\lambda}_{h1}, \quad (5.14)$$

$$\lambda_{ht} \mid \lambda_{h(t-1)}, Z_h, \sigma_h, \rho = \rho \lambda_{h(t-1)} + [Z_h + (1 - Z_h)v_0] \sigma_h \tilde{\lambda}_{ht}, \quad t = 2, \dots, T. \quad (5.15)$$

Under this non-centered parameterization, we shift our inference to $\tilde{\lambda}_{h1}, \dots, \tilde{\lambda}_{hT}$, and reconstruct λ_h through Equations (5.14) and (5.15). From now on, to make the dependence of Λ_t on $Z_{1:d}$ clear, we denote $\Lambda_t(Z_{1:d}) = \text{diag}(\lambda_{1t}(Z_1), \dots, \lambda_{dt}(Z_d))$, where $\lambda_{1t}(Z_1), \dots, \lambda_{dt}(Z_d)$ are defined by the left hand side of Equations (5.14) and (5.15). Note that the elements of $\Lambda_{1:T}(Z_{1:d})$ have the same prior distribution as defined in Equation (5.11). Similarly, we let $\tilde{\gamma}_3, \dots, \tilde{\gamma}_T \stackrel{\text{iid}}{\sim} N(0, 1)$ and $\tilde{\alpha}_3, \dots, \tilde{\alpha}_T \stackrel{\text{iid}}{\sim} N(0, 1)$ and define

$$\gamma_t = \gamma_{t-1} + \sigma_\gamma \tilde{\gamma}_t, \quad \alpha_t = \alpha_{t-1} + \sigma_\alpha \tilde{\alpha}_t, \quad t = 3, \dots, T.$$

which have the same autoregressive prior as defined in Equations (5.5) and (5.6).

Finally, we apply various change-of-variable transformations to the remaining parameters that have a constrained domain. Specifically, we apply a log-transformation to $\phi, \sigma_\gamma, \sigma_\alpha$ and σ_h for $h = 1, \dots, d$. We also apply a logistic transformation to $(\rho+1)/2$ and ν_h for $h = 1, \dots, d$. Note that we correct these transformations with Jacobian adjustments to the posterior. See Section 5.8.1 in the supplementary material for details.

5.4.4 The Metropolis-within-Gibbs Sampler

Here we outline the MCMC algorithm that we use to sample from the dynamic network eigenmodel's posterior. Let $\psi = \{\mathbf{Q}_{\mathbf{B}_1}, \mathbf{U}_{\mathbf{B}_{2:T}}, \Lambda_{1:T}(Z_{1:d}), \gamma_{2:T}, \alpha_{2:T}, \beta, \sigma_{1:d}^2, \rho, \phi, \sigma_\gamma^2, \sigma_\alpha^2, \theta_{1:d}\}$ denote the collection of model

parameters excluding the dimension indicator variables $Z_{1:d}$. Overall, the algorithm is a Gibbs sampler that alternates between sampling ψ and $Z_{1:d}$ from their full conditional distribution $p(\psi \mid Z_{1:d}, \mathbf{Y}_{1:T})$ and $p(Z_{1:d} \mid \psi, \mathbf{Y}_{1:T})$, respectively. To sample from $p(\psi \mid Z_{1:d}, \mathbf{Y}_{1:T})$, we use Hamiltonian Monte Carlo (HMC) with adaptive tuning parameter selection (Hoffman & Gelman, 2014; Neal, 2011) as implemented in NumPyro (Bingham et al., 2019; Phan et al., 2019). To sample from $p(\psi \mid Z_{1:d}, \mathbf{Y}_{1:T})$ using NumPyro’s HMC routine, we must provide the logarithm of this density up to an additive constant, which can be found in Section 5.8.1 of the supplementary material. The full Metropolis-within-Gibbs algorithm is outlined in Algorithm 5.1. The model parameters are initialized according to the scheme outlined in Section 5.8.2 of the supplementary material.

Given the previous parameters $\psi^{(s)}$ and $Z_{1:d}^{(s)}$, update the current parameters as follows:

1. Sample $\psi^{(s+1)}$ from $p(\psi \mid Z_{1:d}^{(s)}, \mathbf{Y}_{1:T})$ defined in Equation (5.17) of the supplementary material using HMC with adaptive tuning.

2. Let $\pi \in \text{Permutation}(\{1, \dots, d\})$ and set $Z_{1:d}^{(s+1)} = Z_{1:d}^{(s)}$. For $h = 1, \dots, d$:

Sample $Z_{\pi(h)}^{(s+1)}$ from $p(Z_{\pi(h)} \mid Z_{-\pi(h)}^{(s+1)}, \psi^{(s+1)}, \mathbf{Y}_{1:T})$, where

$$\text{odds}(Z_{\pi(h)} = 1 \mid Z_{-\pi(h)}^{(s+1)}, \psi^{(s+1)}, \mathbf{Y}_{1:T}) = \frac{p(\mathbf{Y}_{1:T} \mid \boldsymbol{\Theta}_{1:T}(Z_{1:d}^{(s+1)}), Z_{\pi(h)} = 1, Z_{-\pi(h)})}{p(\mathbf{Y}_{1:T} \mid \boldsymbol{\Theta}_{1:T}(Z_{1:d}^{(s+1)}), Z_{\pi(h)} = 0, Z_{-\pi(h)})} \times \frac{\theta_{\pi(h)}^{(s+1)}}{1 - \theta_{\pi(h)}^{(s+1)}}.$$

For $t = 1$:

$$[\boldsymbol{\Theta}_1(Z_{1:d})]_{ij} = \beta^{(s+1)\text{T}} \mathbf{X}_{ij1} + \left[\mathbf{Q}_{\mathbf{B}_1}^{(s+1)} \boldsymbol{\Lambda}_1^{(s+1)}(Z_{1:d}) \mathbf{Q}_{\mathbf{B}_1}^{(s+1)\text{T}} \right]_{ij}.$$

For $2 \leq t \leq T$:

$$[\boldsymbol{\Theta}_t(Z_{1:d})]_{ij} = \beta^{(s+1)\text{T}} \mathbf{X}_{ijt} + \gamma_t^{(s+1)} y_{ij(t-1)} + \alpha_t^{(s+1)} (1 - y_{ij(t-1)}) + \left[\mathbf{U}_{\mathbf{B}_t}^{(s+1)} \boldsymbol{\Lambda}_t^{(s+1)}(Z_{1:d}) \mathbf{U}_{\mathbf{B}_t}^{(s+1)\text{T}} \right]_{ij}.$$

The likelihood terms correspond to model (5.1).

Note that $\text{Permutation}(\{1, \dots, d\})$ denotes a different random permutation of $\{1, \dots, d\}$ at each iteration. Also, we use Z_{-h} to indicate the collection of $Z_{1:d}$ with the h th element removed, i.e., $(Z_1, \dots, Z_{h-1}, Z_{h+1}, \dots, Z_d)$.

Algorithm 5.1: Metropolis-within-Gibbs Sampler for the Dynamic Network Eigenmodel

5.4.5 Post-Processing to Account for Posterior Invariances

Although our use of an $\text{SS-IBP}_d(\alpha, \kappa, \mathbb{P}_{\text{spike}}, \mathbb{P}_{\text{slab}})$ prior places soft identifiability constraints on the parameters, the posterior remains invariant to signed-permutation of $\mathbf{U}_t$ ’s columns for $1 \leq t \leq T$, i.e., $\mathbf{U}_t \rightarrow \mathbf{U}_t \text{diag}(\mathbf{s})\mathbf{P}$ where $\mathbf{s} \in \{-1, 1\}^d$ and $\mathbf{P}$ is a permutation matrix on d elements. This can lead to inflated variance estimates without proper post-processing. As such, we post-process the results by matching each posterior draw $\mathcal{U}^{(s)} = (\mathbf{U}_1^{(s)\text{T}}, \dots, \mathbf{U}_T^{(s)\text{T}})^{\text{T}} \in \mathbb{R}^{nT \times d}$ to some reference trajectory $\mathcal{U}_0 \in \mathbb{R}^{nT \times d}$. For convenience, we choose $\mathcal{U}_0$ to be the maximum a posteriori (MAP) estimator.

Similar to Procrustes matching commonly employed in the latent space literature, which minimizes the Frobenius norm over rotation matrices, we solve

$$\tilde{\mathcal{U}}^{(s)} = \arg \min_{\mathbf{s} \in \{-1, 1\}^d, \mathbf{P} \in \Pi_d} \|\mathcal{U}_0 - \mathcal{U}^{(s)} \text{diag}(\mathbf{s})\mathbf{P}\|_F^2,$$

$$= \arg \max_{\mathbf{s} \in \{-1, 1\}^d, \mathbf{P} \in \Pi_d} \text{tr}(\mathbf{P} \mathcal{U}_0^T \mathcal{U}^{(s)} \text{diag}(\mathbf{s})). \quad (5.16)$$

where Π_d is the set of permutation matrices on d elements. We recognize the second expression as a linear assignment problem (Gower & Dijkstra, 2004) with a $d \times d$ cost matrix C with elements

$$C_{ij} = \max \left(\mathcal{U}_{0i}^T \mathcal{U}_j^{(s)}, -\mathcal{U}_{0i}^T \mathcal{U}_j^{(s)} \right),$$

where $\mathcal{U}_{0i}$ and $\mathcal{U}_j^{(s)}$ are the i th and j th columns of $\mathcal{U}_0$ and $\mathcal{U}^{(s)}$, respectively. Therefore, the solution to Equation (5.16) can be efficiently found in $O(d^3)$ time with algorithms such as the Hungarian method (Kuhn, 1955). In summary, for each posterior draw, we solve the above linear assignment problem to obtain $\mathbf{P}$ and set $\mathbf{s} = \text{sign}(\text{diag}(\mathcal{U}_0^T \mathcal{U}^{(s)} \mathbf{P}))$. Note that we also apply the permutation matrix $\mathbf{P}$ to the diagonal elements of $\mathbf{\Lambda}_t^{(s)}$ for $1 \leq t \leq T$.

5.4.6 Point Estimators for $\mathbf{U}_{1:T}$

Because the posterior mean of $\mathbf{U}_t$ may not lie on the Stiefel manifold, it does not provide a meaningful point estimator for $\mathbf{U}_t$. To define a point estimator that lies on the Stiefel manifold, we take a Bayesian decision theoretic approach and minimize the following constrained Frobenius norm under the posterior:

$$\arg \min_{\mathbf{W}_t \in \mathcal{V}_{d,n}} \mathbb{E} [\|\mathbf{W}_t - \mathbf{U}_t\|_F \mid \mathbf{Y}_{1:T}] = \arg \max_{\mathbf{W}_t \in \mathcal{V}_{d,n}} \text{tr}(\mathbb{E} [\mathbf{U}_t \mid \mathbf{Y}_{1:T}]^T \mathbf{W}_t).$$

If we view the Stiefel manifold as an embedded manifold of nd -dimensional Euclidean space, then the solution is the posterior Fréchet mean of $\mathbf{U}_t$. The above expression is minimized by the orthogonal component of the polar decomposition of $\mathbb{E} [\mathbf{U}_t \mid \mathbf{Y}_{1:T}]$ (Gower & Dijkstra, 2004), i.e., $\mathbf{W}_t = \mathbf{ZV}^T$ where $\mathbb{E} [\mathbf{U}_t \mid \mathbf{Y}_{1:T}] = \mathbf{ZDV}^T$ is the singular value decomposition (SVD) of $\mathbb{E} [\mathbf{U}_t \mid \mathbf{Y}_{1:T}]$.

5.5 Simulation Studies

In this section, we present simulation studies that assess different properties of the static and dynamic network eigenmodel and our estimation method. Overall, we performed three studies to evaluate the following tasks: (1) the ability of the proposed MCMC algorithm to recover the model parameters, (2) the accuracy of the SS-IBP $_d(\alpha, \kappa, \mathbb{P}_{\text{spike}}, \mathbb{P}_{\text{slab}})$ prior in recovering the true dimension of the latent space compared to existing methods, and (3) the predictive performance of the proposed dynamic network eigenmodel compared to existing latent space models for dynamic networks.

5.5.1 Competing Methods

For dimension selection, we compared our proposed spike-and-slab prior with two alternative selection criteria commonly used in the latent space literature: minimization of information criteria and K-fold cross-validation. For the information criteria, we considered the Watanabe-Akaike information criterion (WAIC) and the deviance information criterion (DIC). The K-fold cross-validation (CV) scheme used $K = 4$ folds, where the folds define a non-overlapping partition of the network's dyads, $\{(i, j) : 1 \leq i < j \leq n\}$. For dynamic networks, a fold includes the same group of dyad over all observed time periods. We selected the model that maximized the held-out log-likelihood averaged across all folds. Note that these methods select the

dimension for network eigenmodels without the spike-and-slab prior, i.e., we set $\phi_h = 1$ in Equation (5.11) and Equation (5.12). In addition, for the dynamic model, we use a $\sigma_h^2 \stackrel{\text{ind.}}{\sim} \Gamma^{-1}(5, 25n)$ prior. For the static model, we use the model proposed by P. D. Hoff (2009), which sets $\sigma_1^2 = \dots = \sigma_h^2 = \sigma^2 \sim N(0, n)$. We estimated these models by running NumPyro’s Hamiltonian Monte Carlo algorithm for 5,000 iterations after a burn-in of 5,000 iterations. For more details on how we calculated the information criteria and performed cross-validation, see Section 5.8.3 of the supplementary material.

To compare predictive performance, we considered two competing models based on Sewell and Chen (2015b)’s approach to modeling dynamic networks. The general model is as follows:

$$\begin{aligned} Y_{ijt} &= Y_{jit} \stackrel{\text{ind.}}{\sim} \text{Bernoulli}(\text{logit}^{-1}[\beta_0 + g(\mathbf{u}_{it}, \mathbf{u}_{jt})]), \quad \text{with} \\ (\mathbf{u}_{i1}, \dots, \mathbf{u}_{iT}) &\stackrel{\text{ind.}}{\sim} N(\mathbf{u}_{i1} \mid 0, \tau^2 I_d) \prod_{t=2}^T N(\mathbf{u}_{it} \mid \mathbf{u}_{i(t-1)}, \sigma^2 I_d), \quad 1 \leq i \leq n, \\ \beta_0 &\stackrel{\text{ind.}}{\sim} N(0, 100), \quad \tau \stackrel{\text{ind.}}{\sim} C^+(0, 1), \quad \sigma \stackrel{\text{ind.}}{\sim} C^+(0, 1), \end{aligned}$$

where $g(\cdot, \cdot)$ is a similarity function. For the similarity function, we considered two popular choices: (1) a bilinear function $g(\mathbf{u}_{it}, \mathbf{u}_{jt}) = \mathbf{u}_{it}^T \mathbf{u}_{jt}$ and (2) the negative squared Euclidean distance $g(\mathbf{u}_{it}, \mathbf{u}_{jt}) = -\|\mathbf{u}_{it} - \mathbf{u}_{jt}\|_2^2$. We denote these two models as the bilinear model and the squared distance model, respectively. Note that these methods explicitly assume latent features exhibit positive homophily. We estimated these models by running NumPyro’s Hamiltonian Monte Carlo algorithm for 5,000 iterations after a burn-in of 5,000 iterations.

Lastly, we estimated the proposed network eigenmodels with SS-IBP priors by running the Metropolis-within-Gibbs algorithm proposed in Section 5.4 for 5,000 iterations after a burn-in of 5,000 iterations. To match the theory developed in Section 5.3.4, we set $\alpha = 1/d$ and $\kappa = d^{1+\delta}$. Furthermore, we used a truncation level of $d = 10$ in all simulations. For the remaining hyperparameters, we set $b = \sqrt{n/2}$, $b_\phi = n$, $\delta = 0.1$, $\sigma_{0,\alpha} = 10$, $\sigma_{0,\gamma} = 10$, and $\sigma_\beta = 10$, which resulted in relatively broad priors. Lastly, for the dynamic network eigenmodel we set $v_0 = 0.001$ and for the static model we set $v_0 = 0$ to match the theory developed in Section 5.3.4.

5.5.2 Parameter Recovery

This section presents results that demonstrate the ability of the MCMC algorithm proposed in Section 5.4 to recover the model parameters as the number of nodes increases. For $n = 50, 100, 200$ and $T = 5$, we generated 50 synthetic networks for each value of n . For each network, we set the true dimension $d_0 = 4$ and sampled the latent positions $\mathbf{U}_{1:T}$ from their priors with $\phi = 10n$, which resulted in an average trace correlation between $\mathbf{U}_t$ and $\mathbf{U}_{t-1}$, $\text{tr}(\mathbf{U}_t^T \mathbf{U}_{t-1})/d_0$, of 0.95. We set $\boldsymbol{\lambda}_1 = n \text{diag}(2b_1, 1.5b_2, b_3, 0.75b_4)^T$, where each $b_h \stackrel{\text{iid}}{\sim} (1/2)\delta_1 + (1/2)\delta_{-1}$. Next, we evolved the homophily coefficients over time as $\boldsymbol{\lambda}_t = 0.95\boldsymbol{\lambda}_{t-1} + N(0, 0.1nI_d)$ for $2 \leq t \leq T$. In addition, we included edge (non-edge) persistence parameters: $\gamma_t = 1$ and $\alpha_t = -0.5$ for $2 \leq t \leq 5$. Lastly, we included an intercept, $\beta_0 = -2.5$, and two dyad-wise covariates, where $\mathbf{X}_{ij} \stackrel{\text{iid}}{\sim} N(0, I_2)$, with $\beta_1 = 0.5$ and $\beta_2 = -0.25$. This resulted in networks with a time-averaged density of 0.19.

To evaluate the estimators’ accuracy, we compared the top four dimensions ranked by their posterior inclusion probabilities, $\hat{\mathbb{P}}(Z_h = 1 \mid \mathbf{Y}_{1:T}) = (1/S) \sum_{s=1}^S \mathbb{1}\{Z_h^{(s)} = 1\}$, to the ground truth. Our evaluation metrics included the time-averaged trace correlation between the ground truth and estimated latent vectors, i.e., $(1/T) \sum_{t=1}^T \text{tr}(\mathbf{U}_t^T \hat{\mathbf{U}}_t)/d_0$, and the time-averaged relative error with respect to the squared Frobenius

norm between the true and estimated latent positions $\mathbf{U}_{1:T}$, homophily parameters $\mathbf{\Lambda}_{1:T}$, log-odds probability matrices $\mathbf{\Theta}_{1:T}$, covariates, and edge (non-edge) persistence parameters. Note that we estimated the log-odds matrix based on all ten latent dimensions.

Table 5.1 and Table 5.2 displays the results. Note that the trace correlation ranges from -1 to 1, with one indicating a perfect alignment of the subspaces spanned by $\mathbf{U}_t$ and $\hat{\mathbf{U}}_t$. For the relative errors, smaller is better. In all cases, the MCMC algorithm is able to recover the model’s parameters with high accuracy. Furthermore, the performance increases with the number of nodes. As such, we can conclude that the MCMC algorithm is successful at estimating the model parameters.

n	$\mathbf{U}_{1:T}$ Trace Correlation	$\mathbf{U}_{1:T}$ Rel. Error	$\mathbf{\Lambda}_{1:T}$ Rel. Error	$\mathbf{\Theta}_{1:T}$ Rel. Error
50	0.930 (0.019)	0.141 (0.038)	0.00865 (0.00531)	0.0753 (0.0050)
100	0.963 (0.003)	0.0732 (0.0070)	0.00263 (0.00130)	0.0491 (0.0022)
200	0.979 (0.001)	0.0428 (0.0022)	0.000729 (0.000313)	0.0328 (0.00073)

Table 5.1: Trace correlation and relative estimation error of the model’s parameters as the number of nodes (n) in the networks increased. Each cell contains the mean value of the metric with the standard deviation over the 50 simulation displayed in parenthesis.

n	β_0 Rel. Error	β_1 Rel. Error	β_2 Rel. Error	$\gamma_{2:T}$ Rel. Error	$\alpha_{2:T}$ Rel. Error
50	0.00243 (0.00289)	0.0139 (0.0200)	0.0369 (0.0416)	0.0345 (0.0313)	0.0908 (0.0968)
100	0.000846 (0.00124)	0.00188 (0.00256)	0.0107 (0.00157)	0.0115 (0.0119)	0.0190 (0.0243)
200	0.000325 (0.00040)	0.000519 (0.000734)	0.00241 (0.00362)	0.00391 (0.00420)	0.00814 (0.00963)

Table 5.2: Relative estimation error of the covariate effects and edge (non-edge) persistence parameters as the number of nodes (n) in the networks increased. Each cell contains the mean value of the metric with the standard deviation over the 50 simulation displayed in parenthesis.

