

UCLA

UCLA Electronic Theses and Dissertations

Title

Inference and Size Localization of Mesoscale Structures in Temporal Networks

Permalink

<https://escholarship.org/uc/item/8q68n9v4>

Author

Faust, Theodore Yushin

Publication Date

2025

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Inference and Size Localization
of Mesoscale Structures
in Temporal Networks

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mathematics

by

Theodore Yushin Faust

2025

© Copyright by
Theodore Yushin Faust
2025

ABSTRACT OF THE DISSERTATION

Inference and Size Localization
of Mesoscale Structures
in Temporal Networks

by

Theodore Yushin Faust
Doctor of Philosophy in Mathematics
University of California, Los Angeles, 2025
Professor Mason A. Porter, Chair

In studies of networks, researchers often examine the evolution of mesoscale structures, structures that involve groups of nodes that are larger than a single node but smaller than an overall network. A prominent approach to studying such structures is statistical inference. In the present thesis, we use statistical-inference methods to detect two such mesoscale structures, community structure and core-periphery structure, in time-dependent networks (i.e., “temporal networks”). We represent temporal networks as multilayer networks, with each layer encoding a time step, and we devise statistical-inference methods that avoid common biases in such methods against generating communities or other groups with large or small numbers of nodes. We show that our methods are able to accurately identify mesoscale structure in cases of interest. Additionally, we show that using our generative model

is beneficial for analyzing the community structure of networks with large or small communities. It leads to better accuracy than methods that contain biases against generating groups with large or small numbers of nodes. We also generalize hierarchical core–periphery structure, which is a type of core–periphery structure in which nodes can be members of multiple groups simultaneously, to temporal networks. We use a statistical-inference approach to identify such core–periphery structure in real-world temporal networks.

The dissertation of Theodore Yushin Faust is approved.

Hayden Schaeffer

Deanna Needell

Christopher R. Anderson

Arash A. Amini

Mason A. Porter, Committee Chair

University of California, Los Angeles

2025

*To my family, for whose
love and support I am truly grateful.*

TABLE OF CONTENTS

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Organization | 2 |
| 2 | Background | 4 |
| 2.1 | Network Science and Mesoscale Structures | 4 |
| 2.2 | Temporal and Multilayer Networks | 5 |
| 2.3 | Statistical Inference on Networks | 6 |
| 3 | Notation | 9 |
| 4 | Detection of Community Structure in Temporal Networks | 11 |
| 4.1 | Introduction | 11 |
| 4.2 | Statistical-Inference Methods for Community Detection | 14 |
| 4.2.1 | Statistical Inference | 14 |
| 4.2.2 | Community-Assignment Probability Distributions for Mono- layer Networks | 17 |
| 4.2.3 | Community-Assignment Probability Distributions for Tempo- ral Networks | 20 |
| 4.3 | Localization of Community-Size Distributions | 28 |
| 4.3.1 | Methodology | 29 |

| | | |
|----------|---|-----------|
| 4.3.2 | Community-Size Distributions of Generative Models of Mono-layer Networks | 32 |
| 4.3.3 | Community-Size Distributions of Generative Models of Temporal Networks | 36 |
| 4.3.4 | Asymptotic Behavior of Our Exchangeability-Based Approach | 43 |
| 4.3.5 | Summary | 47 |
| 4.4 | Comparison of Community-Detection Performance of Statistical-Inference Approaches | 47 |
| 4.4.1 | Posterior-Sampling Approaches | 48 |
| 4.4.2 | Setup of our Comparison | 57 |
| 4.4.3 | Results | 59 |
| 4.5 | Conclusions and Discussion | 64 |
| | | |
| 5 | Detection of Hierarchical Core–Periphery Structure in Temporal Networks | 66 |
| 5.1 | Introduction | 66 |
| 5.2 | A Hierarchical Generative Model for Core–Periphery Structure in Temporal Networks | 69 |
| 5.2.1 | Fixed Number of Groups | 70 |
| 5.2.2 | Main Model: Unspecified Number of Groups | 73 |
| 5.3 | Statistical-Inference Approach | 74 |
| 5.3.1 | Our Main MCMC Algorithm | 74 |
| 5.3.2 | Acceptance Probability | 79 |