5.5.3 Dimension Selection

Here, we assess the ability of the proposed $\text{SS-IBP}_d(\alpha, \kappa, \mathbb{P}_{\text{spike}}, \mathbb{P}_{\text{slab}})$ prior to select the correct number of dimensions for both the static and dynamic network eigenmodel. For both the static and dynamic networks, we generated 50 synthetic networks from a model with $d_0 = 4$ latent dimensions for different values of $n \in \{100, 200, 300\}$. The ground truth for the static network set $\mathbf{\Lambda} = n \text{diag}(2.5, -2, -1.5, 1)$ and drew $\mathbf{U}$ from its prior. For the covariates, we only included an intercept $\beta_0 = -3.5$, which resulted in static networks with an average density of 0.17. For the dynamic networks, we generated the networks in the same way as in the previous section; however, we did not include dyad-wise covariates and the edge (non-edge) persistence parameters. The edge (non-edge) persistence parameters were removed to satisfy the conditional independence assumption necessary for the competing WAIC estimator. See Section 5.8.3 of the supplementary material for details.

First, we analyze the performance of the static model. Figure 5.1 contains boxplots of the estimated posterior probability of d_0 , where $\hat{\mathbb{P}}(d_0 = k \mid \mathbf{Y}) = (1/S) \sum_{s=1}^S \mathbb{1}\{\sum_{h=1}^d Z_h^{(s)} = k\}$, over the 50 simulations for different values of n . In all cases, the SS-IBP prior places the highest posterior probability on the correct dimension $d_0 = 4$. Figure 5.2 shows the fraction of times a dimension was selected by the proposed SS-IBP and its competitors. Under all scenarios, the competing estimators (DIC, WAIC, K-Fold CV) drastically overestimate the number of latent dimensions. For K-fold cross-validation, this overestimation was partly due the cross-validation error curves forming an elbow shape that increased drastically around the true dimension

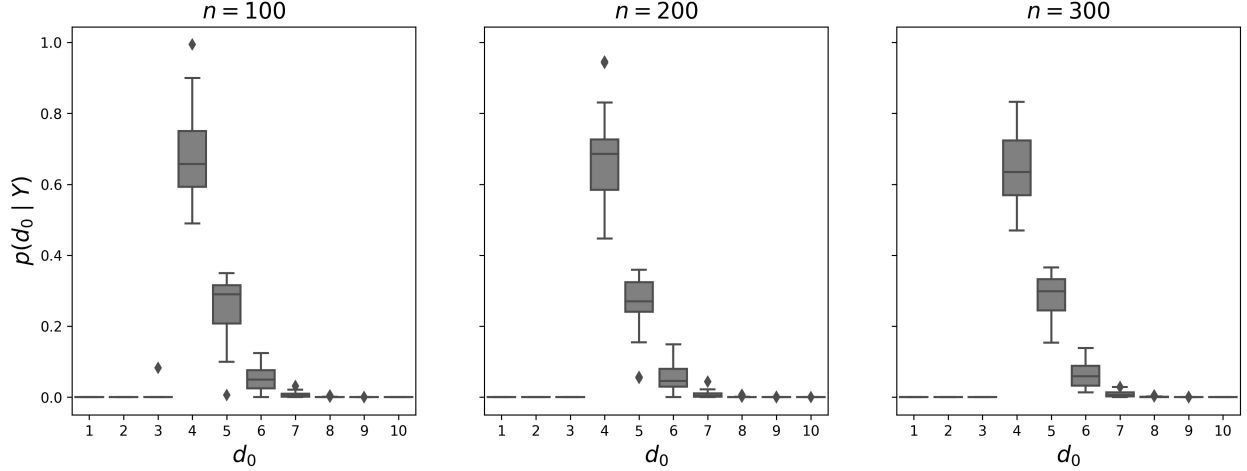


Figure 5.1: Boxplots of $p(d_0 | \mathbf{Y})$ for the static network eigenmodel with an SS-IBP prior over all 50 simulations as n increases from left to right. The true value of $d_0 = 4$.

and slowly decreased afterward. To compensate for this phenomena, we included the one standard error rule used with cross-validation, which selects the smallest model within one standard deviation of the best model. This is indicated by K-Fold (1SE) in Figure 5.1. This ad-hoc method performs better; however, it still underestimates the true latent dimension by selecting $d = 3$ in almost all simulations.

Next, we evaluate the accuracy of the dynamic model. Similar to the static case, Figure 5.3 contains boxplots of the estimated posterior probability of d_0 and Figure 5.4 contains the fraction of time a dimension was selected by the SS-IBP or the competitors over the 50 simulations. Note that we were unable to compute the cross-validation estimator for networks with $n = 200$ because it takes days to compute for a single simulation. Overall, the SS-IBP prior puts the highest posterior mass on the correct dimension $d_0 = 4$ for $n = 50, 100$ and 200 ; however, it occasionally underestimates the dimension for networks with a small number of nodes ($n = 25$). In this case, both the cross-validation estimator and SS-IBP prior select the true dimension 52% of the time. However, the cross-validation estimator is much more computationally expensive and takes a day to compute for even the $n = 100$ networks. Such a study indicates that the SS-IBP prior is performs as well or better than the competing methods considered in this simulation study.

5.5.4 Predictive Performance

This last simulation study compares the dynamic network eigenmodel with existing latent space models for dynamic networks, see Section 5.5.1. We considered three scenarios with a fixed number of nodes $n = 100$ and time steps $T = 6$ and a latent space with $d_0 = 4$ dimensions. The scenarios are distinguished by whether the network is initially fully disassortative (negative homophily) with $\mathbf{\Lambda}_1 = n \text{diag}(-2, -1, -2, -1)$, fully assortative (positive homophily) with $\mathbf{\Lambda}_1 = n \text{diag}(2, 1, 2, 1)$, or a mixture of both with $\mathbf{\Lambda}_1 = n \text{diag}(-2, -1, 2, 1)$. The remaining parameters and their temporal evolution are the same as in the previous section, which did not include dyad-wise covariates or edge (non-edge) persistence parameters in order to compare fairly with the competing methods. For each scenario, we generated 50 networks and measured a model's predictive performance by calculating the area under the receiver operator characteristic (ROC) curve (AUC) for predicting in-sample, held-out, and future dyads. To evaluate the held-out predictions, we removed 20% of the dyads from each network. The same set of dyads was removed from each time-point. To evaluate forecasting

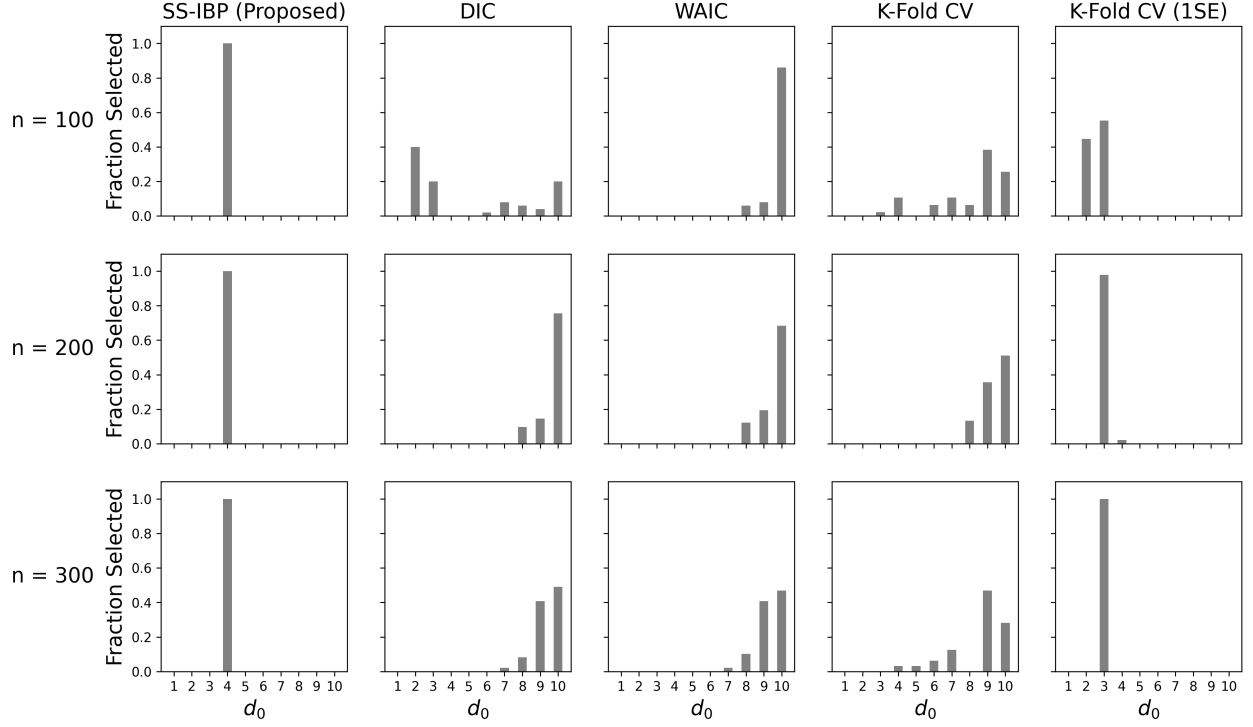


Figure 5.2: Fraction of the 50 simulations in which $\hat{d}_0$ equaled a particular value for the SS-IBP (proposed), DIC, WAIC, K-fold CV, and K-fold CV (1SE) estimators. The true value of $d_0 = 4$.

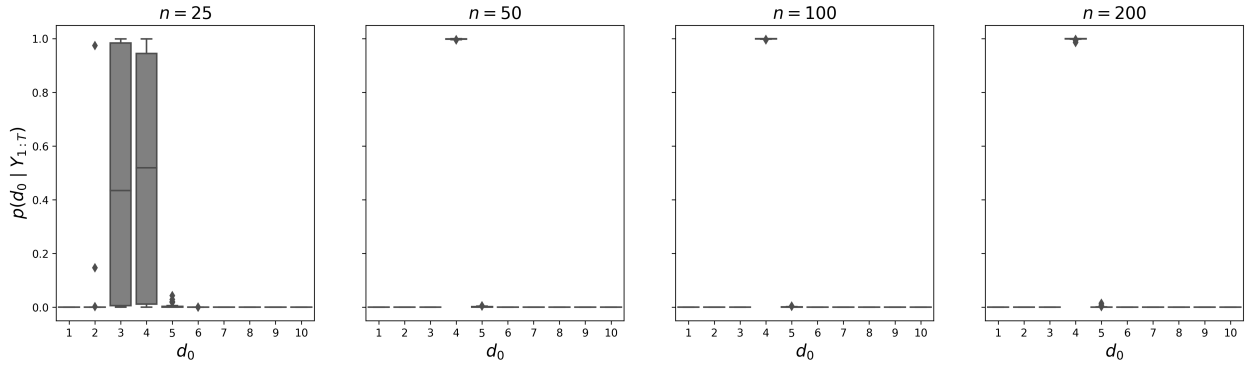


Figure 5.3: Boxplots of $p(d_0 | \mathbf{Y}_{1:T})$ for the dynamic network eigenmodel with an SS-IBP prior over all 50 simulations as n increases from left to right. The true value of $d_0 = 4$.

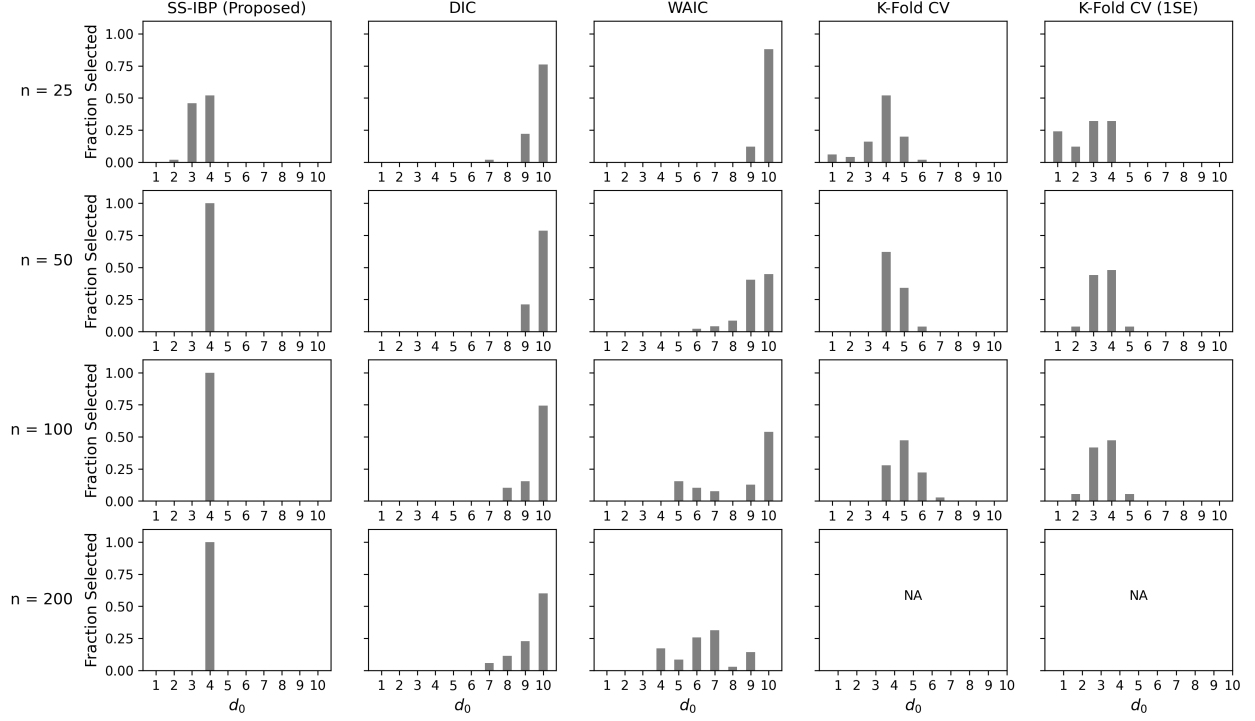


Figure 5.4: Fraction of the 50 simulations in which $\hat{d}_0$ equaled a particular value for the SS-IBP (proposed), DIC, WAIC, K-fold CV, and K-fold CV (1SE) estimators as n increases from left to right. The true value of $d_0 = 4$.

performance, we removed the network at $T = 6$ from estimation and predicted its dyads based on each model's forecast. As a baseline, we also included forecasts from a naive model that uses the values of $\mathbf{Y}_t$ to forecast $\mathbf{Y}_{t+1}$.

Table 5.3 displays the results of this experiment. These results highlight the advantage of the dynamic network eigenmodel for networks with disassortative dimensions. In particular, for networks with purely disassortative dimensions, the competitors' held-out AUC almost equals 0.5, which indicates performance equivalent to a random classifier. In addition, when all dimensions are assortative, the competitors perform similarly to the eigenmodel. This is expected since the models are equivalent in this case. Overall, we conclude that the dynamic network eigenmodel performs better than or equivalently to existing methods, with the gap in performance increasing as the number of disassortative dimensions increases.

5.6 Real Data Applications

To illustrate the benefits of our dimension selection procedure, we use the proposed static and dynamic network eigenmodels to analyze real-world networks. We consider networks from biology and economics. The first example studies a static network of interactions between 270 proteins of *E. coli*. The second example analyzes arms-trade between 88 nations over the four decades of the Cold War.

Network Type	Model	In-Sample AUC	Held-Out AUC	Forecast AUC
Disassortative	Network Eigenmodel (Proposed)	0.925 (0.003)	0.892 (0.010)	0.804 (0.010)
	Bilinear Model	0.700 (0.0036)	0.561 (0.024)	0.620 (0.01)
	Squared Distance Model	0.717 (0.007)	0.571 (0.026)	0.623 (0.011)
	Naive Forecast	–	–	0.681 (0.010)
Assortative	Network Eigenmodel (Proposed)	0.925 (0.003)	0.891 (0.01)	0.805 (0.009)
	Bilinear Model	0.920 (0.006)	0.880 (0.01)	0.827 (0.01)
	Squared Distance Model	0.897 (0.008)	0.852 (0.012)	0.801 (0.01)
	Naive Forecast	–	–	0.682 (0.015)
Mixed	Network Eigenmodel (Proposed)	0.926 (0.003)	0.891 (0.011)	0.807 (0.009)
	Bilinear Model	0.825 (0.007)	0.765 (0.02)	0.748 (0.011)
	Squared Distance Model	0.820 (0.006)	0.748 (0.019)	0.735 (0.012)
	Naive Forecast	–	–	0.685 (0.01)

Table 5.3: Predictive performance on simulated networks ($n = 100$, $T = 6$). Each cell contains the mean value of the metric with the standard deviation over the 50 simulations displayed in parentheses. The best performing model is shaded in gray. The naive forecast corresponds to using $\mathbf{Y}_t$ to forecast $\mathbf{Y}_{t+1}$.

5.6.1 A Static Protein Interaction Network

Here, we re-analyze a static network of protein-protein interaction data first introduced as a benchmark for the static network eigenmodel in P. D. Hoff (2009). Each node corresponds to an essential protein of *E. coli* with $n = 270$ proteins overall. An undirected binary edge ($y_{ij} = 1$) is recorded if protein i and protein j interact, otherwise $y_{ij} = 0$. For a detailed biological interpretation of the network, see the original reference (Butland et al., 2005).

In this analysis, we demonstrate that the proposed SS-IBP prior selects a better fitting model than the K-fold cross-validation estimator used in previous analyses (P. D. Hoff, 2008, 2009; Jauch et al., 2021). Specifically, we estimated a static network eigenmodel using both our proposed SS-IBP prior and the $N(0, nI_d)$ prior assumed by the original analysis. We estimated both models by running the MCMC algorithm described in Section 5.4 for 20,000 iterations after a burn-in of 5,000 iterations. The normal model using K-fold cross-validation with $K = 4$ estimated $\hat{d}_0 = 3$, which matches the original analysis. See Figure 5.12 in Section 5.8.5 of the supplementary material for details. For the SS-IBP prior we set $d = 10$ and made the same hyperparameter choices as described in Section 5.5.1. Figure 5.5 displays various summaries of the posterior of d_0 . In particular, the posterior mode selects $\hat{d}_0 = 5$ dimensions.

A natural approach to comparing the goodness-of-fit of two network models is based on their predictions of observed quantities (Hunter et al., 2008). As such, we demonstrate that the five dimensional model better describes the data by comparing the two models’ posterior predictive distributions of the transitivity coefficient and the degree distribution. We chose to use the transitivity coefficient because latent space models are often used to explain the high levels of transitivity found in real-world networks. We included the degree distribution because the network science literature puts tremendous focus on this statistic. Figure 5.6 displays the results. The proposed model better fits the data based on both statistics. While the model with $d_0 = 3$ severely underestimates the transitivity, the observed transitivity falls well within the bulk of the SS-IBP model’s posterior predictive distribution. Similarly, the degree distribution more closely matches the SS-IBP model, although both models underestimate the number of nodes with a degree of one and overestimate the number of nodes with a degree of zero.

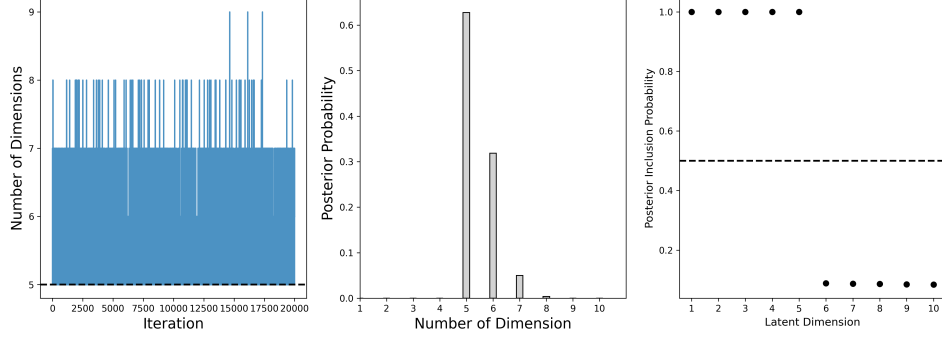


Figure 5.5: Summaries of the number of latent dimensions inferred by the SS-IBP prior. (Left) The trace plot of the number of dimensions visited by the Markov chain, (Center) The posterior probabilities over the number of dimensions, (Right) The posterior inclusion probabilities of a given dimension.

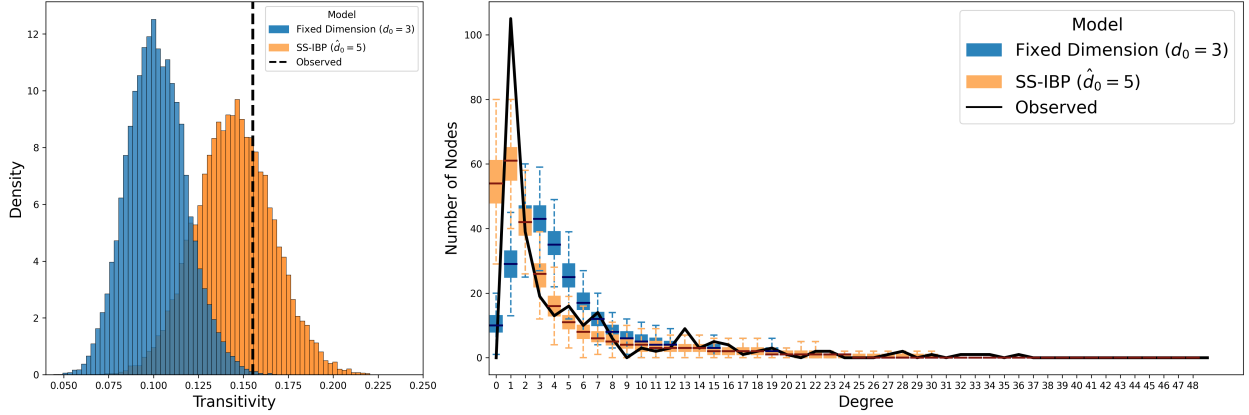


Figure 5.6: Plots of goodness-of-fit statistics comparing a static network eigenmodel with a fixed dimension of $d_0 = 3$ selected through K-fold cross-validation and the proposed SS-IBP prior ($\hat{d}_0 = 5$).

Lastly, we visualize the five dimensional latent space. First, the posterior mean for the five non-zero homophily coefficients has three positive values and two negative values with $\hat{\mathbf{\Lambda}} = \text{diag}(362, 232, 231, -276, -184)$. The latent positions of the three dimensions associated with the positive homophily coefficients are plotted in the first two cells of Figure 5.7. For dimensions with positive homophily coefficients, $\mathbf{u}_i$ and $\mathbf{u}_j$ being in the same direction increases the odds of forming a link between protein i and protein j . Furthermore, “hub” proteins, i.e., proteins with many connections, lie on the periphery of the latent space because the probability of a connection increases with the magnitude of $\mathbf{u}_i$ and $\mathbf{u}_j$. These “hub” proteins are labeled with their name. The last two cells of Figure 5.7 displays the two dimensions with negative homophily coefficients on the y-axis. These dimensions identify two groups defined by proteins with very positive or negative values of u_{i4} or u_{i5} , respectively. In Figure 5.7, these groups are separated by the x-axis (the horizontal dotted-black line). For negative homophily coefficients, proteins with latent positions with opposite signs tend to interact. As such, these dimensions reveal a bipartite substructure in the network defined by proteins in these groups tending to interact across groups (across the x-axis) but not within groups (on the same side of the x-axis).

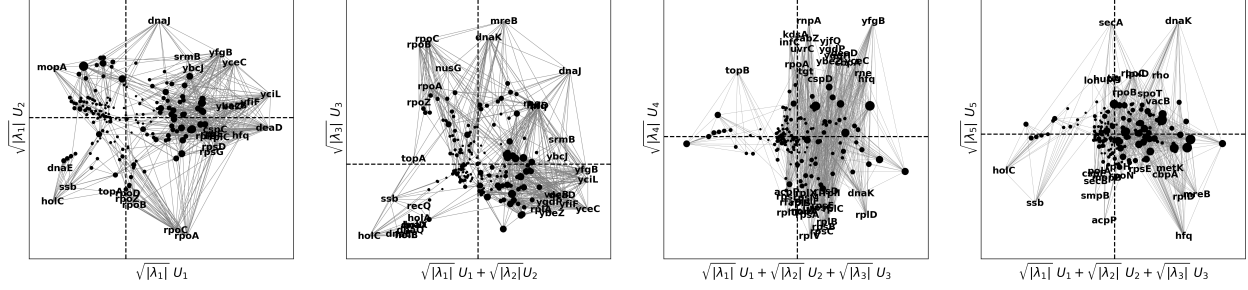


Figure 5.7: Plots of the latent positions estimated for the protein interaction network. A node's size is proportional to its degree in the observed network. The horizontal and vertical dotted-black lines denote the x-axis and y-axis, respectively.

5.6.2 A Dynamic Global Arms-Trade Network

In this section, we use the dynamic network eigensmodel to analyze a network of major conventional weapon transfers between nations during the Cold War from 1950 to 1991. Our goal is to identify factors that influence international weapons transfers, which can help guide political, security, and economic policy. The networks were constructed from data taken from the Stockholm International Peace Research Institute (SIPRI) Arms Transfers Database (<https://www.sipri.org/databases/armstransfers>). For more information on the database and the data curation methodology, see Holtom et al. (2012).

Based on the information in the SIPRI Arms Transfers Database, we constructed a dataset of yearly bilateral arms transfers between pairs of countries. From this dataset, we formed an undirected dynamic network by recording an undirected binary edge ($y_{ijt} = 1$) between two nations if nation i and nation j transferred arms during a given decade, otherwise we recorded $y_{ijt} = 0$. For simplicity, we only considered nations that traded arms during each of the four decades under study. This preprocessing resulted in a dynamic network with $T = 4$ time points and $n = 88$ nodes, where $1 \leq t \leq 4$ enumerates the following time periods: 1950 - 1959, 1960 - 1969, 1970 - 1979, and 1980 - 1991.

As a competitor to the dynamic network eigensmodel, we consider the gravity model from economic theory for modeling international trade (Isard, 1954). The standard gravity model states that the trade volume between two nations is proportional to their market sizes and inversely proportional to their spatial distance. As we will show shortly, this model is a special case of the dynamic network eigensmodel with covariates. As such, we include the logarithm of the gross domestic product (GDP, in millions of US dollars) as a measure of market size. The GDP data was taken from Gleditsch (2002). In addition, we included the spatial distance between nations i and j , which was taken from Gleditsch (2013). Lastly, we included a measure of the difference in political regimes between two trading partners through the Polity score, which is an index that ranges from -10 (hereditary monarchy) to 10 (consolidated democracy). The yearly Polity scores were taken from the Center for Systematic Peace (2022). We considered both the absolute difference in Polity score and whether the Polity score shared the same sign as a measure of alignment of two nations' political regimes. With these covariates, the dynamic network eigensmodel states that the log-odds of forming an edge at time t is

$$\begin{aligned} \text{logit}[\mathbb{P}(Y_{ijt} = 1 | Y_{ij(t-1)} = y_{ij(t-1)})] = & \beta_0 + \beta_1(\log \text{GDP}_{it} + \log \text{GDP}_{jt}) + \beta_2 \log \text{Distance}_{ij} \\ & + \beta_3 |\text{Polity}_{it} - \text{Polity}_{jt}| + \beta_4 \mathbb{1}\{\text{Polity}_{it} \times \text{Polity}_{jt} > 0\} \\ & + \gamma_t y_{ij(t-1)} + \alpha_t (1 - y_{ij(t-1)}) + \mathbf{u}_{it}^T \boldsymbol{\Lambda}_t \mathbf{u}_{jt}. \end{aligned}$$

As previously asserted, the standard gravity model is a special case of the above equation without the $\mathbf{u}_{it}^T \mathbf{\Lambda}_t \mathbf{u}_{jt}$ term. This generalization highlights the SS-IBP prior’s utility in this context, which is to formally select the dynamic network eigenmodel over the gravity model by estimating $\hat{d}_0 > 0$.

We estimated both the gravity model and the dynamic network eigenmodel by running the MCMC algorithm described in Section 5.4 for 20,000 iterations after a burn-in of 5,000 iterations. The dynamic network eigenmodel set the maximum number of dimensions $d = 10$ and used the same hyperparameters as described in Section 5.5.1. Figure 5.8 displays various summaries of the dynamic network eigenmodel’s posterior for d_0 . In particular, the posterior mode selects a model with $\hat{d}_0 = 3$, which indicates a strong preference for a model that includes latent positions. We further confirm this conclusion by comparing multiple goodness-of-fit statistics calculated based on the gravity model and the dynamic network eigenmodel in Figure 5.9. In all cases, the goodness-of-fit statistics prefer the dynamic network eigenmodel.

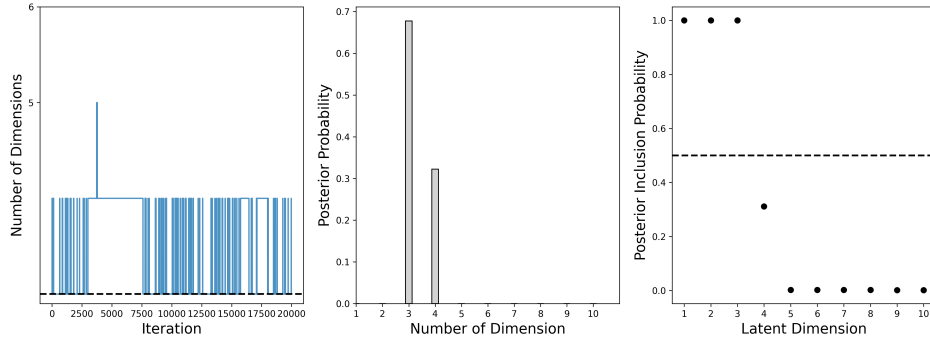


Figure 5.8: Posterior summaries for the number of latent dimensions.

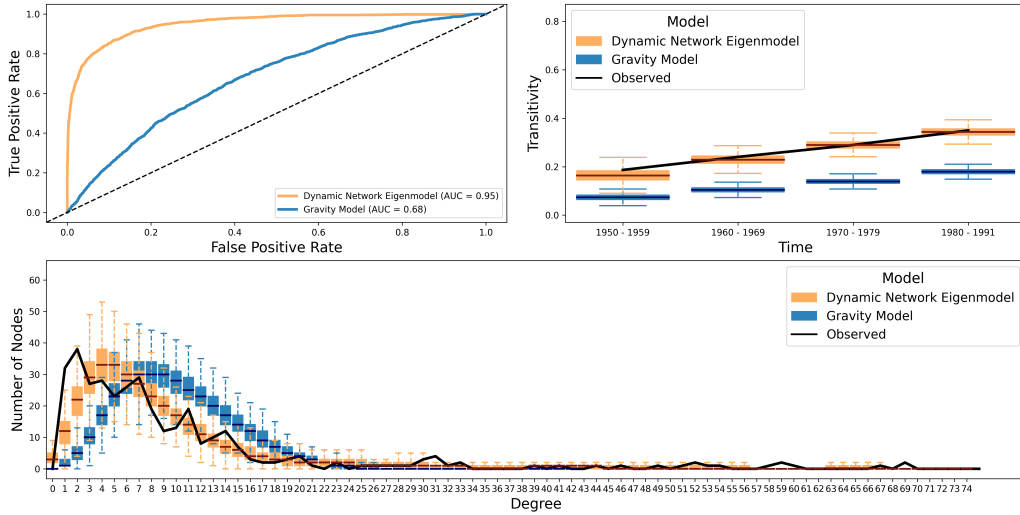


Figure 5.9: Goodness-of-fit statistics comparing the standard gravity model and the dynamic network eigenmodel on the dynamic global arms-trade networks. (Top Left) ROC curves comparing two models on predicting in-sample dyads. The area under the ROC curve (AUC) for each model is displayed in the legend. (Top Right) Posterior predictive distribution of the transitivity coefficient. (Bottom) Posterior predictive degree distribution aggregated over all time points.

Next, we interpret the covariate effects under the dynamic network eigenmodel. Figure 5.10(a) shows the marginal posterior distributions of the four regression coefficients along with 95% credible intervals. As

predicted by economic theory, the log-odds of trading arms has a positive association with GDP and a strong negative association with distance. Furthermore, there is no indication that the Polity score, a proxy for regime dissimilarity, has an effect on arms trade. Lastly, Figure 5.10(b) shows the posterior means and 95% credible intervals of the edge (non-edge) persistence parameters plotted over time. The results indicate a strong positive edge persistence effect, which corroborates the intuitive hypothesis that established trade partnerships tend to persist.

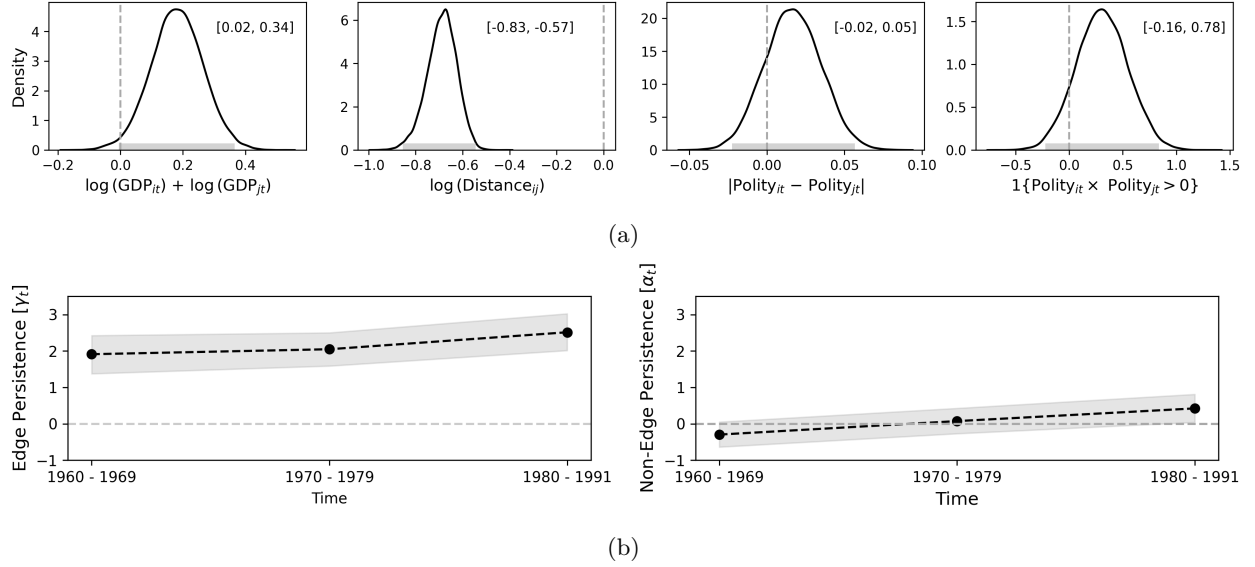


Figure 5.10: (a) Posterior densities for β 's elements. The horizontal gray lines are the 95% credible intervals whose numerical values are shown in the top-right of each plot. (b) Inferred edge and non-edge persistence parameters. The dashed black lines are the posterior means and the shaded gray regions indicate 95% credible intervals.

Finally, we examine the inferred latent space with the goal of determining other factors that contribute to forming arms-trade partnerships. Figure 5.11(a) displays the posterior estimates of the three non-zero homophily coefficients. The results indicate two positive dimensions and one negative dimension over all time points. The first row of Figure 5.11(b) plots the estimated latent positions for the two dimensions with positive homophily coefficients. In this case, $u_{it,1}$ separates members of NATO (USA, UKG, FRN, CAN) and the Warsaw Pact (USR, CZE, POL, RUM) across the y-axis. Furthermore, $u_{it,2}$ separates the two super powers, USA and USR, from the remaining members of each bloc across the x-axis. Lastly, $u_{it,3}$, which has a negative homophily coefficient, reveals a bipartite structure in global arms trade. This structure separates major arms producers in the Western and Eastern blocs from smaller buyers in Third World countries, e.g., countries not formally aligned with either NATO or the Warsaw Pact such as Egypt and India, across the x-axis.

5.7 Discussion

In this chapter, we proposed a new network eigenmodel for dynamic networks that reduces to the original static network eigenmodel in the case of a single static network (P. D. Hoff, 2008). This model generalizes existing LSMs for dynamic networks and models time-varying homophily patterns that previous models could not capture.

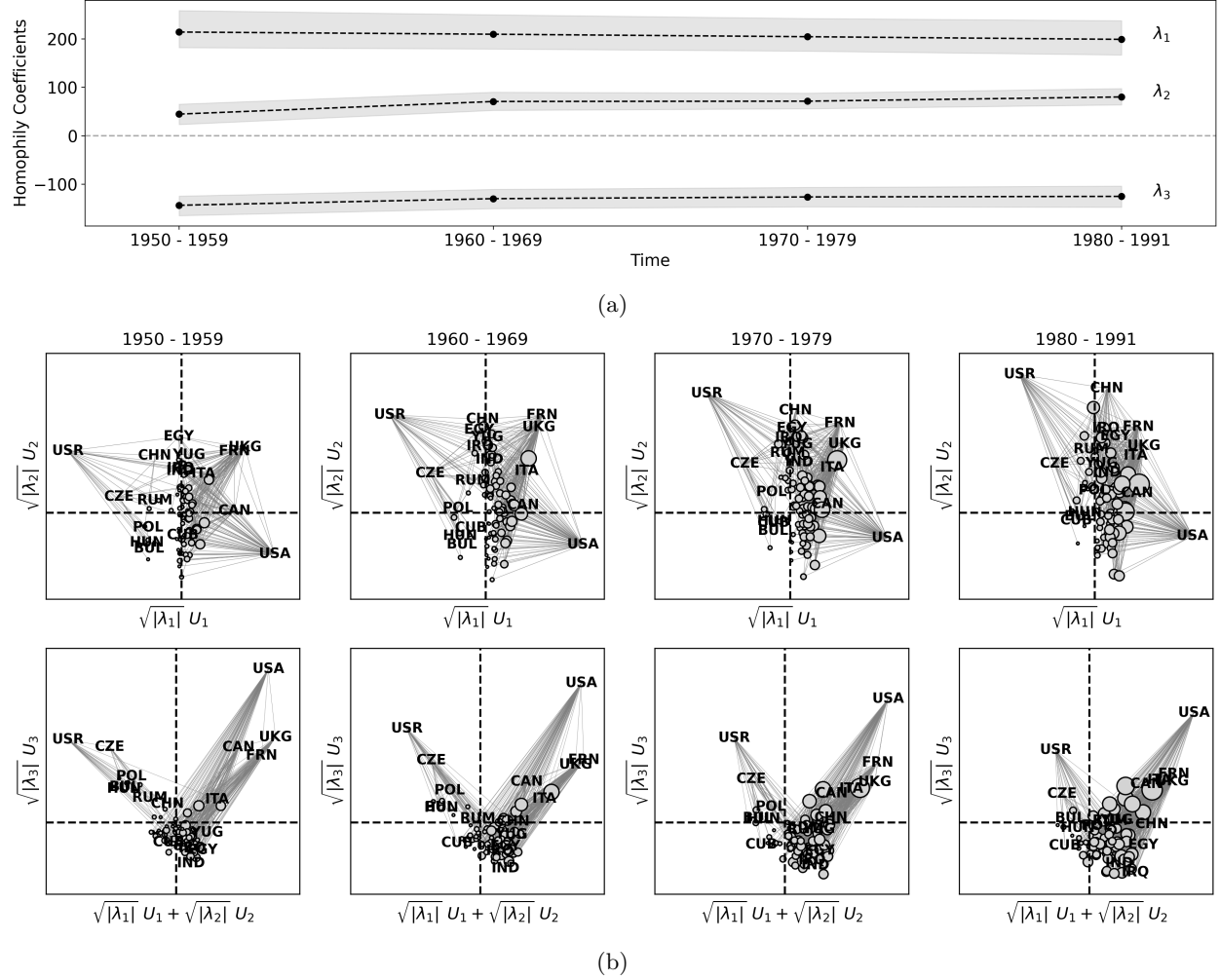


Figure 5.11: (a) Inferred time-varying homophily coefficients $\Lambda_{1:T}$ for the first three dimensions selected by the SS-IBP prior. The dashed black lines are the posterior means and the shaded gray regions are 95% credible intervals. (b) Posterior estimates of the latent factors $\mathbf{U}_{1:T}$ for the first three dimensions selected by the SS-IBP prior. A node's size is proportional to its degree in the observed network. The horizontal and vertical dotted-black lines denote the x-axis and y-axis, respectively.