| | | |
|----------|---|------------|
| 5.3.3 | Statement of our Main MCMC Algorithm | 80 |
| 5.3.4 | MCMC Algorithm for a Fixed Number of Groups | 81 |
| 5.3.5 | Intermediate Algorithm for a Variable Number of Groups | 86 |
| 5.3.6 | Revisiting our Main MCMC Algorithm | 89 |
| 5.3.7 | Computation of Acceptance Probability | 92 |
| 5.4 | Discussion of our Group-Assignment Approach | 94 |
| 5.5 | Application to Real-World Networks | 95 |
| 5.5.1 | Jammu–Kashmir Terrorist Network | 96 |
| 5.5.2 | Literary Co-Appearance Network | 98 |
| 5.6 | Conclusions and Discussion | 105 |
| 6 | Conclusion | 107 |
| 6.1 | Summary | 107 |
| 6.1.1 | Future Directions | 108 |
| A | Table of Commonly-Used Notation | 110 |
| B | Expressions for the Community-Assignment Probabilities | 112 |
| B.1 | Closed-Form Expression for $\mathbb{P}(g)$ for the Bazzi et al. Approach | 112 |
| B.2 | Closed-Form Expression for $\mathbb{P}(g)$ for our Exchangeability-Based Approach | 115 |
| B.2.1 | Computing $J(k_1, k_2)$ | 118 |
| | References | 119 |

LIST OF FIGURES

| | | |
|-----|---|----|
| 4.1 | An example of the community-size frequency histogram for a monolayer network. | 30 |
| 4.2 | An example of the community-size histograms for a temporal network with 5 layers. | 31 |
| 4.3 | Community-size histograms for monolayer networks for $k = 2$ communities for a uniform distribution on community assignments and nodewise community assignments. | 33 |
| 4.4 | Community-size histograms for monolayer networks for $k = 5$ communities for a uniform distribution on community assignments and nodewise community assignments. | 34 |
| 4.5 | Community-size histograms for each layer of a temporal network for $k = 2$ communities for a uniform distribution on community assignments, the Yang et al. approach [88], the Bazzi et al. approach [1], and our exchangeability-based approach. | 37 |
| 4.6 | Community-size histograms for the overall network for $k = 2$ communities for a uniform distribution on community assignments, the Yang et al. approach, the Bazzi et al. approach, and our exchangeability-based approach. | 38 |
| 4.7 | Community-size histograms for each layer of a temporal network for $k = 5$ communities for a uniform distribution on community assignments, the Yang et al. approach, the Bazzi et al. approach, and our exchangeability-based approach. | 39 |

| | | |
|------|---|----|
| 4.8 | Community-size histograms for the overall network for $k = 5$ communities for a uniform distribution on community assignments, the Yang et al. approach, the Bazzi et al. approach, and our exchangeability-based approach. | 40 |
| 4.9 | A comparison of the community-size histogram for layer 50 of a temporal network with $k = 2$ groups for a community-assignment probability distribution that samples using the procedure (4.22) and the conjectured distribution from (4.23). | 46 |
| 4.10 | Heat maps of an example of actual community structure and an illustration of the permuted community structure commonly observed in local maxima of $\mathbb{P}(g A)$ in a 100-node network with 5 layers. | 51 |
| 4.11 | Heat maps of the seeded community structure that we use to generate the adjacency structure A for community-1 size $q = 50$ and community-1 size $q = 90$ | 58 |
| 4.12 | The mean NMI for our exchangeability-based approach and the Bazzi et al. approach with and without multi-node moves for several values of the community-1 size q | 61 |
| 4.13 | The mean NMI for each of the four examined approaches for generating community assignments for various choices of 1-community sizes q | 63 |
| 5.1 | Heat maps of an example of actual core–periphery structure and an illustration of the permuted group structure commonly observed in local maxima of $\mathbb{P}(g A)$ in a 100-node network with 5 layers. | 90 |