In addition, we introduced a novel SS-IBP_d($\alpha, \kappa, \mathbb{P}_{spike}, \mathbb{P}_{slab}$) prior that addresses the important problem of automatically selecting the latent space dimension d_0 . We showed that this prior induces a stochastic ordering that counteracts certain identifiability issues in the model's likelihood. In addition, we provided a posterior tail bound on the number of dimensions that reflects the true model's underlying dimensionality. To the best of our knowledge, this is the first such theoretical result for Bayesian LSMs. Furthermore, this result is an essential first step toward establishing a posterior concentration rate for network eigenmodel estimation.

Our approach, theoretical results, and estimation method can be directly extended to networks with non-Bernoulli edges drawn from an exponential family distribution or models that use a link function other than the logit, such as the probit. An interesting research direction is to extend our approach to directed networks. The typical network eigenmodel for directed networks involves a decomposition based on the singular value decomposition: the latent space component, $\mathbf{U}\mathbf{A}\mathbf{U}^T$, becomes $\mathbf{U}\mathbf{S}\mathbf{V}$, where $\mathbf{U} \in \mathcal{V}_{d,n}$, $\mathbf{V} \in \mathcal{V}_{d,n}$, and the matrix of singular values $\mathbf{S} = \text{diag}(s_1, \dots, s_d)$ has positive real-valued entries. The positivity of the singular values invalidates our use of a Laplace(b) slab, so choosing an appropriate prior requires additional computational and theoretical care. Finally, while our approach to dimension selection is more computationally efficient than existing model selection criteria for Bayesian LSMs, our use of MCMC sampling is still computationally expensive for large networks. The exploration of an appropriate EM algorithm (Dempster et al., 1977) to estimate the posterior mode or a variational inference algorithm (Wainwright & Jordan, 2008) to approximate the posterior is an area of future research interest.

5.8 Supplementary Material

5.8.1 Full Conditional Distributions

In this section, we present the logarithm of the full conditional distribution used in the Hamiltonian Monte Carlo algorithm presented in Section 5.4.4 of the main text. First, we define some more notation. Denote that change-of-variable transformations described in Section 5.4.3 of the main text by

$$\begin{aligned}\eta_{\sigma_h} &= \log \sigma_h, & h &= 1, \dots, d, \\ \eta_\phi &= \log \phi, & \eta_\gamma &= \log \sigma_\gamma, & \eta_\alpha &= \log \sigma_\alpha, \\ \eta_\rho &= \log \left(\frac{\tilde{\rho}}{1 - \tilde{\rho}} \right), & \text{where } \tilde{\rho} &= (\rho + 1)/2, \\ \eta_{\nu_h} &= \log \left(\frac{\nu_h}{1 - \nu_h} \right), & h &= 1, \dots, d.\end{aligned}$$

Also, let $\sigma(x) = 1/(1 + e^{-x})$ denote the inverse logistic transform. With these definitions, the logarithm of $p(\boldsymbol{\psi} \mid Z_{1:d}, \mathbf{Y}_{1:T})$ up to an additive constant is

$$\begin{aligned}\log p(\boldsymbol{\psi} \mid Z_{1:d}, \mathbf{Y}_{1:T}) &= \sum_{t=1}^T \sum_{i < j} y_{ijt} [\boldsymbol{\Theta}_t(Z_{1:d})]_{ij} + \log \left\{ 1 - \sigma \left([\boldsymbol{\Theta}_t(Z_{1:d})]_{ij} \right) \right\} \\ &\quad - \frac{\boldsymbol{\beta}^T \boldsymbol{\beta}}{2\sigma_\beta^2} - \frac{1}{2} \sum_{t=1}^T \text{tr}(\mathbf{B}_t^T \mathbf{B}_t)\end{aligned}$$

$$\begin{aligned}
& - \sum_{h=1}^d \sum_{t=1}^T \frac{\tilde{\lambda}_{th}^2}{2} - \frac{\gamma_2^2}{2\sigma_{0,\gamma}^2} - \sum_{t=3}^T \frac{\tilde{\gamma}_t^2}{2} - \frac{\alpha_2^2}{2\sigma_{0,\alpha}^2} - \sum_{t=3}^T \frac{\tilde{\alpha}_t^2}{2} \\
& - \sum_{h=1}^d \frac{e^{2\eta_{\sigma_h}}}{2b^2} - \frac{e^{\eta_\phi}}{b_\phi} + \log[2\sigma(\eta_\rho) - 1] - \log(1 + e^{2\eta_\gamma}) - \log(1 + e^{2\eta_\alpha}) \\
& + \kappa \log[1 - \sigma(\eta_{\nu_1})] + (\alpha - 1) \sum_{h=1}^d \log[\sigma(\eta_{\nu_h})] \\
& + \sum_{h=1}^d [Z_h \log(\theta_h) + (1 - Z_h) \log(1 - \theta_h)] \\
& - (T - 1) \log {}_0\tilde{F}_1 \left(\frac{n}{2}, \frac{e^{2\eta_\phi}}{4} I_d \right) \\
& + \text{tr} (e^{\eta_\phi} \mathbf{Q}_{\mathbf{B}_1}^T \mathbf{U}_{\mathbf{B}_2}) + \text{tr} \left(e^{\eta_\phi} \sum_{t=3}^T \mathbf{U}_{\mathbf{B}_{t-1}}^T \mathbf{U}_{\mathbf{B}_t} \right) \\
& + \log |\mathbf{J}(\eta_{\sigma_{1:d}}, \eta_\phi, \eta_\gamma, \eta_\alpha, \eta_\rho, \eta_{\nu_{1:d}})| + C.
\end{aligned} \tag{5.17}$$

For $t = 1$:

$$[\boldsymbol{\Theta}_1(Z_{1:d})]_{ij} = \boldsymbol{\beta}^T \mathbf{X}_{ij1} + [\mathbf{Q}_{\mathbf{B}_1} \boldsymbol{\Lambda}_1(Z_{1:d}) \mathbf{Q}_{\mathbf{B}_1}^T]_{ij}.$$

For $2 \leq t \leq T$:

$$[\boldsymbol{\Theta}_t(Z_{1:d})]_{ij} = \boldsymbol{\beta}^T \mathbf{X}_{ijt} + \gamma_t y_{ij(t-1)} + \alpha_t (1 - y_{ij(t-1)}) + [\mathbf{U}_{\mathbf{B}_t} \boldsymbol{\Lambda}_t(Z_{1:d}) \mathbf{U}_{\mathbf{B}_t}^T]_{ij}.$$

The term involving the logarithm of the Jacobian determinant is

$$\begin{aligned}
\log |\mathbf{J}(\eta_{\sigma_{1:d}}, \eta_\phi, \eta_\gamma, \eta_\alpha, \eta_\rho, \eta_{\nu_{1:d}})| &= \eta_\phi + \eta_\gamma + \eta_\alpha + \sum_{h=1}^d \eta_{\sigma_h} \\
&+ \log[\sigma(\eta_\rho)(1 - \sigma(\eta_\rho))] + \sum_{h=1}^d \log[\sigma(\eta_{\nu_h})(1 - \sigma(\eta_{\nu_h}))].
\end{aligned}$$

5.8.2 Parameter Initialization

We can greatly reduce the number of iterations required to reach convergence by choosing good initial values for the model parameters. We initialize the parameters with the last sample from a short MCMC chain (500 iterations) targeting a model without dimension selection, i.e., we repeat step one of Algorithm 5.1 for 500 iterations with $Z_h = 1$ for $1 \leq h \leq d$. We remove the dimension selection indicators to allow the sampler to initially freely explore the unconstrained parameter space before applying shrinkage. The initial parameters of this short chain are drawn randomly according to NumPyro's recommended random initialization scheme, which draws the unconstrained variables uniformly from the interval $(-2, 2)$ (Phan et al., 2019). However, we drew the initial values of $\mathbf{B}_{1:T}$ randomly as follows: We construct $\mathbf{B}_1$ with iid entries drawn from a $N(0, 1)$ distribution, and we evolve the entries as $(\mathbf{B}_t)_{ij} = (\mathbf{B}_{(t-1)})_{ij} + N(0, 0.1^2)$ for $2 \leq t \leq T$ and $1 \leq i \leq j \leq d$. We do this to ensure the initial values for $\mathbf{Q}_{\mathbf{B}_1}$ and $\mathbf{U}_{\mathbf{B}_{2:T}}$ have a smooth trajectory.

5.8.3 Definitions of Competing Model Selection Criteria

First, we define the estimators of d_0 based on minimizing information criteria estimated with the posterior samples. Let $\{\boldsymbol{\xi}^{(s)}\}_{s=1}^S$ denote the S posterior samples from the Bayesian model under consideration. In addition, let $\hat{\boldsymbol{\xi}}$ denote an estimate of the parameters based on these posterior samples. The DIC is defined as

$$\text{DIC} = -2 \log p(\mathbf{Y}_{1:T} \mid \hat{\boldsymbol{\xi}}) + 2 \times p_{DIC},$$

where $p_{DIC} = 2 \left(\log p(\mathbf{Y}_{1:T} \mid \hat{\boldsymbol{\xi}}) - \frac{1}{S} \sum_{s=1}^S \log p(\mathbf{Y}_{1:T} \mid \boldsymbol{\xi}^{(s)}) \right)$. Smaller values of DIC are better. The first term measures model fit, while the second term penalizes model complexity.

Next, WAIC approximates the leave-one-out cross-validation (LOO-CV) procedure with the expected log-predictive density as the loss function. For the dynamic network eigenmodel, we use the following WAIC estimator

$$\text{WAIC} = -2 \sum_{t=1}^T \log \left[\frac{1}{S} \sum_{s=1}^S p(\mathbf{Y}_t \mid \boldsymbol{\xi}^{(s)}) \right] + 2 \times \sum_{t=1}^T V_{s=1}^S \log p(\mathbf{Y}_t \mid \boldsymbol{\xi}^{(s)}),$$

where $V_{s=1}^S \log p(\mathbf{Y}_t \mid \boldsymbol{\xi}^{(s)})$ denotes the sample variance of the log-likelihood at time t evaluated over the posterior samples. Smaller values of WAIC are better. The first term measures the model fit, while the second term penalizes model complexity. This estimator approximates a leave one time-point out cross-validation scheme. The WAIC estimator assumes the time-points are independent conditioned on the model parameters, which is only valid when no edge (non-edge) persistence parameters are included in the dynamic network eigenmodel. For this reason, the simulation studies did not include the persistence parameters in Section 5.5.3. For the static network eigenmodel, we used the following WAIC estimator:

$$\text{WAIC} = -2 \sum_{i < j} \log \left[\frac{1}{S} \sum_{s=1}^S p(Y_{ij} \mid \boldsymbol{\xi}^{(s)}) \right] + 2 \times \sum_{i < j} V_{s=1}^S \log p(Y_{ij} \mid \boldsymbol{\xi}^{(s)}),$$

which approximates a leave one dyad out cross-validation scheme.

Lastly, we considered a dynamic extension to the K-fold cross-validation estimator originally proposed by P. D. Hoff (2008) for the static case. We outline the proposed K-fold cross-validation estimator in Algorithm 5.2.

1. Randomly split the set of dyads $\{(i, j) : i < j\}$ into K test sets $A_1, \dots, A_K$.
2. For $d = 1, \dots, D$:
 - (a) For $k = 1, \dots, K$:
 - (1) Perform the MCMC algorithm using only $\{y_{ijt} : (i, j) \notin A_k\}$ and estimate the log-odds matrices $\{\hat{\boldsymbol{\Theta}}_t\}_{t=1}^T$ for all dyads by their posterior means.
 - (2) Based on these estimates, compute the log predictive probability over all held-out dyads $L_d^{(k)} = \sum_{t=1}^T \sum_{(i,j) \in A_k} \log \mathbb{P}(Y_{ijt} = y_{ijt} \mid \hat{\boldsymbol{\Theta}}_t)$.
 - (b) Measure the predictive performance of d as $L_d = \sum_{k=1}^K L_d^{(k)} / K$.
3. Select $\hat{d} = \arg \max_{1 \leq d \leq D} L_d$.

Algorithm 5.2: K-Fold Cross-Validation for the Dynamic Network Eigenmodel

5.8.4 Proofs of Main Results

Proofs of Propositions 5.1, 5.2, 5.3, and 5.4, and Corollary 5.1

Proof of Proposition 5.1. First, we assert that the individual log-odds at $t = 1$, $\theta_{ij1} = \beta^T \mathbf{X}_{ij1} + \mathbf{u}_{i1}^T \Lambda_1 \mathbf{u}_{j1}$, are identifiable for all $1 \leq i \leq j \leq n$. To see this, note that the condition that two probability distributions under two different parameterizations coincide implies that their marginal distributions must also coincide. In this case, we can consider the initial distribution of $\mathbf{Y}_1$ parameterized by two sets of parameters $\{\theta_{ij1}\}_{i \leq j}$ and $\{\tilde{\theta}_{ij1}\}_{i \leq j}$. Furthermore, the marginal distributions being equal for an individual dyad Y_{ij1} is equivalent to

$$Y_{ij1}(\theta_{ij1} - \tilde{\theta}_{ij1}) - \log \left[\frac{1 + e^{\theta_{ij1}}}{1 + e^{\tilde{\theta}_{ij1}}} \right] = 0,$$

which implies $\theta_{ij1} = \tilde{\theta}_{ij1}$ for $Y_{ij1} = 0$ or $Y_{ij1} = 1$ and $1 \leq i \leq j \leq n$.

Now, let $\mathbf{W}_1$ and $\tilde{\mathbf{W}}_1$ denote matrices with entries $\beta^T \mathbf{X}_{ij1}$ and $\tilde{\beta}^T \mathbf{X}_{ij1}$, respectively. The previous discussion showed that the log-odds values at $t = 1$ are equal for all $1 \leq i \leq j \leq n$, which in matrix notation becomes

$$\mathbf{W}_1 + \mathbf{U}_1 \Lambda_1 \mathbf{U}_1^T = \tilde{\mathbf{W}}_1 + \tilde{\mathbf{U}}_1 \tilde{\Lambda}_1 \tilde{\mathbf{U}}_1^T. \quad (5.18)$$

Now, note that

$$\mathbf{W}_1 \mathbf{1}_n = \begin{pmatrix} \sum_{j=1}^n \beta^T \mathbf{X}_{1j1} \\ \vdots \\ \sum_{j=1}^n \beta^T \mathbf{X}_{nj1} \end{pmatrix} = \begin{pmatrix} \beta^T \sum_{j=1}^n \mathbf{X}_{1j1} \\ \vdots \\ \beta^T \sum_{j=1}^n \mathbf{X}_{nj1} \end{pmatrix} = \begin{pmatrix} \left[\sum_{j=1}^n \mathbf{X}_{1j1} \right]^T \beta \\ \vdots \\ \left[\sum_{j=1}^n \mathbf{X}_{nj1} \right]^T \beta \end{pmatrix} = \bar{\mathbf{X}}_1 \beta.$$

Similarly, $\tilde{\mathbf{W}}_1 \mathbf{1}_n = \bar{\mathbf{X}}_1 \tilde{\beta}$. Multiplying Equation (5.18) on the right by $\mathbf{1}_n$, applying the previous identities, and rearranging, we have

$$\bar{\mathbf{X}}_1 (\beta - \tilde{\beta}) = \mathbf{0}_n.$$

Next, multiplying on the left by $\bar{\mathbf{X}}_1^T$, we have

$$\bar{\mathbf{X}}_1^T \bar{\mathbf{X}}_1 (\beta - \tilde{\beta}) = \mathbf{0}_p.$$

Since $\bar{\mathbf{X}}_1^T \bar{\mathbf{X}}_1$ is full rank, we must have $\beta = \tilde{\beta}$. □

Proof of Proposition 5.2. Before proving the result, we need to formally define the support of a probability measure and prove a general measure theoretic lemma.

Definition 5.1. Let $\mathbb{P}$ be a probability measure on a measurable space $(X, \mathcal{B}(X))$ where X is a topological space and $\mathcal{B}(X)$ is the associated Borel σ -algebra. The support of $\mathbb{P}$, denoted by $\text{supp}(\mathbb{P})$, is the set of all $x \in X$ such that every open neighborhood of x has non-zero probability.

Lemma 5.1. Let $(X_1, \mathcal{B}(X_1))$ and $(X_2, \mathcal{B}(X_2))$ be measurable spaces where X_1 and X_2 are topological spaces and $\mathcal{B}(X_1)$ and $\mathcal{B}(X_2)$ are their associated Borel σ -algebras. Also, let $\mathbb{P}$ be a probability measure on X_1 and $f : X_1 \rightarrow X_2$ be a continuous, surjective function. If $\text{supp}(\mathbb{P}) = X_1$, then $\mathbb{P}_f$, the push-forward measure on X_2 associated with f defined as $\mathbb{P}(f^{-1}(B))$ for all $B \in \mathcal{B}(X_2)$, has $\text{supp}(\mathbb{P}_f) = X_2$.

Proof. First note that the continuity of f implies that f is $\mathcal{B}(X_1)/\mathcal{B}(X_2)$ -measurable, so the statements

made above are well defined. Now pick an arbitrary $x \in X_2$. Note that for any open neighborhood N_x of x , $f^{-1}(N_x) \neq \emptyset$ because f is surjective. In addition, since $f^{-1}(N_x)$ is a non-empty open set in X_1 , it is also an open neighborhood of some point in X_1 . Therefore, $\mathbb{P}_f(N_x) = \mathbb{P}(f^{-1}(N_x)) > 0$ for any N_x by the assumption that $\mathbb{P}$ has full support on X_1 . Since x was arbitrary, this proves that $\mathbb{P}_f$ has full support on X_2 . $\square$

We are now ready to prove Proposition 5.2. First, we show that the function $f : \mathbb{R}^{n \times d} \rightarrow \tilde{\mathcal{V}}_{d,n}$ defined by steps (2) and (3) in Section 5.2.3 is surjective. Pick an arbitrary $\mathbf{A} \in \tilde{\mathcal{V}}_{d,n}$. Let $\mathbf{B}_1 = \mathbf{A}$, then $\tilde{\mathbf{B}}_1 = [\mathbf{1}_n, \mathbf{A}] I_{d+1}$ so that $\mathbf{Q} = [\mathbf{1}_n, \mathbf{A}]$ and $\mathbf{R} = I_{d+1}$. Therefore, $\mathbf{U}_1 = \mathbf{Q}_{2:(d+1)} = \mathbf{B}$, which proves f is surjective. Furthermore, f is differentiable with respect to $\mathbf{B}_1$ and thus continuous. The result follows from Lemma 5.1 with $X_1 = \mathbb{R}^{n \times d}$, $X_2 = \tilde{\mathcal{V}}_{d,n}$, and $\mathbb{P}$ the Gaussian measure on $\mathbb{R}^{n \times d}$ which has full support. $\square$

Proof of Proposition 5.3. The first result is found in Section 2.3.1 of Chikuse (2003a). Next, Chikuse (2003b) showed that one can express

$$\mathbf{U}_t = \mathbf{U}_{t-1} + \phi^{-1/2} \mathbf{U}_{t-1} \mathbf{Z}_1 + \phi^{-1/2} \mathbf{U}_{t-1}^\perp \mathbf{Z}_2 + O_p(\phi^{-1})$$

for large ϕ where $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are independent random matrices and $\mathbf{U}_{t-1}^\perp$ is a matrix chosen such that $[\mathbf{U}_{t-1}, \mathbf{U}_{t-1}^\perp] \in \mathbb{O}(n)$ the n -dimensional orthogonal group. From this expression, we see that $\mathbf{U}_t$ converges to $\mathbf{U}_{t-1}$ in probability as ϕ goes to infinity. The third assertion follows from the definition of $Z(\phi) = {}_0F_1(\frac{1}{2}n, \frac{1}{4}\phi^2 I_d) = \int_{\mathbb{O}(n)} \text{etr}(\phi \mathbf{V}) d\mu(\mathbf{V})$, where $d\mu(\mathbf{V})$ denotes the normalized invariant measure on $\mathbb{O}(n)$, i.e., the uniform probability measure on $\mathbb{O}(n)$. The previous expression equals one when $\phi = 0$. $\square$

Proof of Proposition 5.4. The proof proceeds directly:

$$\begin{aligned} \mathbb{P}(|\eta_h - \eta_0| \leq \epsilon) &= \mathbb{E}[\mathbb{P}(\mathbb{B}_\epsilon(\eta_0) \mid \theta)] \\ &= \mathbb{E}(\theta_h) \mathbb{P}_{slab}(\mathbb{B}_\epsilon(\eta_0)) + \mathbb{E}(1 - \theta_h) \mathbb{P}_{spike}(\mathbb{B}_\epsilon(\eta_0)) \\ &= \left(\frac{\alpha}{\alpha + \kappa + 1} \right) \left(\frac{\alpha}{\alpha + 1} \right)^{h-1} \mathbb{P}_{slab}(\mathbb{B}_\epsilon(\eta_0)) + \left[1 - \left(\frac{\alpha}{\alpha + \kappa + 1} \right) \left(\frac{\alpha}{\alpha + 1} \right)^{h-1} \right] \mathbb{P}_{spike}(\mathbb{B}_\epsilon(\eta_0)) \\ &= \mathbb{P}_{spike}(\mathbb{B}_\epsilon(\eta_0)) + \left(\frac{\alpha}{\alpha + \kappa + 1} \right) \left(\frac{\alpha}{\alpha + 1} \right)^{h-1} [\mathbb{P}_{slab}(\mathbb{B}_\epsilon(\eta_0)) - \mathbb{P}_{spike}(\mathbb{B}_\epsilon(\eta_0))]. \end{aligned}$$

Since $[\alpha/(\alpha + 1)]^{h-1}$ is decreasing in h , the result follows. $\square$

Proof of Corollary 5.1. Before proving Corollary 5.1, we provide a known result about the Laplace distribution arising from a mixture of normal random variables. We include a proof for completeness.

Lemma 5.2. Fix $a > 0$ and $b > 0$. If $X \mid \sigma^2 \sim N(0, a^2 \sigma^2)$ and $\sigma^2 \sim \text{Exponential}(1/2b^2)$, then $X \sim \text{Laplace}(ab)$.

Proof. The proof proceeds by calculating the moment generating function of X :

$$\begin{aligned} \mathbb{E}[e^{tX}] &= \mathbb{E}[\mathbb{E}[e^{tX} \mid \sigma^2]] \\ &= \mathbb{E}[e^{a^2 \sigma^2 t^2 / 2}] \\ &= \frac{1}{2b^2} \int_0^\infty e^{a^2 \sigma^2 t^2 / 2} e^{-\sigma^2 / 2b^2} d\sigma^2 \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2b^2} \int_0^\infty e^{-\sigma^2[1/2b^2 - a^2t^2/2]} d\sigma^2 \\
&= \frac{1}{1 - a^2b^2t^2}.
\end{aligned}$$

The proof is complete by recognizing that the last line is the moment generating function of a $\text{Laplace}(ab)$ random variable. $\square$

Now we proceed to prove Corollary 5.1. Since $(\lambda_{h1}, \dots, \lambda_{ht}) \mid \phi_h, \sigma_h^2, \rho$ follows a stationary AR(1) process, λ_{ht} is marginally distributed as $\lambda_{ht} \mid \phi_h, \sigma_h^2, \rho \sim N(0, \phi_h^2 \sigma_h^2 / (1 - \rho^2))$ for $1 \leq h \leq d$ and $1 \leq t \leq T$. Applying Lemma 5.2 with $a = \phi_h^2 / (1 - \rho^2)$, we have $\lambda_{ht} \mid \phi_h, \rho \sim \text{Laplace}(\phi_h b / \sqrt{1 - \rho^2})$. We can then marginalize over the indicator variables Z_h in Equation (5.11) to conclude that

$$\lambda_{ht} \mid \theta_h, \rho \stackrel{\text{ind.}}{\sim} \theta_h \text{Laplace}\left(\frac{b}{\sqrt{1 - \rho^2}}\right) + (1 - \theta_h) \text{Laplace}\left(\frac{v_0 b}{\sqrt{1 - \rho^2}}\right), \quad 1 \leq h \leq d,$$

where the independence is across h (not t). The previous expression implies that marginally $(\lambda_{1t}, \dots, \lambda_{dt}) \mid \rho \sim \text{SS-IBP}_d(\alpha, \kappa, \mathbb{P}_{\text{spike}}, \mathbb{P}_{\text{slab}})$ with $\mathbb{P}_{\text{slab}} = \text{Laplace}(b / \sqrt{1 - \rho^2})$ and $\mathbb{P}_{\text{spike}} = \text{Laplace}(v_0 b / \sqrt{1 - \rho^2})$. Note that the Laplace distribution puts greater mass around zero when the scale parameter is near zero. Therefore, the condition $v_0 < 1$ ensures that the spike distribution is more concentrated around zero than the slab distribution. As such, we can apply Proposition 5.4 with $\lambda_0 = 0$ to conclude that $\mathbb{P}(|\lambda_{ht}| \leq \epsilon \mid \rho) < \mathbb{P}(|\lambda_{(h+1)t}| \leq \epsilon \mid \rho)$ for all ρ . We obtain the result by taking expectations of both sides of the previous inequality. $\square$

Proof of Theorem 5.1 and Theorem 5.2

We use the following notation in Section 5.8.4 and Section 5.8.4. For two densities p and q , we let $K(p, q) = \mathbb{E}_p \log(p/q)$ and $V(p, q) = \mathbb{E}_p [\log(p/q) - K(p, q)]^2$ denote the Kullback-Leibler (KL) divergence and variation, respectively. For two positive sequences $\{a_n\}$ and $\{b_n\}$, we write $a_n \asymp b_n$ to denote $0 < \liminf_{n \rightarrow \infty} a_n/b_n \leq \limsup_{n \rightarrow \infty} a_n/b_n < \infty$. If $\lim_{n \rightarrow \infty} a_n/b_n = 0$, we write $a_n = o(b_n)$. We use $a_n = O(b_n)$ to denote that for sufficiently large n , there exists a constant $K > 0$ independent of n such that $a_n \leq Kb_n$. For a vector $\mathbf{v} = (v_1, \dots, v_p)^T \in \mathbb{R}^p$, we let $\|v\|_0, \|v\|_2, \|v\|_\infty$ denote the ℓ_0 , ℓ_2 , and ℓ_∞ norm. For a matrix $C \in \mathbb{R}^{m \times m}$, we denote the Frobenius norm as $\|C\|_F = (\sum_{ij} C_{ij}^2)^{1/2}$.

As in Section 5.3.4 of the main text, we assume the entries of the adjacency matrix, Y_{ij} , are independently drawn from a Bernoulli density of the form,

$$q_{ij}(Y_{ij}) = \exp\{Y_{ij}\theta_{ij} - \log(1 + e^{\theta_{ij}})\},$$

where

$$\theta_{ij} = \beta^T \mathbf{X}_{ij} + [\mathbf{U} \mathbf{A} \mathbf{U}^T]_{ij}. \quad (5.19)$$

We assume $\mathbf{X}_{ij} \in \mathbb{R}^p$ is known and p is fixed. We use $\mathbf{W} \in \mathbb{R}^{n \times n}$ to denote the matrix with entries $\beta^T \mathbf{X}_{ij}$. In addition, we assume that $\{Y_{ij}\}_{j < i}$ are generated from the model in Equation (5.8) with true parameters

$$\theta_{0ij} = \beta_0^T \mathbf{X}_{ij} + [\mathbf{U}_0 \mathbf{A}_0 \mathbf{U}_0^T]_{ij}, \quad (5.20)$$

where $\beta_0 \in \mathbb{R}^p$, $\mathbf{A}_0 \in \mathbb{R}^{d_0 \times d_0}$, and $\mathbf{U}_0 \in \mathcal{V}_{d_0, n}$. Let $\mathbf{W}_0 \in \mathbb{R}^{n \times n}$ denote the matrix with entries $\beta_0^T \mathbf{X}_{ij}$. Also,

let $\mathbb{P}_0^{(n)}$ and $\mathbb{E}_0^{(n)}$ denote the probability and expectation under this true data generating process, respectively. Finally, we denote the Bernoulli density under this true data generating process as

$$q_{0ij}(Y_{ij}) = \exp\{Y_{ij}\theta_{0ij} - \log(1 + e^{\theta_{0ij}})\}.$$

In addition, we make the same assumptions as in Section 5.3.4, which we include here for convenience:

$$(A5.1) \quad d = \lceil n^\gamma \rceil \text{ for some } \gamma \in (0, 1],$$

$$(A5.2) \quad d_0 = o(\log d),$$

$$(A5.3) \quad b = O(d),$$

$$(A5.4) \quad \|\boldsymbol{\lambda}_0\|_\infty \leq M \text{ for some } M > 0,$$

$$(A5.5) \quad \max_{1 \leq k \leq p} \|\mathbf{X}_k\|_F \asymp n \text{ where } \mathbf{X}_k \in \mathbb{R}^{n \times n} \text{ has elements } X_{ij,k}.$$

Our proofs of Theorem 5.1 and Theorem 5.2 makes use of the theory developed by Goshal and van der Vaart (2007) and Jeong and Ghoshal (2021) to demonstrate that the posterior concentrates around low-dimensional models for the static network eigenmodel. Furthermore, we synthesize recent theoretical results for sparse Gaussian factor models (Ohn & Kim, *in press*; Ročková & George, 2016) to derive our result on the $\text{SS-IBP}_d(\alpha, \kappa, \mathbb{P}_{\text{spike}}, \mathbb{P}_{\text{slab}})$ prior's ability to concentrate on low-dimensional models and derive our posterior concentration rates.

Proof of Theorem 5.1. Since the inequality follows trivially when $t \geq d$, we assume $t < d$. Define the random variable $S_d = \|\boldsymbol{\Lambda}\|_0 = \sum_{h=1}^d \mathbb{1}_{\{\lambda_h \neq 0\}}$. We bound $\mathbb{P}(S_d > t)$ as follows:

$$\mathbb{P}(S_d > t) \leq \mathbb{E} \left[\mathbb{P}(S_d > t \mid \{\theta_h\}_{h=1}^d) \mathbb{1}_{\{\theta_1 < \tilde{\theta}\}} \right] + \mathbb{P}(\theta_1 \geq \tilde{\theta}),$$

where $\tilde{\theta} = t\delta \log d / 6d^{1+\delta}$. Conditioned on $\{\theta_h\}_{h=1}^d$, S_d is the sum of independent Bernoulli random variables with success probabilities

$$p_h := \mathbb{P}(\lambda_h \neq 0 \mid \{\theta_h\}_{h=1}^d) = \theta_h,$$

with the property $p_1 > p_2 > \dots > p_d$ by construction. In addition, on the event $\{\theta_1 < \tilde{\theta}\}$, we have $p_1 = \theta_1 \leq t\delta \log d / 6d^{1+\delta} \leq t/d < 1$, so we can apply Lemma 5.5:

$$\begin{aligned} \mathbb{P}(S_d > t \mid \{\theta_h\}_{h=1}^d) &= \mathbb{P}(S_d > \frac{t}{d}d \mid \{\theta_h\}_{h=1}^d) \\ &\leq \left[\left\{ \theta_1 \frac{d}{t} \right\}^{t/d} e^{t/d} \right]^d \\ &\leq \left\{ \frac{t\delta \log d}{6d^{1+\delta}} \frac{d}{t} \right\}^t e^t \\ &= \left\{ \frac{\log d^{\delta/2}}{3d^\delta} \right\}^t e^t, \\ &\leq e^{-t(\delta/2) \log d}, \end{aligned} \tag{5.21}$$

where we used the fact that $\log x \leq (1/e)x$ for any $x > 0$ in the fifth line.

Now we bound $\mathbb{P}(\theta_1 \geq \tilde{\theta}) = \mathbb{P}(\nu_1 \geq \tilde{\theta})$, where $\nu_1 \sim \text{Beta}(\alpha, \kappa + 1)$ with $\kappa = d^{1+\delta}$. Since $0 < \alpha < 1$, Gautschi's inequality (Gautschi, 1959) implies

$$B(\alpha, \kappa + 1) = \frac{\Gamma(\alpha) \Gamma(\kappa + 1)}{\kappa + \alpha \Gamma(\kappa + \alpha)} > \frac{\Gamma(\alpha)}{\kappa + 1} \kappa^{1-\alpha} \geq \frac{\kappa^{1-\alpha}}{\kappa + 1}.$$

The last line of the previous display used the fact that $\Gamma(\alpha) \geq 1$ for $\alpha \in (0, 1)$, which follows from $\Gamma(\alpha)$ being monotonically decreasing on $(0, 1)$ with $\Gamma(1) = 1$. Then we have

$$\begin{aligned} \mathbb{P}(\nu_1 > \tilde{\theta}) &= \frac{1}{B(\alpha, \kappa + 1)} \int_{\tilde{\theta}}^1 \nu^{\alpha-1} (1-\nu)^{\kappa} d\nu \\ &\leq \frac{\kappa + 1}{\kappa^{1-\alpha}} \tilde{\theta}^{\alpha-1} \int_{\tilde{\theta}}^1 (1-\nu)^{\kappa} d\nu \\ &= \left(\frac{1}{\kappa \tilde{\theta}} \right)^{1-\alpha} \left(1 - \frac{t\delta \log d}{6d^{1+\delta}} \right)^{d^{1+\delta}+1} \\ &= \left(\frac{6}{t\delta \log d} \right)^{1-\alpha} \left(1 - \frac{t\delta \log d}{6d^{1+\delta}} \right)^{d^{1+\delta}+1} \\ &\leq \exp\{(\alpha-1) \log(t\delta \log d/6)\} \exp\left(-\frac{t\delta}{6} \log d\right), \end{aligned}$$

where we used the fact that $(1 - b/x)^x \leq e^{-b}$ for $b \in \mathbb{R}$ and $x > 0$ in the last line. Next, we note that $t\delta \log d/6 \geq \delta \log d/6 \geq 1$ by the assumption that $\delta > 6/\log d$. It follows that the first term in the last line of the previous display is bounded above by 1, so that

$$\mathbb{P}(\nu_1 > \tilde{\theta}) \leq \exp(-t(\delta/6) \log d). \quad (5.22)$$

Combining (5.21) and (5.22), we obtain the desired bound. $\square$

Before proving Theorem 5.2 we need the following Lemma.

Lemma 5.3 (Evidence lower bound). *Suppose that the conditions of Theorem 5.2 hold. Then $\mathbb{P}_0^{(n)}(\mathcal{A}_n^c) = o(1)$, where the set $\mathcal{A}_n$ is*

$$\mathcal{A}_n = \left\{ Y_{ij} \in \{0, 1\}, 1 \leq i < j \leq n : \int \int \int \prod_{i < j} \frac{q_{ij}(Y_{ij})}{q_{0ij}(Y_{ij})} d\Pi(\boldsymbol{\beta}) d\Pi(\boldsymbol{\Lambda}) d\Pi(\mathbf{U}) \geq e^{-C_1 n(n-1) \epsilon_n^2} \right\}$$

for

$$\epsilon_n = M \sqrt{\frac{d_0 \log d}{n(n-1)}},$$

and some constant $C_1 > 1$ that only depends on δ .

Proof. Denote the KL-neighborhood for the model parameters by

$$\mathcal{B}_n = \left\{ (\boldsymbol{\beta}, \boldsymbol{\Lambda}, \mathbf{U}) : \frac{2}{n(n-1)} \sum_{i < j} K(q_{0ij}, q_{ij}) \leq \epsilon_n^2, \frac{2}{n(n-1)} \sum_{i < j} V(q_{0ij}, q_{ij}) \leq \epsilon_n^2 \right\}.$$

By Lemma 10 of Goshal and van der Vaart (2007), we have for any $C > 0$,

$$\mathbb{P}_0^{(n)} \left\{ \int_{\mathcal{B}_n} \prod_{i < j} \frac{q_{ij}(Y_{ij})}{q_{0ij}(Y_{ij})} d\Pi(\boldsymbol{\beta}) d\Pi(\boldsymbol{\Lambda}) d\Pi(\mathbf{U}) \geq e^{-(1+C)n(n-1)\epsilon_n^2} \mathbb{P}(\mathcal{B}_n) \right\} \geq 1 - \frac{1}{C^2 n(n-1)\epsilon_n^2}.$$

Therefore, it suffices to show that the prior probability $\mathbb{P}(\mathcal{B}_n) \geq e^{-C_2 n(n-1)\epsilon_n^2}$ for some constant $C_2 > 0$. By Taylor expanding the KL divergence and variation, Lemma 1 of Jeong and Ghoshal (2021) established that for two Bernoulli densities

$$\max \{K(q_{0ij}, q_{ij}), V(q_{0ij}, q_{ij})\} \leq \frac{1}{4}(\theta_{ij} - \theta_{0ij})^2 + o\{(\theta_{ij} - \theta_{0ij})^2\}.$$

Therefore, both $K(q_{0ij}, q_{ij})$ and $V(q_{0ij}, q_{ij})$ can be bounded above by a constant multiple of $(\theta_{ij} - \theta_{0ij})^2$ for sufficiently large n . Thus, we have for some constant $b_1 > 0$ and sufficiently large n ,

$$\begin{aligned} \mathbb{P}(\mathcal{B}_n) &\geq \mathbb{P} \left(\sum_{i < j} (\theta_{ij} - \theta_{0ij})^2 \leq b_1 n(n-1)\epsilon_n^2 \right) \\ &\geq \mathbb{P} \left(\sum_{i=1}^n \sum_{j=1}^n (\theta_{ij} - \theta_{0ij})^2 \leq b_1 n(n-1)\epsilon_n^2 \right) \\ &= \mathbb{P} \left(\|\mathbf{W} + \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T - \mathbf{W}_0 - \mathbf{U}_0\boldsymbol{\Lambda}_0\mathbf{U}_0^T\|_F^2 \leq b_1 n(n-1)\epsilon_n^2 \right) \\ &\geq \mathbb{P} \left(\|\mathbf{W} - \mathbf{W}_0\|_F \leq \sqrt{\frac{b_1 n(n-1)}{4}} \epsilon_n \right) \mathbb{P} \left(\|\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T - \mathbf{U}_0\boldsymbol{\Lambda}_0\mathbf{U}_0^T\|_F \leq \sqrt{\frac{b_1 n(n-1)}{4}} \epsilon_n \right) \end{aligned}$$

where the last line follows from the triangle inequality and the fact that $\boldsymbol{\beta}$ is independent of the remaining variables under our prior specification.