| | | |
|-----|---|-----|
| 5.2 | The inferred core–periphery structure in the Jammu–Kashmir terrorist network. | 97 |
| 5.3 | Permuted adjacency matrices to illustrate the inferred core–periphery structure in the Jammu–Kashmir terrorist network. | 98 |
| 5.4 | The inferred core–periphery structure from one run of Algorithm 4 for the <i>Luke Gospel</i> literary co-appearance network. | 100 |
| 5.5 | The inferred core–periphery structure from one run of Algorithm 4 for the <i>Luke Gospel</i> literary co-appearance network. | 102 |
| 5.6 | Permuted adjacency matrices to illustrate the inferred core–periphery structure in the <i>Luke Gospel</i> literary co-appearance network. | 104 |

LIST OF TABLES

| | | |
|-----|--|-----|
| 4.1 | Values of the inverse participation ratio (IPR) for single-layer community-size distributions. | 35 |
| 4.2 | IPR values of the community-size distribution of each layer for each examined generative model for community assignments in temporal networks. | 41 |
| 4.3 | IPR values of the community-size distribution of the overall network for each examined generative model of community assignments in temporal networks. | 42 |
| 4.4 | The p -values for our comparison of our exchangeability-based approach with and without multi-node moves. | 61 |
| 4.5 | The p -values for our comparison of the Bazzi et al. [1] approach with and without multi-node moves. | 62 |
| A.1 | Table of commonly-used notation. | 110 |

ACKNOWLEDGMENTS

As a student, I have been fortunate to have been taught, mentored, and assisted by many people. Undoubtedly, I would not be where I am today, as either a scholar or a person, without them. In this section, I want to express my gratitude for these people.

I want to begin by thanking my advisor, Mason A. Porter. Mason has been a fantastic advisor in so many aspects, from his many pieces of advice that helped shape the direction of our research to his detailed and thorough comments on my drafts, which greatly helped me improve the clarity of my mathematical writing. I am truly grateful to have been his student.

I also want to thank Arash A. Amini, one of my committee members. Arash played a key role in the work in Chapter 4. In particular, he helped troubleshoot the lack of convergence of our Gibbs-sampling approach, he stated and sketched a proof of an analytical result about the limit behavior of community-size distributions (see Conjecture 1), and he provided many useful comments and suggestions about the notation and text in Chapter 4. Additionally, I want to thank the other members of my committee — Christopher R. Anderson, Deanna Needell, and Hayden Schaeffer — for their helpful questions and comments on my thesis and at my presentations.

I want to thank the NSF for awarding me the NSF Graduate Research Fellowship (GRFP). The material in this thesis is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-2034835. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

I want to thank my co-authors, without whom our papers would not have been possible. Specifically, Chapter 4 is based on [18], which I co-authored with Mason A. Porter and Arash A. Amini, and Chapter 5 is based on [19], which I co-authored with Mason A. Porter. I also want to thank Grace Li for helpful comments on Chapter 4 and Arash A. Amini for helpful comments on Chapter 5.

I want to thank the members of Mason Porter's research group for their help and advice. Specifically, I want to thank Michael Johnson, Leah Keating, Grace Li, Jerry Luo, Kaiyan Peng, and Sarah Tymochko; my time at UCLA would undoubtedly have been much less enjoyable and much more difficult without them.

I want to thank my former professors and mentors for preparing me for my Ph.D. both as a mathematician and as a person. In particular, I want to thank Mark Iwen, Christopher Manon, Robert Sachs, Timothy Sauer, Tsvetanka Sendova, and Jeanne Wald. I am truly grateful for their teaching and mentoring. I cannot overstate how much of an impact they have had on me; without them, I would not be where I am today.

Last, but definitely not least, I want to thank my family for their love and support. To Dad and Mom, I cannot possibly express in words how much you mean to me. Whether it is teaching me how to read a fly ball, finding math circles and camps for me, giving me life advice, or helping me in my most difficult moments, I am so grateful for the time we spend together and for everything you have done for me. To Nathan, thank you for being the best brother anyone could ever ask for. It means so much to me to know that I can always turn to you and that you will always have my back.

VITA

- 2016–2020 B.S. (Mathematics, Advanced), Michigan State University.
- 2024 Research in Industrial Projects for Students (RIPS) Academic Mentor

PUBLICATIONS

Theodore Y. Faust and Christopher Manon. The Gorenstein property for projective coordinate rings of the moduli of parabolic SL_2 -principal bundles on a smooth curve. *The Electronic Journal of Combinatorics*, 26(4):P4.25, 2019

Theodore Faust, Mark A. Iwen, Rayan Saab, and Rongrong Wang. On the ℓ^∞ -norms of the singular vectors of arbitrary powers of a difference matrix with applications to sigma-delta quantization.” *Linear Algebra and its Applications*, 626:79–151, 2021

Theodore Y. Faust, Arash A. Amini, and Mason A. Porter. Community-size biases in statistical inference in temporal networks. In preparation.

Theodore Y. Faust and Mason A. Porter. A statistical-inference method for identifying hierarchical core–periphery structure in temporal networks. In preparation.

CHAPTER 1

Introduction

In this thesis, we develop statistical-inference methods to identify mesoscale structures in temporal networks.

The study of networks, which consist of nodes (representing agents or other entities) and edges between nodes (which encode interactions or relationships between objects), has become commonplace [56]. A common research topic in network science is the study of “mesoscale structures”, which involve groups of nodes that are larger than a single node but smaller than an overall network [56]. For example, it is very common when analyzing networks to seek to identify communities — which are sets of nodes that are densely connected to each other but sparsely connected to other sets of nodes — in networks [20, 56, 69]. The study of mesoscale structures has led to a greater understanding of a variety of complex systems in many fields, such as sociology [26, 60, 83, 84], biology [10], and economics [8, 76, 85].

In many applications — such as in the study of social networks [60, 83], economic networks [8], citation networks [38], and biological networks [10] — it is important to consider networks whose structures can change with time. One can represent such data as a temporal network, in which the nodes and/or the edges between them are time-dependent [30–32]. A temporal network is a sequence of networks in which each network encodes the relationships between entities during one time point or

time period.

The study of mesoscale structures in temporal networks has led to insights into a wealth of applications, including causal inference in social networks [59], identification of epidemic sources [39], role detection in bicycle-sharing networks [9], and analysis of migration networks [13]. Our work focuses on the use of statistical-inference methods [23, 67, 79, 88, 89] to identify mesoscale structures in temporal networks. Such methods offer many benefits, including convergence guarantees and mitigation of overfitting issues [66]. In this thesis, we devise a statistical-inference method that identifies community structure in temporal networks and avoids certain assumptions that negatively impact other statistical-inference approaches. We show that, in cases of interest, our method outperforms existing methods that make such assumptions. We also devise a novel approach for the identification of core–periphery structure in temporal networks.

1.1 Organization

The thesis proceeds as follows.

In Chapter 2, we give background information on mesoscale structures, temporal networks, and statistical-inference methods on networks.

In Chapter 3, we introduce the notation for temporal networks that we use throughout the thesis.

In Chapter 4, we discuss our community-detection method for temporal networks. We show that our method avoids the common biases in such methods against generating communities with large or small numbers of nodes. We also show that our

method is beneficial for analyzing the community structure of networks with such communities, as it leads to better accuracy than methods that include these biases.

In Chapter 5, we discuss our hierarchical core–periphery identification method for temporal networks. We prove that our Markov-chain Monte Carlo (MCMC) method samples from the exact posterior distribution in the limit as the number of nodes of a network goes to infinity. We then use our method to study two real-world temporal networks and demonstrate that it can identify time-dependent hierarchical core–periphery structure in those data sets.

Finally, in Chapter 6, we present our overall conclusions and discuss future directions.