Let $\mathbf{U}_{0,h} \in \mathbb{R}^n$ and $\mathbf{U}_h \in \mathbb{R}^n$ denote the h th column of $\mathbf{U}_0$ and $\mathbf{U}$, respectively. Furthermore, we define $\tilde{\boldsymbol{\Lambda}}_0 = \text{diag}(\tilde{\boldsymbol{\lambda}}_0) \in \mathbb{R}^{n \times n}$ and $\tilde{\mathbf{U}}_0 \in \mathbb{R}^{n \times n}$ such that $\tilde{\lambda}_{0h} = \lambda_{0h}$ and $\tilde{\mathbf{U}}_{0,h} = \mathbf{U}_{0,h}$ for $h \leq d_0$ and $\tilde{\lambda}_{0h} = 0$ and $\tilde{\mathbf{U}}_{0,h} = \mathbf{0}_n$ for $h > d_0$. In particular, this implies $[\tilde{\mathbf{U}}_{0,1}, \dots, \tilde{\mathbf{U}}_{0,d_0}] \in \mathcal{V}_{d_0,n}$. Next, we have

$$\begin{aligned} \|\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T - \mathbf{U}_0\boldsymbol{\Lambda}_0\mathbf{U}_0^T\|_F &= \|\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T - \tilde{\mathbf{U}}_0\tilde{\boldsymbol{\Lambda}}_0\tilde{\mathbf{U}}_0^T\|_F \\ &= \|\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T - \mathbf{U}\tilde{\boldsymbol{\Lambda}}_0\mathbf{U}^T + \mathbf{U}\tilde{\boldsymbol{\Lambda}}_0\mathbf{U}^T - \tilde{\mathbf{U}}_0\tilde{\boldsymbol{\Lambda}}_0\tilde{\mathbf{U}}_0^T\|_F \\ &\leq \|\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T - \mathbf{U}\tilde{\boldsymbol{\Lambda}}_0\mathbf{U}^T\|_F + \|\mathbf{U}\tilde{\boldsymbol{\Lambda}}_0\mathbf{U}^T - \tilde{\mathbf{U}}_0\tilde{\boldsymbol{\Lambda}}_0\tilde{\mathbf{U}}_0^T\|_F \\ &= \|\boldsymbol{\Lambda} - \tilde{\boldsymbol{\Lambda}}_0\|_F + \left\| \sum_{h=1}^{d_0} \lambda_{0h} (\mathbf{U}_h \mathbf{U}_h^T - \mathbf{U}_{0,h} \mathbf{U}_{0,h}^T) \right\|_F \\ &\leq \|\boldsymbol{\Lambda} - \tilde{\boldsymbol{\Lambda}}_0\|_F + \sum_{h=1}^{d_0} |\lambda_{0h}| \|\mathbf{U}_h \mathbf{U}_h^T - \mathbf{U}_{0,h} \mathbf{U}_{0,h}^T\|_F \\ &\leq \|\boldsymbol{\Lambda} - \tilde{\boldsymbol{\Lambda}}_0\|_F + \|\boldsymbol{\lambda}_0\|_\infty \sum_{h=1}^{d_0} \|\mathbf{U}_h \mathbf{U}_h^T - \mathbf{U}_{0,h} \mathbf{U}_{0,h}^T\|_F. \end{aligned}$$

Combining this with the previous inequality, we have

$$\begin{aligned} \mathbb{P}(\mathcal{B}_n) &\geq \mathbb{P}\left(\|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_2 \leq \sqrt{\frac{b_1 n(n-1)}{4p \max_{1 \leq k \leq p} \|\mathbf{X}_k\|_F^2}} \epsilon_n\right) \times \\ &\quad \mathbb{P}\left(\|\mathbf{A} - \tilde{\mathbf{A}}_0\|_F \leq \sqrt{\frac{b_1 n(n-1)}{16}} \epsilon_n\right) \mathbb{P}\left(\sum_{h=1}^{d_0} \|\mathbf{U}_h \mathbf{U}_h^T - \mathbf{U}_{0,h} \mathbf{U}_{0,h}^T\|_F \leq \sqrt{\frac{b_1 n(n-1)}{16}} \frac{\epsilon_n}{\|\boldsymbol{\lambda}_0\|_\infty}\right), \end{aligned} \quad (5.23)$$

where we used the fact that $\mathbf{A}$ and $\mathbf{U}$ are independent under the prior. Now we bound each term in Equation (5.23) from below.

We start with the last term. From Assumption (A5.4), we assume that $\|\boldsymbol{\lambda}_0\|_\infty \leq M$, so we have the following lower bound

$$\begin{aligned} \mathbb{P}\left(\sum_{h=1}^{d_0} \|\mathbf{U}_h \mathbf{U}_h^T - \mathbf{U}_{0,h} \mathbf{U}_{0,h}^T\|_F \leq \sqrt{\frac{b_1 n(n-1)}{16}} \frac{\epsilon_n}{\|\boldsymbol{\lambda}_0\|_\infty}\right) \\ \geq \mathbb{P}\left(\sum_{h=1}^{d_0} \|\mathbf{U}_h \mathbf{U}_h^T - \mathbf{U}_{0,h} \mathbf{U}_{0,h}^T\|_F \leq \sqrt{\frac{b_1 n(n-1)}{16}} \frac{\epsilon_n}{M}\right) \\ = \mathbb{P}\left(\frac{1}{\sqrt{2}} \sum_{h=1}^{d_0} \|\mathbf{U}_h \mathbf{U}_h^T - \mathbf{U}_{0,h} \mathbf{U}_{0,h}^T\|_F \leq \sqrt{\frac{b_1 d_0 \log d}{32}}\right). \end{aligned} \quad (5.24)$$

We assert that $\sum_{h=1}^{d_0} \|\mathbf{U}_h \mathbf{U}_h^T - \mathbf{U}_{0,h} \mathbf{U}_{0,h}^T\|_F / \sqrt{2} \leq d_0$. Indeed,

$$\begin{aligned} \sum_{h=1}^{d_0} \|\mathbf{U}_h \mathbf{U}_h^T - \mathbf{U}_{0,h} \mathbf{U}_{0,h}^T\|_F &= \sum_{h=1}^{d_0} \left[\|\mathbf{U}_h \mathbf{U}_h^T\|_F^2 + \|\mathbf{U}_{0,h} \mathbf{U}_{0,h}^T\|_F^2 - 2 \operatorname{tr}(\mathbf{U}_h \mathbf{U}_h^T \mathbf{U}_{0,h} \mathbf{U}_{0,h}^T) \right]^{1/2} \\ &= \sum_{h=1}^{d_0} \left[2 - 2(\mathbf{U}_h^T \mathbf{U}_{0,h})^2 \right]^{1/2} \\ &\leq \sqrt{2} d_0. \end{aligned}$$

Therefore, the last probability in Equation (5.24) equals one for sufficiently large n by Assumption (A5.2).

Next, we apply Lemma 5.4 to bound the second term in Equation (5.23) from below. Note that we satisfy the condition that $d_0 \leq d/\sqrt{2}$ for sufficiently large n by Assumption (A5.2), so the lemma implies that

$$\begin{aligned} \mathbb{P}\left(\|\mathbf{A} - \tilde{\mathbf{A}}_0\|_F \leq \sqrt{\frac{b_1 n(n-1)}{16}} \epsilon_n\right) &= \mathbb{P}\left(\|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}_0\|_2 \leq M \sqrt{b_1 d_0 \log d / 16}\right) \\ &\geq e^{-\tilde{C} d_0 \log [bd \vee (M \sqrt{b_1 d_0 \log d / 16})^{-1}] - M \sqrt{b_1 d_0 \log d / 16} - M d_0} \\ &\geq e^{-C_3 M^2 d_0 \log bd} \\ &\geq e^{-C_4 M^2 d_0 \log d}, \end{aligned}$$

for some constants C_3 and C_4 that depends on δ . In the third line of the previous display, we used the fact that $bd \vee (M \sqrt{b_1 d_0 \log d / 16})^{-1} = bd$ for sufficiently large n because $b \geq 1$. Also, we used Assumption (A5.3) in the last line of the previous display.

Lastly, we provide a lower bound for the first term in Equation (5.23). First, we have that

$$\sqrt{b_1 n(n-1)/(4p \max_{1 \leq k \leq p} \|\mathbf{X}_k\|_F^2)} \geq b_3$$

for some constant b_3 and n sufficiently large by Assumption (A5.5). Thus,

$$\begin{aligned} \mathbb{P}\left(\|\beta - \beta_0\|_2 \leq \sqrt{\frac{b_1 n(n-1)}{4p \max_{1 \leq k \leq p} \|\mathbf{X}_k\|_F^2}} \epsilon_n\right) &\geq \mathbb{P}(\|\beta - \beta_0\|_2 \leq b_3 \epsilon_n) \\ &\geq \prod_{k=1}^p \mathbb{P}(|\beta_k - \beta_{0k}| \leq \frac{b_3}{\sqrt{p}} \epsilon_n). \end{aligned}$$

For simplicity, let $b_4 = b_3/\sqrt{p}$. Since the prior density on $\beta_k \stackrel{\text{iid}}{\sim} N(0, \sigma_\beta^2)$ is bounded away from zero in a neighborhood of β_{0k} for $1 \leq k \leq p$, we have

$$\begin{aligned} \mathbb{P}(|\beta_k - \beta_{0k}| \leq b_4 \epsilon_n) &= (2\pi\sigma_\beta^2)^{-1/2} \int_{|\beta_k - \beta_{0k}| \leq b_4 \epsilon_n} e^{-\beta_k^2/2\sigma_\beta^2} d\beta_k \\ &\geq m_k (2\pi\sigma_\beta^2)^{-1/2} \int_{|\beta_k - \beta_{0k}| \leq b_4 \epsilon_n} d\beta_k \\ &= \tilde{b}_k \epsilon_n, \end{aligned}$$

where $\tilde{b}_k = 2b_4 m_k (2\pi\sigma_\beta^2)^{-1/2}$ and $m_k > 0$ is the lower bound on the normal density in the neighborhood of β_{0k} . Combining this result with the previous expression, we have that there is a constant b_4 such that for large enough n

$$\begin{aligned} \mathbb{P}\left(\|\beta - \beta_0\|_2 \leq \sqrt{\frac{b_1 n(n-1)}{4p \max_{1 \leq k \leq p} \|\mathbf{X}_k\|_F^2}} \epsilon_n\right) &\geq b_4 \epsilon_n^p \\ &= e^{-p \log(n(n-1)) + p \log M + (p/2) \log(d_0 \log d) + \log b_4} \\ &\geq e^{-b_5 p \log(n)} \\ &\geq e^{-b_5 (p/\gamma) \log d} \\ &\geq e^{-C_4 n(n-1) \epsilon_n^2}, \end{aligned}$$

for some constants $b_5 > 0$ and $C_4 > 0$, where we used Assumption (A5.1) on the second to last line.

Combining the previous three bounds for the probabilities in Equation (5.23) with $C_2 = \max(C_3, C_4)$ gives the desired bound on $\mathbb{P}(\mathcal{B}_n)$. The result follows with $C_1 = 1 + C + C_2$. $\square$

Proof of Theorem 5.2. As in Jeong and Ghoshal (2021), we write the posterior as

$$\mathbb{P}(\|\Lambda\|_0 > C d_0 \mid \mathbf{Y}) = \frac{\int \int \int_{\|\Lambda\|_0 > C d_0} \prod_{i < j} \frac{q_{ij}(Y_{ij})}{q_{0ij}(Y_{ij})} d\Pi(\Lambda) d\Pi(\beta) d\Pi(\mathbf{U})}{\int \int \int \prod_{i < j} \frac{q_{ij}(Y_{ij})}{q_{0ij}(Y_{ij})} d\Pi(\Lambda) d\Pi(\beta) d\Pi(\mathbf{U})} = \frac{N_n}{D_n}, \quad (5.25)$$

where $\Pi(\Lambda)$, $\Pi(\beta)$, and $\Pi(\mathbf{U})$ denote the prior distributions for Λ , β , and $\mathbf{U}$, respectively. Now, we introduce an event $\mathcal{A}_n$ with large probability under the true data generating process. In particular, let $\mathcal{A}_n = \{\mathbf{Y} : \int \int \int \prod_{i < j} \frac{q_{ij}(Y_{ij})}{q_{0ij}(Y_{ij})} d\Pi(\Lambda) d\Pi(\beta) d\Pi(\mathbf{U}) \geq e^{-C_1 n(n-1) \epsilon_n^2}\}$, where ϵ_n^2 is the rate in Lemma 5.3. If we decompose the probability in Equation (5.25) into the sum of two complementary conditional probabilities (conditioning

on $\mathcal{A}_n$ and its complement $\mathcal{A}_n^c$), we can write:

$$\mathbb{E}_0^{(n)} [\mathbb{P}(\|\mathbf{\Lambda}\|_0 > Cd_0 \mid \mathbf{Y})] \leq \mathbb{E}_0^{(n)} [\mathbb{P}(\|\mathbf{\Lambda}\|_0 > Cd_0 \mid \mathbf{Y}) \mathbb{1}_{\mathcal{A}_n}] + \mathbb{P}_0^{(n)}(\mathcal{A}_n^c),$$

where $\mathbb{P}_0^{(n)}$ denotes the probability under the true data generating process. On the event $\mathcal{A}_n$, D_n is bounded below by $e^{-C_1 M^2 d_0 \log d}$. On the other hand, the expected value of the numerator can be bounded above by $\mathbb{P}(\|\mathbf{\Lambda}\|_0 > Cd_0)$ using Fubini's theorem:

$$\begin{aligned} \mathbb{E}_0^{(n)} \left[\int \int \int_{\|\mathbf{\Lambda}\|_0 > Cd_0} \prod_{i < j} \frac{q_{ij}(Y_{ij})}{q_{0ij}(Y_{ij})} d\Pi(\mathbf{\Lambda}) d\Pi(\boldsymbol{\beta}) d\Pi(\mathbf{U}) \right] = \\ \int \int \int_{\|\mathbf{\Lambda}\|_0 > Cd_0} \prod_{i < j} \mathbb{E}_0^{(n)} \left[\frac{q_{ij}(Y_{ij})}{q_{0ij}(Y_{ij})} \right] d\Pi(\mathbf{\Lambda}) d\Pi(\boldsymbol{\beta}) d\Pi(\mathbf{U}) \leq \mathbb{P}(\|\mathbf{\Lambda}\|_0 > Cd_0). \end{aligned}$$

Therefore, we can use Lemma 5.3 to conclude that

$$\mathbb{E}_0^{(n)} [\mathbb{P}(\|\mathbf{\Lambda}\|_0 > Cd_0 \mid \mathbf{Y})] \leq \mathbb{P}(\|\mathbf{\Lambda}\|_0 > Cd_0) e^{C_1 M^2 d_0 \log d} + o(1).$$

In addition, for n sufficiently large we have that $\delta > 6/\log d$, so we can apply Theorem 5.1 to bound the previous expression from above as

$$\mathbb{E}_0^{(n)} [\mathbb{P}(\|\mathbf{\Lambda}\|_0 > Cd_0 \mid \mathbf{Y})] \leq 3e^{-(C(\delta/6) - C_1 M^2) d_0 \log d} + o(1),$$

for n large enough, which goes to zero as $n \rightarrow \infty$ for $C > C_1(6/\delta)M^2 \vee 1$. □

Auxiliary Lemmas

Lemma 5.4. *Let $\boldsymbol{\lambda} \sim SS\text{-IBP}_d(\alpha, d^{1+\delta}, \mathbb{P}_{\text{spike}}, \mathbb{P}_{\text{slab}})$ with $\delta > 0$, $\mathbb{P}_{\text{slab}} = N(0, \sigma_h^2)$, $\mathbb{P}_{\text{spike}} = \delta_0$, and $\sigma_h^2 \stackrel{iid}{\sim} \text{Exponential}(1/2b^2)$. If $\boldsymbol{\lambda}_0 \in \mathbb{R}^d$ has nonzero support on the first d_0 entries with $1 \leq d_0 \leq d/\sqrt{2}$, i.e., $\lambda_{0j} \neq 0$ for $j \leq d_0$ and $\lambda_{0j} = 0$ for $j > d_0$, then for any $\epsilon > 0$,*

$$\mathbb{P}(\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_2 \leq \epsilon) \geq e^{-\epsilon/2b} e^{-\|\boldsymbol{\lambda}_0\|_1/b} \alpha^{d_0} e^{-Cd_0 \log(bd \vee \epsilon^{-1})},$$

for some positive constant C_1 that only depends on δ . Moreover, if $\|\boldsymbol{\lambda}_0\|_\infty \leq M$ for some $M > 0$, $b \geq 1$, and $\alpha = 1/d$, we have

$$\mathbb{P}(\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_2 \leq \epsilon) \geq e^{-C_2 d_0 \log(bd \vee \epsilon^{-1}) - \epsilon - d_0 M},$$

for some positive constant C_2 that depends only on δ .

Proof. Conditioned on $\{\theta_h\}_{h=1}^d$, it follows from Lemma 5.7 that

$$\begin{aligned} \mathbb{P}(\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_2 \leq \epsilon \mid \{\theta_h\}_{h=1}^d) &\geq \theta_{d_0}^{d_0} (1 - \theta_1)^{d-d_0} \left[e^{-\|\boldsymbol{\lambda}_0\|_1/b - \epsilon/\sqrt{2}b} \left(\frac{\epsilon}{\sqrt{2}bd_0} \right)^{d_0} \right] \\ &\geq e^{-\|\boldsymbol{\lambda}_0\|_1/b - \epsilon/\sqrt{2}b - 2d_0 \log(\sqrt{2}bd_0 \vee \epsilon^{-1})} \left[\theta_{d_0}^{d_0} (1 - \theta_1)^{d-d_0} \right] \\ &\geq e^{-\|\boldsymbol{\lambda}_0\|_1/b - \epsilon/\sqrt{2}b - 2d_0 \log(bd \vee \epsilon^{-1})} \left[\theta_{d_0}^{d_0} (1 - \theta_1)^{d-d_0} \right], \end{aligned}$$

where in the last line we used $\sqrt{2}d_0 \leq d$. Focusing on the terms involving $\{\theta_h\}_{h=1}^d$, we have

$$\begin{aligned}\theta_{d_0}^{d_0}(1-\theta_1)^{d-d_0} &= \left(\prod_{h=1}^{d_0} \nu_h\right)^{d_0} (1-\nu_1)^{d-d_0} \\ &= \left(\prod_{h=2}^{d_0} \nu_h^{d_0}\right) \nu_1^{d_0} (1-\nu_1)^{d-d_0},\end{aligned}$$

so that

$$\mathbb{E}\left[\theta_{d_0}^{d_0}(1-\theta_1)^{d-d_0}\right] = \left[\mathbb{E}(\nu_2^{d_0})\right]^{d_0-1} \mathbb{E}\left[\nu_1^{d_0}(1-\nu_1)^{d-d_0}\right],$$

due to the independence of $\nu_1, \dots, \nu_d$ and the fact that $\nu_2, \dots, \nu_d$ are identically distributed. Since $B(\alpha, d^{1+\delta} + 1) < B(\alpha, 1) = 1/\alpha$, $\alpha - 1 < 0$, and $d^{1+\delta} > d - d_0$, it follows that

$$\begin{aligned}\mathbb{E}\left[\nu_1^{d_0}(1-\nu_1)^{d-d_0}\right] &= \frac{1}{B(\alpha, d^{1+\delta} + 1)} \int_0^1 \nu_1^{d_0+\alpha-1} (1-\nu_1)^{d-d_0+d^{1+\delta}} d\nu_1 \\ &\geq \alpha \left(1 - \frac{1}{d^{1+\delta}}\right)^{2d^{1+\delta}} \int_0^{d^{-(1+\delta)}} \nu_1^{d_0} d\nu_1 \\ &\geq \frac{\alpha}{(d_0 + 1)} e^{-4} \left(\frac{1}{d^{1+\delta}}\right)^{d_0+1} \\ &\geq \alpha e^{-C_1 d_0 \log d}\end{aligned}$$

for some $C_1 > 0$ depending only on δ , where the second inequality used the fact that $(1-x)^{1/x} \geq e^{-2}$ for $0 < x < 1/2$. Similarly we have

$$\begin{aligned}\mathbb{E}\left(\nu_2^{d_0}\right) &= \frac{1}{B(\alpha, 1)} \int_0^1 \nu_2^{d_0+\alpha-1} d\nu_2 \\ &\geq \alpha \int_0^1 \nu_2^{d_0} d\nu_2 \\ &= \frac{\alpha}{d_0 + 1} \\ &\geq \alpha e^{-\log d},\end{aligned}$$

where the second line used the fact that $\alpha < 1$. Combining these results, we recover the desired bound:

$$\mathbb{P}(\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_2 \leq \epsilon) = \mathbb{E}\left[\mathbb{P}(\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_2 \leq \epsilon \mid \{\theta\}_{h=1}^d)\right] \geq e^{-\epsilon/\sqrt{2}b} e^{-\|\boldsymbol{\lambda}_0\|_1/b} \alpha^{d_0} e^{-C_4 d_0 \log(bd \vee \epsilon^{-1})},$$

where C_4 is a positive constant that only depends on δ .