In Appendix A, we provide a table of commonly-used notation. In Appendix B, we derive computable expressions for various probability distributions. We need these expressions to implement the examined statistical-inference approaches.

CHAPTER 2

Background

2.1 Network Science and Mesoscale Structures

The use of networks to encode the behavior of complex systems, which involve relational and interactional behaviors, has become commonplace [56]. A network consists of nodes and edges. Nodes represent entities or other objects in a system, while edges between two nodes encode pairwise ties or interactions between the objects. As an example, consider a social network in which the edges represent people and two nodes are connected by an edge if they are friends with each other. The notion of a network is very versatile, and the study of networks has led to advances in a wide variety of fields, including sociology [80], biology [64], epidemiology [11], and other areas.

In network science, it is common to study “mesoscale structures”, which involve groups of nodes that are larger than a single node but smaller than an overall network [56]. Two common mesoscale structures, which we consider in this thesis, are community structure [56] and core–periphery structure [71].

When studying community structure, one supposes that a network consists of “communities” of nodes that are connected densely to each other but connected sparsely to other sets of nodes [20, 56, 69]. Communities in real-world networks can represent important structures in their associated systems. For example, in a

social network, the identification of communities allows one to algorithmically detect groups of friends [3]. Investigations of community structure have led to insights into the study of social networks [60, 83], economic networks [8], citation networks [38], biological networks [10], and other applications.

When detecting core–periphery structure, one supposes that a network consists of a set of densely-connected core nodes and sparsely-connected peripheral nodes. Core nodes can be either densely connected to peripheral nodes or sparsely connected to them [71]. Similarly to community structure, core–periphery structure can represent important structures in a complex system [7, 71]. For example, in a trade network in which nodes represent countries and edges encode trade flows, the identification of core–periphery structure allows one to differentiate countries’ involvement in world trade systems [5, 42]. Studies of core–periphery structure have also led to insights into the study of social networks [26, 84], academic networks [16, 86], economic networks [76, 85], and many other systems.

2.2 Temporal and Multilayer Networks

In many situations — including in the analysis of the spread of diseases through face-to-face contacts [21], transportation systems [53], legislation cosponsorships [46, 55], and other applications — it is important to consider relationships and/or interactions that change with time. One can represent such data as a temporal network, in which the entities and/or the ties between them can change with time [30–32]. A temporal network is a sequence of networks in which each network encodes the relationships between entities during one time point or time period.

It is also common in network science to consider multilayer networks, which con-

sist of layers (which are each networks) along with edges that connect nodes in different layers [40]. Multilayer networks are able to encode a variety of structures. Notably for our work, one can use multilayer networks to encode temporal networks by considering the network structure at each time point to be a layer [40]. Throughout this thesis, we use this multilayer formulation of temporal networks, as it is convenient and it allows the possibility of generalizing our work to broader varieties of multilayer networks.

2.3 Statistical Inference on Networks

In this thesis, we are concerned with the identification of community structure and core–periphery structure in temporal networks. To identify such structures, we use a statistical-inference approach [66, 67, 79, 88]. In an inferential approach, one uses a generative model to algorithmically detect a desired type of structure. We employ a Bayesian approach to sample from a posterior distribution and obtain a network’s community structure [65]. There are statistical-inference approaches to detect many types of mesoscale structures in networks [68, 81, 89], and there are a particularly large number of such methods for community structure [20, 66]. Although statistical-inference approaches have a tendency to be somewhat computationally costly, they also have several favorable properties, such as results that guarantee convergence and a better ability to avoid overfitting than other approaches [66].