For the second assertion, note that $\|\boldsymbol{\lambda}_0\|_\infty \leq M$ implies

$$\|\boldsymbol{\lambda}_0\|_1 \leq d_0 \|\boldsymbol{\lambda}_0\|_\infty \leq d_0 M,$$

which completes the proof. $\square$

Lemma 5.5. *Let $X_1, \dots, X_n$ be independent Bernoulli random variables with $\mathbb{P}(X_i = 1) = p_i$ for $i = 1, \dots, n$,*

such that $p_1 > p_2 > \dots > p_n$. Let $S_n = \sum_{i=1}^n X_i$ and $p = \sum_{i=1}^n p_i$, then

$$\mathbb{P}(S_n > an) \leq \left[\left(\frac{p_1}{a} \right)^a e^a \right]^n$$

for $p_1 \leq a < 1$.

Proof. The proof follows the derivation of a similar Chernoff bound found in Hagerup and Rüb (1990). For $t \geq 0$, we have that

$$\begin{aligned} \mathbb{P}(S_n \geq an) &\leq e^{-tan} \mathbb{E}(e^{tS_n}) \\ &= e^{-tan} \prod_{i=1}^n (1 + p_i(e^t - 1)) \\ &\leq e^{-tan} (1 + p_1(e^t - 1))^n, \end{aligned}$$

where the last line follows from the ordering of the p_i . For $t = \log(a(1 - p_1)/[p_1(1 - a)])$, this become

$$\mathbb{P}(S_n \geq an) \leq \left[\left(\frac{p_1}{a} \right)^a \left(\frac{1 - p_1}{1 - a} \right)^{1-a} \right]^n, \quad p_1 \leq a < 1.$$

Note that

$$\left(\frac{1 - p_1}{1 - a} \right)^{1-a} = \left(1 + \frac{a - p_1}{1 - a} \right)^{1-a} \leq e^{a - p_1} \leq e^a, \quad \text{for } p_1 \leq a,$$

so that

$$\mathbb{P}(S_n \geq an) \leq \left[\left(\frac{p_1}{a} \right)^a e^a \right]^n.$$

Noting that $\mathbb{P}(S_n > an) \leq \mathbb{P}(S_n \geq an)$ gives the desired bound. $\square$

Lemma 5.6. Assume $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_d)^T$ is distributed as $\lambda_j \stackrel{iid}{\sim} \text{Laplace}(b)$ for $1 \leq j \leq d$. Then for any $\boldsymbol{\lambda}_0 \in \mathbb{R}^d$ and any $\epsilon > 0$,

$$\mathbb{P}(\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_1 \leq \epsilon) \geq e^{-\frac{\|\boldsymbol{\lambda}_0\|_1}{b} - \frac{\epsilon}{b} - d \log(bd/\epsilon)}.$$

Proof. Using the change-of-variables $\mathbf{u} = \boldsymbol{\lambda} - \boldsymbol{\lambda}_0$, we get

$$\begin{aligned} \mathbb{P}(\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_1 \leq \epsilon) &= \int_{\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_1 \leq \epsilon} \left(\frac{1}{2b} \right)^d e^{-\|\boldsymbol{\lambda}\|_1/b} d\boldsymbol{\lambda} \\ &\geq e^{-\|\boldsymbol{\lambda}_0\|_1/b} \int_{\|\mathbf{u}\|_1 \leq \epsilon} \left(\frac{1}{2b} \right)^d e^{-\|\mathbf{u}\|_1/b} d\mathbf{u} \\ &\geq e^{-\|\boldsymbol{\lambda}_0\|_1/b} \mathbb{P}\left(\sum_{i=1}^d E_i \leq \epsilon \right), \end{aligned}$$

where $E_1, \dots, E_d$ are iid exponential random variables with scale b . Recall that the sum of n iid exponential random variables with scale θ follows a gamma distribution with shape n and scale θ . Thus, we have

$$\begin{aligned} \mathbb{P}\left(\sum_{i=1}^d E_i \leq \epsilon \right) &= \frac{1}{\Gamma(d)b^d} \int_0^\epsilon x^{d-1} e^{-xb} dx \\ &\geq \frac{1}{b^d(d-1)!} e^{-\epsilon/b} \int_0^\epsilon x^{d-1} = \frac{\epsilon^d}{b^d d!} e^{-\epsilon/b}. \end{aligned}$$

The fact that $d! \leq d^d$ completes the proof. $\square$

Lemma 5.7. Assume $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_d)^T$ is distributed as

$$\begin{aligned} \lambda_j \mid \xi_j, \sigma_j^2 &\stackrel{\text{ind.}}{\sim} \xi_j N(0, \sigma_j^2) + (1 - \xi_j) \delta_0 \\ \xi_j &\stackrel{\text{ind.}}{\sim} \text{Bernoulli}(\theta_j) \quad \sigma_j^2 \stackrel{\text{iid}}{\sim} \text{Exponential}(1/2b^2), \end{aligned}$$

for $\theta_1 > \theta_2 > \dots > \theta_d$ with $\theta_j \in (0, 1)$ for $1 \leq j \leq d$. Assume $\boldsymbol{\lambda}_0 = (\lambda_{01}, \dots, \lambda_{0d})^T \in \mathbb{R}^d$ has nonzero support on the first d_0 entries, i.e., $\lambda_{0j} \neq 0$ for $j \leq d_0$ and $\lambda_{0j} = 0$ for $j > d_0$. Then for any $\epsilon > 0$,

$$\mathbb{P}(\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_2 \leq \epsilon) \geq \theta_{d_0}^{d_0} (1 - \theta_1)^{d-d_0} \left[e^{-\frac{\|\boldsymbol{\lambda}_0\|_1}{b} - \epsilon/\sqrt{2}b} \left(\frac{\epsilon}{\sqrt{2}bd_0} \right)^{d_0} \right].$$

Proof. First, we can marginalize out ξ_j and σ_j^2 for $1 \leq j \leq d$ to conclude that

$$\lambda_j \mid \theta_j \stackrel{\text{ind.}}{\sim} \theta_j \text{Laplace}(b) + (1 - \theta_j) \delta_0.$$

Next, we have the inequality

$$\mathbb{P}(\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_2 \leq \epsilon) \geq \left[\prod_{j>d_0} \mathbb{P} \left(|\lambda_j| \leq \frac{\epsilon}{\sqrt{2(d-d_0)}} \right) \right] \mathbb{P} \left(\|\lambda_{1:d_0} - \lambda_{01:d_0}\|_2 \leq \frac{\epsilon}{\sqrt{2}} \right).$$

Note that $\mathbb{P}(|\lambda_j| \leq \epsilon/\sqrt{2(d-d_0)}) \geq \mathbb{P}(\lambda_j = 0) = 1 - \theta_j \geq 1 - \theta_1$. Also,

$$\begin{aligned} \mathbb{P} \left(\|\lambda_{1:d_0} - \lambda_{01:d_0}\|_2 \leq \frac{\epsilon}{\sqrt{2}} \right) &\geq \left[\prod_{j \leq d_0} \theta_j \right] \mathbb{P}_{\text{Laplace}(b)} \left(\|\lambda_{1:d_0} - \lambda_{01:d_0}\|_2 \leq \frac{\epsilon}{\sqrt{2}} \right) \\ &\geq \theta_{d_0}^{d_0} \mathbb{P}_{\text{Laplace}(b)} \left(\|\lambda_{1:d_0} - \lambda_{01:d_0}\|_1 \leq \frac{\epsilon}{\sqrt{2}} \right), \end{aligned}$$

where $\mathbb{P}_{\text{Laplace}(b)}$ denotes the probability measure under a $\text{Laplace}(b)$ density. Applying Lemma 5.6 completes the proof. $\square$

5.8.5 Additional Figures

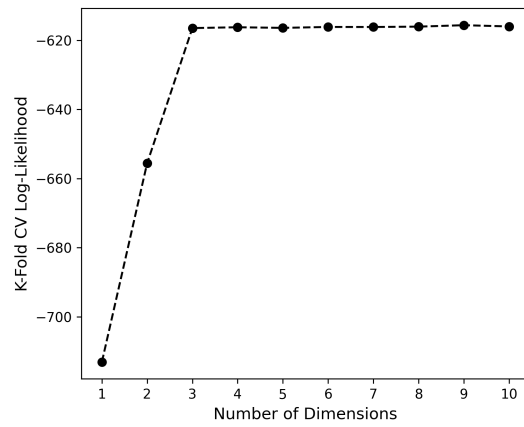


Figure 5.12: Dimension selection using K-Fold cross-validation with $K = 4$ for the static network eigenmodel with a $N(0, nI_d)$ prior for $\mathbf{\Lambda}$.

References

- Adragni, K. P., & Cook, R. D. (2009). Sufficient dimension reduction and prediction in regression. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906), 4385–4405.
- Airoldi, E. M., Blei, D. M., Fienberg, S. E., & Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9, 1981–2014.
- Aliverti, E., & Russo, M. (2022). Stratified stochastic variational inference for high-dimensional network factor model. *Journal of Computational and Graphical Statistics*, 31(2), 502–511.
- Amaratunga, D., Cabrera, J., & Lee, Y.-S. (2008). Enriched random forests. *Bioinformatics*, 24(18), 2010–2014.
- Anderson, R. M., & May, R. M. (1991). *Infectious diseases of humans: Dynamics and control* (Vol. 1). Oxford University Press.
- Andersson, H. (1997). Epidemics in a population with social structure. *Mathematical Biosciences*, 14(2), 79–84.
- Andersson, H. (1998). Limit theorems for a random graph epidemic model. *The Annals of Applied Probability*, 8(4), 1331–1349.
- Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. *The Annals of Statistics*, 47(2), 1148–1178.
- Athreya, A., Fishkind, D. E., Tang, M., Priebe, C. E., Park, Y., Vogelstein, J. T., Levin, K., Lyzinski, V., Qin, Y., & Sussman, D. L. (2018). Statistical inference on random dot product graphs: A survey. *Journal of Machine Learning Research*, 18(226), 1–92.
- Barabási, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439), 509–512.
- Barber, D., & Chiappa, S. (2007). Unified inference for variational Bayesian linear Gaussian state-space models. *Advances in neural information processing systems* (pp. 81–88).
- Battacharya, A., & Dunson, D. (2011). Sparse Bayesian infinite factor models. *Biometrika*, 98(2), 291–306.
- Beal, M. J. (2003). *Variational algorithms for approximate bayesian inference* (Doctoral dissertation). University of London. London.
- Beveridge, A., & Chemers, M. (2018). The game of ‘The Game of Thrones’: Networked concordances and fractal dramaturgy. In P. Brembilla & I. A. D. Pascalis (Eds.), *Reading contemporary serial television universes: A narrative ecosystem framework*. Routledge.
- Biau, G., & Scornet, E. (2016). A random forest guided tour. *TEST*, 25, 197–227.
- Binder, D. A. (1978). Bayesian cluster analysis. *Biometrika*, 65(1), 31–38.
- Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P. A., Horsfall, P., & Goodman, N. D. (2019). Pyro: Deep universal probabilistic programming. *Journal of Machine Learning Research*, 20, 28:1–28:6. <http://jmlr.org/papers/v20/18-403.html>

- Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518), 859–877.
- Boccaletti, S., Bianconi, G., Criado, R., del Genio, C. I., Gómez-Gardeñes, J., Romance, M., Sendiña-Nadal, I., Wang, Z., & Zanin, M. (2014). The structure and dynamics of multilayer networks. *Physics Reports*, 544(1), 1–122.
- Bollobás, B., & Riordan, O. M. (2003). Mathematical results on scale-free random graphs. In S. Bornholdt & H. G. Schuster (Eds.), *Handbook of graphs and networks: From the genome to the internet* (pp. 1–34). Wiley-VCH Verlag GmbH & Co. KGaA.
- Boschee, E., Lautenschlager, J., O'Brien, S., Shellman, S., Starz, J., & Ward, M. (2015). *ICEWS Coded Event Data*. <https://doi.org/10.7910/DVN/28075>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32.
- Breiman, L., Freidman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Wadsworth International.
- Butland, G., Peregrín-Alvarez, J. M., Li, J., Yang, W., Yang, X., Candien, V., Starostine, A., Richards, D., Beattie, B., Krogan, N., Davey, M., Parkinson, J., Greenblatt, J., & Emili, A. (2005). Interaction network containing conserved and essential protein complexes in *Escherichia coli*. *Nature*, 433(7025).
- Byrne, S., & Girolami, M. (2013). Geodesic Monte Carlo on embedded manifolds. *Scandinavian Journal of Statistics, Theory and Applications*, 40(4), 825–845.
- Castillo, I., & van der Vaart, A. (2012). Needles and straw in a haystack: Posterior concentration for possible sparse sequences. *The Annals of Statistics*, 40(4), 2069–2101.
- Cattuto, C., Van den Broeck, W., Barrat, A., Colizza, V., Pinton, J., & Vespignani, A. (2010). Dynamics of person-to-person interactions from distributed RFID sensor networks. *PLoS ONE*, 5(7), e11596.
- Center for Systematic Peace. (2022). Polity V: Political regime characteristics and transitions, 1800–2018 [Accessed 11th May 2022].
- Chen, K., & 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, Z., Chen, D., Zhao, C., Kwan, M., Cai, J., Zhuang, Y., Zhao, B., Wang, X., Chen, B., Yang, J., Li, R., He, B., Gao, B., Wang, K., & Xu, B. (2020). Influence of meteorological conditions on PM_{2.5} concentrations across China: A review of methodology and mechanism. *Environmental International*, 139, 105558.
- Chiaromonte, F., Cook, R. D., & Li, B. (2002). Sufficient dimension reduction in regressions with categorical predictors. *The Annals of Statistics*, 30(2), 475–497.
- Chikuse, Y. (2003a). *Statistics on special manifolds* (Vol. 174). Springer Science+Business Media.
- Chikuse, Y. (2003b). Concentrated matrix Langevin distributions. *Journal of Multivariate Analysis*, 85(2), 375–394.
- Chikuse, Y. (2006). State space models on special manifold. *Journal of Multivariate Analysis*, 97(6), 1284–1294.
- Clauset, A., Shalizi, C. R., & Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Review*, 51(4), 661–703.
- Cook, R. D. (1998). *Regression graphics: Ideas for studying regression through graphics*. Wiley.
- Cook, R. D., & Weisberg, S. (1991). Sliced inverse regression for dimension reduction: Comment. *Journal of the American Statistical Association*, 86(414), 328–332.

- Cui, Y., Zhu, R., Zhou, M., & Kosorok, M. R. (in press). Consistency of survival tree and forest models: Splitting bias and correction. *Statistica Sinica*.
- D’Angelo, S., Murphy, T. B., & Altó, M. (2019). Latent space modelling of multidimensional networks with applications to the exchange of votes in Eurovision song contest. *Annals of Applied Statistics*, 13(2), 900–930.
- Daudin, J.-J., Picard, F., & Robin, S. (2008). A mixture model for random graphs. *Statistics and Computing*, 18(2), 151–171.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39(1), 1–38.
- Diaz-Urriarte, R., & de Andrés, S. A. (2006). Gene selection and classification of microarray data using random forests. *BMC Bioinformatics*, 7(3).
- Domenico, M. D., Lima, A., Mougél, P., & Musolesi, M. (2013). The anatomy of a scientific rumor. *Scientific Reports*, 3(2980).
- Dua, D., & Graff, C. (2017). UCI machine learning repository [<http://archive.ics.uci.edu/ml>] [Irvine, CA: University of California, School of Information and Computer Science]. <http://archive.ics.uci.edu/ml>
- Durante, D. (2017). A note on the multiplicative gamma process. *Statistics & Probability Letters*, 122(1), 198–204.
- Durante, D., & Dunson, D. B. (2014). Nonparametric Bayes dynamic modelling of relational data. *Biometrika*, 101(4), 883–898.
- Durante, D., Mukherjee, N., & Steorts, R. C. (2017). Bayesian learning of dynamic multilayer networks. *Journal of Machine Learning Research*, 18(43), 1–29.
- Duval, R. D., & Thompson, W. R. (1980). Reconsidering the aggregate relationship between size, economic development, and some types of foreign policy behavior. *American Journal of Political Science*, 24(3), 511–525.
- Erdős, P., & Rényi, A. (1959). On random graphs. *Publicationes Mathematicae Debrecen*, 6(290), 290–297.
- Escobar, M. D., & West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430), 577–588.
- Fan, J., & Gijbels, I. (1996). *Local polynomial modelling and its applications* (Vol. 66). Chapman; Hall.
- Fan, X., Cao, L., & Xu, R. Y. D. (2015). Dynamic infinite mixed-membership stochastic blockmodel. *IEEE Transactions on Neural Networks and Learning Systems*, 26(9), 2072–2085.
- Fox, E., Sudderth, E. B., Jordan, M. I., & Willsky, A. S. (2011). Bayesian nonparametric inference of switching dynamic linear models. *IEEE Transactions on Signal Processing*, 59(4), 1569–1585.
- Fox, E. B., Sudderth, E. B., Jordan, M. I., & Willsky, A. S. (2011). A sticky HDP-HMM with applications to speaker diarization. *Annals of Applied Statistics*, 5(2A), 1020–1056.
- Frank, O., & Strauss, D. (1986). Markov graphs. *Journal of the American Statistical Association*, 81(395), 832–842.
- Friedberg, R., Tibshirani, J., Athey, S., & Wager, S. (2021). Local linear forests. *Journal of Computational and Graphical Statistics*, 30(2), 503–517.
- Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1), 1–67.
- Friel, N., Rastelli, R., Wyse, J., & Raftery, A. E. (2016). Interlocking directorates in Irish companies using a latent space model for bipartite networks. *Proceedings of the National Academy of Sciences*, 113(24), 6629–6634.

- Fruchterman, T. M. J., & Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11), 1129–1164.
- Fu, W., Song, L., & Xing, E. P. (2009). Dynamic mixed membership blockmodel for evolving networks. *Proceedings of the 26th Annual International Conference of Machine Learning*, 329–336.
- Gautschi, W. (1959). Some elementary inequalities relating the gamma and incomplete gamma function. *Journal of Mathematics and Physics*, 38(1–4), 77–81.
- Genolini, C., & Falissard, B. (2010). Kml: K-means for longitudinal data. *Computational Statistics*, 25(2), 317–328.
- Gerner, D. J., Schrod, P. A., & Yilmaz, Ö. (2008). Conflict and mediation event observations (CAMEO): An event data framework for a post-cold war world. In J. Bercovitch & S. S. Gartner (Eds.), *International conflict mediation: New approaches and findings* (pp. 287–304). Routledge.
- Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63, 3–42.
- Gibler, D. M. (2009a). The changing nature of military alliances. In D. M. Gibler (Ed.), *International military alliances, 1648 - 2008* (pp. 553–556). CQ Press.
- Gibler, D. M. (2009b). *International military alliances, 1648 - 2008* (Vol. 2). CQ Press.
- Gilbert, E. N. (1959). Random graphs. *Annals of Mathematical Statistics*, 30(4), 1141–1144.
- Gleditsch, K. S. (2002). Expanded trade and GDP data. *The Journal of Conflict Resolution*, 46(5), 712–724.
- Gleditsch, K. S. (2013). Distance between capital cities [Accessed 11th May 2022].
- Goldenberg, A., Zheng, A. X., Fienberg, S. E., & Airoldi, E. M. (2010). A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2), 129–233.
- Gollini, I., & Murphy, T. B. (2016). Joint modeling of multiple network views. *Journal of Computational and Graphical Statistics*, 25(1), 246–265.
- Golub, G. G., & Loan, C. F. V. (2012). *Matrix computations* (Vol. 3). The Johns Hopkins University Press.
- Goodreau, S. M., Handcock, M. S., Hunter, D. R., Butts, C. T., & Morris, M. (2008). A statnet tutorial. *Journal of Statistical Software*, 24(9), 1–26. <https://doi.org/10.18637/jss.v024.i09>
- Goshal, S., & van der Vaart, A. (2007). Convergence rates of posterior distributions for noniid observations. *The Annals of Statistics*, 35(1), 192–223.
- Gower, J. C., & Dijksterhuis, G. B. (2004). *Procrustes problems*. Oxford University Press.
- Griffiths, T. L., & Ghahramani, Z. (2011). The Indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12(32), 1185–1224.
- Guha, S., & Rodriguez, A. (2021). Bayesian regression with undirected network predictors with an application to brain connectome data. *Journal of the American Statistical Association*, 116(534), 581–593.
- Guhaniyogi, R., & Rodriguez, A. (2020). Joint modeling of longitudinal relational data and exogenous variables. *Bayesian Analysis*, 15(2), 477–503.
- Hagerup, T., & Rüb, C. (1990). A guided tour of Chernoff bounds. *Information Processing Letters*, 33(6), 305–308.
- Handcock, M. S. (2003). *Assessing degeneracy in statistical models of social networks* (tech. rep. No. 39). Center for Statistics and the Social Science, University of Washington.
- Handcock, M. S., Raftery, A. E., & Tantrum, J. M. (2007). Model-based clustering of social networks. *Journal of the Royal Statistical Society - Series A*, 170(2), 301–354.
- Hanneke, S., Fu, W., & Xing, E. P. (2010). Discrete temporal models of social networks. *Electronic Journal of Statistics*, 4, 585–605.

- He, Y., & Hoff, P. D. (2019). Multiplicative coevolution regression models for longitudinal networks and nodal attributes. *Social Networks*, 57, 54–62.
- Ho, Q., Song, L., & Xing, E. P. (2011). Evolving cluster mixed-membership blockmodel for time-evolving networks. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 15, 342–350.
- Hoff, P. (2021). Additive and multiplicative effects network models. *Statistical Science*, 36(1), 34–50.
- Hoff, P. D. (2005). Bilinear mixed-effects models for dyadic data. *Journal of the American Statistical Association*, 100(469), 286–295.
- Hoff, P. D. (2008). Modeling homophily and stochastic equivalence in symmetric relational data. *Advances in neural information processing systems* (pp. 657–664).
- Hoff, P. D. (2009). Simulation of the matrix Bingham-von Mises-Fisher distribution, with applications to multivariate and relational data. *Journal of Computational and Graphical Statistics*, 18(2), 438–456.
- Hoff, P. D. (2015). Multilinear tensor regression for longitudinal relational data. *Annals of Applied Statistics*, 9(3), 1169–1193.
- Hoff, P. D., Raftery, A. E., & Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460), 1090–1098.
- Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(4), 1303–1347.
- Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn Sampler: Adaptively setting path length in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47), 1593–1623.
- Holland, P. W., Laskey, K. B., & Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5(2), 109–137.
- Holland, P. W., & Leinhardt, S. (1976). Local structure in social networks. *Sociological Methodology*, 7, 1–45.
- Holland, P. W., & Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373), 33–50.
- Holtom, P., Bromley, M., & Simmel, V. (2012). Measuring international arms transfers. *Stockholm International Peace Research Institute*.
- Hunter, D. R., Goodreau, S. M., & Handcock, M. S. (2008). Goodness of fit of social network models. *Journal of the American Statistical Association*, 103(408), 248–258.
- Hunter, D. R., & Handcock, M. S. (2006). Inference in curved exponential family models for networks. *Journal of Computational and Graphical Statistics*, 15(3), 565–583.
- Isard, W. (1954). Location theory and trade theory: Short-run analysis. *The Quarterly Journal of Economics*, 68(2), 305–320.
- Ishiguro, K., Iwata, T., Ueda, N., & Tenenbaum, J. B. (2010). Dynamic infinite relational model for time-varying relational data analysis. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, & A. Culotta (Eds.), *Advances in neural information processing systems 23* (pp. 919–927). Curran Associates, Inc.
- Ishwaran, H., & Zarepour, M. (2000). Markov chain Monte Carlo in approximate Dirichlet and beta two-parameter process hierarchical models. *Biometrika*, 87(2), 371–390.
- Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008). Random survival forests. *The Annals of Applied Statistics*, 2(3), 841–860.
- Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: Frequentist and Bayesian strategies. *Annals of Statistics*, 33(2), 730–773.

- Jauch, M., Hoff, P. D., & Dunson, D. B. (2021). Monte Carlo simulation on the Stiefel manifold via polar expansion. *Journal of Computational and Graphical Statistics*, 0(00), 1–10.
- Jeong, S., & Ghoshal, S. (2021). Posterior contraction in sparse generalized linear models. *Biometrika*, 108(2), 367–379.
- Karrer, B., & Newman, M. E. J. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1), 016107.
- Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., & Ueda, N. (2006). Learning systems of concepts with an infinite relational model. *Proceedings of the 21st National Conference on Artificial Intelligence*, 381–388.
- Khanin, R., & Wit, E. (2006). How scale-free are biological networks. *Journal of Computational Biology*, 13(3), 810–818.
- Kim, B., Lee, K. H., Xue, L., & Niu, X. (2018). A review of dynamic network models with latent variables. *Statistical Surveys*, 12, 105–135.
- Kim, M., & Leskovec, J. (2013). Nonparametric multi-group membership model for dynamic networks. *Advances in neural information processing systems 26* (pp. 1385–1393). Curran Associates, Inc.
- Kiss, I. Z., Green, D. M., & Kao, R. R. (2008). The effect of network mixing patterns on epidemic dynamics and the efficacy of disease contact tracing. *Journal of the Royal Society Interface*, 5(1), 791–799.
- Kiss, I. Z., Miller, J. C., & Simon, P. L. (2017). *Mathematics of epidemics on networks*. Springer.
- Kolaczyk, E. D. (2017). *Statistical analysis of network data*. Springer.
- Krivitsky, P. N., & Handcock, M. S. (2008). Fitting position latent cluster models for social networks with latentnet. *Journal of Statistical Software*, 24(5).
- Krivitsky, P. N., & Handcock, M. S. (2013). A separable model for dynamic networks. *Journal of the Royal Statistical Society, Series B*, 76(1), 29–46.
- Krivitsky, P. N., Handcock, M. S., & Raftery, A. E. (2009). Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, 31(3), 204–213.
- Krivitsky, P. N., & Morris, M. (2017). Inference for social network models from egocentrically sampled data, with application to understanding persistent racial disparities in HIV prevalence in the US. *Annals of Applied Statistics*, 11(1), 427–455.
- Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2), 83–97.
- Laber, E. B., & Zhao, Y. Q. (2015). Tree-based methods for individualized treatment regimes. *Biometrika*, 102(3), 501–514.
- Latouche, P., Birmelé, E., & Ambroise, C. (2011). Overlapping stochastic block models with application to the French political blogosphere. *The Annals of Applied Statistics*, 5(1), 309–336.
- Lau, J. W., & Green, P. J. (2012). Bayesian model-based clustering procedures. *Journal of Computational and Graphical Statistics*, 16(3), 526–558.
- Li, B. (2018). *Sufficient dimension reduction: Methods and applications with R*. Chapman; Hall/CRC.
- Li, K.-C. (1991). Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414), 316–327.
- Li, L., Levina, E., & Zhu, J. (2020). Network cross-validation by edge sampling. *Biometrika*, 107(2), 257–276.
- Lin, Q., Zhao, Z., & Liu, J. S. (2018). On consistency and sparsity for sliced inverse regression in high dimensions. *The Annals of Statistics*, 46(2), 580–610.

- Lin, Q., Zhao, Z., & Liu, J. S. (2019). Sparse sliced inverse regression via lasso. *Journal of the American Statistical Association*, 114(528), 1726–1739.
- Liu, J. S., & Wu, Y. N. (1999). Parameter expansion for data augmentation. *Journal of the American Statistical Association*, 94(448), 1264–1274.
- Louppe, G. (2014). Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502*.
- Loyal, J. D., & Chen, Y. (2020). Statistical network analysis: A review with applications to the Coronavirus Disease 2019 pandemic. *International Statistical Review*, 88(2), 419–440.
- Ma, Y., & Zhu, L. (2012). A semiparametric approach to dimension reduction. *Journal of the American Statistical Association*, 107(497), 168–179.
- Ma, Z., Ma, Z., & Yuan, H. (2020). Universal latent space model fitting for large networks with edge covariates. *Journal of Machine Learning Research*, 21(4), 1–67.
- MacDonald, P. W., Levina, E., & Zhu, J. (in press). Latent space models for multiplex networks with shared structure. *Biometrika*.
- Matias, C., & Miele, V. (2017). Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society - Series B*, 79(4), 1119–1141.
- Meilă, M. (2007). Comparing clusterings - an information based distance. *Journal of Multivariate Analysis*, 98(5), 873–895.
- Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, 7(35), 983–999.
- Mentch, L., & Hooker, G. (2016). Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *The Journal of Machine Learning Research*, 17(26), 841–881.
- Menze, B. H., Kelm, B. M., Splitthoff, D. N., Koethe, U., & Hamprecht, F. A. On oblique random forests. In: *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 2011, 453–469.
- Miller, K., Jordan, M. I., & Griffiths, T. L. (2009). Nonparametric latent feature models for link prediction. *Advances in Neural Information Processing Systems*, 1276–1284.
- Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., & Alon, U. (2002). Network motifs: Simple building blocks of complex networks. *Science*, 298(5594), 824–827.
- Minhas, S., Hoff, P. D., & Ward, M. D. (2018). Inferential approaches for network analysis: Amen for latent factor models. *Political Analysis*, 27(2), 208–222.
- Molloy, M., & Reed, B. (1995). A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6(2-3), 161–180.
- Molloy, M., & Reed, B. (1998). The size of the giant component of a random graph with a given degree sequence. *Combinatorics, Probability and Computing*, 7(3), 295–305.
- Muirhead, R. J. (1978). Latent roots of matrix variates: A review of some asymptotic results. *The Annals of Statistics*, 6(1), 5–33.
- Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. I. Jones, & X.-L. Meng (Eds.), *Handbook of Markov chain Monte Carlo* (pp. 113–162). CRC Press.
- Newman, M. E. J. (2002). Assortative mixing in networks. *Physical Review Letters*, 89(20), 208701.
- Newman, M. E. J. (2010). *Networks: An introduction*. Oxford University Press.
- Nowicki, K., & Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455), 1077–1087.
- Ohn, I., & Kim, Y. (in press). Posterior consistency for factor dimensionality in high-dimensional sparse factor models. *Bayesian Analysis*, 1–24.

- Paez, M. S., Amini, A. A., & Lin, L. (2019). Hierarchical stochastic block model for community detection in multiplex networks. *arXiv preprint arXiv:1904.05330*.
- Papaspiliopoulos, O., Roberts, G. O., & Sköld, M. (2007). A general framework for the parametrization of hierarchical models. *Statistical Science*, 22(1), 59–73.
- Papastamoulis, P., & Iliopoulos, G. (2010). An artificial allocations based solution to the label switching problem in bayesian analysis of mixtures of distributions. *Journal of Computational and Graphical Statistics*, 19(2), 313–331.
- Pastor-Satorras, R., & Vespignani, A. (2002). Epidemic dynamics in finite size scale-free networks. *Physical Review E*, 65(3), 035108(R).
- Phan, D., Pradhan, N., & Jankowiak, M. (2019). Composable effects for flexible and accelerated probabilistic programming in numpyro. *arXiv preprint arXiv:1912.11554*.
- Polson, N. G., Scott, J. G., & Windle, J. (2013). Bayesian inference of logistic models using Pólya-gamma latent variables. *Journal of the American Statistical Association*, 108(504), 1339–13349.
- Prasad, A. M., Iverson, L. R., & Liaw, A. (2006). Newer classification and regression tree techniques: Bagging and random forests for ecological prediction. *Ecosystems*, 9, 181–199.
- Qi, Y., & Jaakkola, T. S. (2006). Parameter expanded variational Bayesian methods. *Advances in neural information processing systems* (pp. 1097–1104).
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257–286.
- Raftery, A. E., Niu, X., Hoff, P. D., & Yeung, K. Y. (2012). Fast inference for the latent space network model using a case-control approximate likelihood. *Journal of Computational and Graphical Statistics*, 21(4), 901–919.
- Rainforth, T., & Wood, F. (2015). Canonical correlation forests. *arXiv preprint arXiv:1507.05444*.
- Rastelli, R., Friel, N., & Raftery, A. E. (2016). Properties of latent variable network models. *Network Science*, 4(4), 407–432.
- Robert, C. P., Ryden, T., & Titterton, D. M. (2000). Bayesian inference in hidden Markov models through the reversible jump Markov chain Monte Carlo method. *Journal of the Royal Statistical Society - Series B*, 62(1), 57–75.
- Robins, G., Pattison, P., Kalish, Y., & Lusher, D. (2007). An introduction to exponential random graph (p*) models for social networks. *Social Networks*, 29, 173–191.
- Ročková, V., & George, E. I. (2016). Fast Bayesian factor analysis via automatic rotations to sparsity. *Journal of the American Statistical Association*, 111(516), 1608–1622.
- Rubin-Delanchy, P., Cape, J., Tang, M., & Priebe, C. (2017). A statistical interpretation of spectral embedding: The generalised random dot product graph. *arXiv preprint arXiv:1709.05506*.
- Ruppert, D., & Wand, M. P. (1994). Multivariate locally weighted least squares regression. *The Annals of Statistics*, 22(3), 1346–1370.
- Ryan, C., Wyse, J., & Friel, N. (2017). Bayesian model selection for the latent position cluster model for social networks. *Network Science*, 5(1), 70–91.
- Salter-Townshend, M., & McCormick, T. H. (2017). Latent space models for multiview network data. *Annals of Applied Statistics*, 11(3), 1217–1244.
- Salter-Townshend, M., & Murphy, T. B. (2013). Variational Bayesian inference for the latent position clustering model for network data. *Computational Statistics and Data Analysis*, 57(1), 661–671.
- Sampson, S. (1969). *Crisis in a cloister* (Doctoral dissertation). Cornell University.

- Sarkar, P., & Moore, A. W. (2005). Dynamic social network analysis using latent space models. *SIGKDD Explorations*, 7(2), 31–40.
- Scornet, E. (2016). Random forests and kernel methods. *IEEE Transactions on Information Theory*, 62(3), 1485–1500.
- Scott, S. L. (2002). Methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 97(457), 337–351.
- Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4(2), 639–650.
- Sewell, D. K., & Chen, Y. (2015a). Analysis of the formation of the structure of social networks by using latent space models for ranked dynamic networks. *Journal of the Royal Statistical Society, Series C*, 64(4), 611–633.
- Sewell, D. K., & Chen, Y. (2015b). Latent space models for dynamic networks. *Journal of the American Statistical Association*, 110(512), 1646–1657.
- Sewell, D. K., & Chen, Y. (2016). Latent space models for dynamic networks with weighted edges. *Social Networks*, 44, 105–116.
- Sewell, D. K., & Chen, Y. (2017). Latent space approaches to community detection in dynamic networks. *Bayesian Analysis*, 12(2), 351–377.
- Sewell, D. K., Chen, Y., Bernhard, W. T., & Sulkin, T. E. (2016). Model-based longitudinal clustering with varying cluster assignments. *Statistica Sinica*, 26(1), 205–233.
- Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. Chapman; Hall/CRC.
- Small, M., & Singer, D. J. (1969). Formal alliances, 1816 - 1966: An extension of the basic data. *Journal of Peace Research*, 6(3), 257–282.
- Snijders, T. A. B., Pattison, P. E., Robins, G. L., & Handcock, M. S. (2006). New specifications for exponential random graph models. *Sociological Methodology*, 36(1), 99–153.
- Snijders, T. A., Lomi, A., & Tori , V. J. (2013). A model for the multiplex dynamics of two-mode and one-mode networks, with an application to employment preference, friendship, and advice. *Social Networks*, 35(2), 265–276.
- Sosa, J., & Buitrago, L. (2021). A review of latent space models for social networks. *Revista Colombiana de Estadística*, 44(1), 171–200.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & Linde, A. V. D. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal statistical Society Series B*, 64(4), 583–639.
- Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J.-F., Quaghiotto, M., den Broeck, W. V., Régis, C., Lina, B., & Vanhems, P. (2011). High-resolution measurements of face-to-face contact patterns in primary school. *PLoS ONE*, 6(8), e23176.
- Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics*, 9(307).
- Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476), 1566–1581.
- van Dyk, D. A., & Meng, X.-L. (2001). The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1), 1–50.
- Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2013). OpenML: Networked science in machine learning. *SIGKDD Explorations*, 15(2), 49–60.
- Wade, S., & Ghahramani, Z. (2018). Bayesian cluster analysis: Point estimation and credible balls (with discussion). *Bayesian Analysis*, 13(2), 559–626.

- Wainwright, M. J., & Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2), 1–305.
- Wallinga, J., Teunis, P., & Kretzschmar, M. (2006). Using data on social contacts to estimate age-specific transmission parameters for respiratory-spread infectious agents. *American Journal of Epidemiology*, 164(10), 936–944.
- Walter, S. F., Lehmann, L., & Lamour, R. (2012). On evaluating higher-order derivatives of the QR decomposition of tall matrices with full column rank in forward and reverse mode algorithmic differentiation. *Optimization Methods & Software*, 27(2), 391–403.
- Wang, L., Zhang, Z., & Dunson, D. (2019). Common and individual structure of brain networks. *Annals of Applied Statistics*, 13(1), 85–112.
- Wang, Q. (2009). *Sufficient dimension reduction and sufficient variable selection* (Doctoral dissertation). The University of Georgia.
- Wasserman, S., & Pattison, P. (1996). Logit models and logistic regressions for social networks: I. An introduction to Markov graphs and p^* . *Psychometrika*, 61(3), 401–425.
- Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11, 867–897.
- Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, 393, 440–442.
- Xia, Y., Tong, H., Li, W. K., & Zhu, L.-X. (2002). An adaptive estimation of dimension reduction space. *Journal of the Royal Statistical Society, Series B*, 64(3), 363–410.
- Young, S., & Scheinerman, E. (2007). Random dot product graph models for social networks. *Proceedings of the 5th International Conference on Algorithms and Models for the Web-Graph*, 138–149.
- Yuan, X.-T., & Zhang, T. (2013). Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14, 899–925.
- Zagheni, E., Billari, F. C., Manfredi, P., Melegaro, A., Mossong, J., & Edmunds, W. J. (2008). Using time-use data to parameterize models for the spread of close-contact infectious diseases. *American Journal of Epidemiology*, 168(9), 1082–1090.
- Zhang, X., Xue, S., & Zhu, J. (2020). A flexible latent space model for multilayer networks. *Proceedings of the International Conference on Machine Learning*, 8546–8555.
- Zheng, Y., Yi, X., Li, M., Li, R., Shan, Z., Chang, E., & Li, T. (2015). Forecasting fine-grained air quality based on big data. *Proceedings of the 21th SIGKDD conference on Knowledge Discovery and Data Mining*. <https://www.microsoft.com/en-us/research/publication/forecasting-fine-grained-air-quality-based-on-big-data/>
- Zhu, L.-X., & Fang, K.-T. (1996). Asymptotics for kernel estimate of sliced inverse regression. *The Annals of Statistics*, 24(3), 1053–1068.
- Zhu, L.-X., & Ng, K. W. (1995). Asymptotics of sliced inverse regression. *Statistica Sinica*, 5(2), 727–736.