In statistical inference, it is desirable to employ generative models that are based on realistic assumptions [66]. A generative model that relies on unrealistic assumptions can have a detrimental impact on the accurate detection of community structure, core–periphery structure, and any other structure in networks. However, it

is not always straightforward to avoid unrealistic assumptions, and many generative models that appear to make reasonable choices include such assumptions. One example of a problematic assumption that is present in many generative models for community structure involves the probability distribution of the number of nodes in a community. We highlight an example in the context of temporal networks. It is common for generative models of temporal community structure to use a discrete-time Markov process to assign nodes to communities [1, 25, 50, 88]. However, due to the choices that are made in many such models, the community-size distributions become increasingly localized over time. Therefore, at later times of a temporal network, the probability of generating a small or large community is much smaller than the probability of generating a community of moderate size. Real-world temporal networks can have communities of many sizes [61], including ones that are small or large, so it is desirable that generative models of networks are able to successfully infer such communities. To mitigate this issue, in Chapter 4, we introduce a novel community-evolution approach that yields community-size distributions with substantially less localization than in the above models. Our approach thereby leads to more accurate community identification than in these types of approaches in networks with small or large communities.

The importance of avoiding unrealistic assumptions extends beyond community structure. Indeed, in Chapter 5, we demonstrate that generative models that are based on discrete-time Markov processes cause our inference approach to consistently underestimate the number of groups in identification of core–periphery structure. To mitigate this issue, we use a modified version of our community-evolution approach from Chapter 4 to evolve the core–periphery assignments in our generative model. This approach mitigates the underestimation of the number of groups and improves

core-periphery identification in temporal networks.

CHAPTER 3

Notation

In this section, we introduce notation for both single-layer (i.e., “monolayer”) networks and temporal networks, which we represent as multilayer networks in which each layer corresponds to one time step [40]. In Appendix A, we provide a table of the commonly-used notation in the thesis.

We first discuss our notation for monolayer networks. For simplicity, we consider unweighted and undirected networks without self-edges or multi-edges. A monolayer network is a graph $G = (V, E)$, which consists of a set $V = \{1, \dots, n\}$ of nodes (i.e., vertices) and a set $E \subseteq V \times V$ of edges. We denote an undirected edge by (i, j) . We represent a monolayer network G using an adjacency matrix $A \in \{0, 1\}^{n \times n}$, where $A_{ij} = 1$ if nodes i and j are connected directly by an edge (i.e., they are adjacent) and $A_{ij} = 0$ otherwise. We consider only undirected and unweighted graphs.

We represent a temporal network as a multilayer network in which each layer encodes the adjacencies between nodes at its associated time step. We model a temporal network as a sequence of network layers (i.e., times) $\ell \in \{1, \dots, L\}$. At each time $\ell \in \{1, \dots, L\}$, we suppose that all nodes $i \in \{1, \dots, n\}$ are present. We refer to an instantiation of a node in a given layer as a node-layer $(i, \ell) \in \{1, \dots, n\} \times \{1, \dots, L\}$. We again use an adjacency representation, so we have a sequence $(A^{(1)}, \dots, A^{(L)})$, with $A^{(\ell)} \in \{0, 1\}^{n \times n}$ for each layer ℓ . We assume that the networks are unweighted

and undirected, so $A_{ij}^{(\ell)} = 1$ if node-layers (i, ℓ) and (j, ℓ) are adjacent and $A_{ij}^{(\ell)} = 0$ if they are not adjacent. For notational convenience, we let A denote the sequence $(A^{(1)}, \dots, A^{(L)})$. Technically, this is an abuse of notation because we already used A to refer to a single adjacency matrix for a monolayer network, but we always clearly state whether we are considering a monolayer network or a multilayer network. As general terminology, we refer to A as an *adjacency structure*. For convenience (and despite the additional associated abuse of notation), we also sometimes refer to A as a “network”.

CHAPTER 4

Detection of Community Structure in Temporal Networks

4.1 Introduction

As we discussed in Section 2.1, in network analysis, it is common to study “communities” of nodes that are connected densely to each other but connected sparsely to sets of nodes [20, 56, 69]. Investigations of community structure in networks have led to insights in the study of social networks [60, 83], economic networks [8], citation networks [38], biological networks [10], and many others.

As we stated in Section 2.2, it is important in many situations to consider relationships and/or interactions that change with time. One can represent such data as a temporal network; in such a network, the entities and/or the ties between them can change with time [30–32]. A temporal network is a sequence of networks in which each network encodes the relationships between entities at one time point or time period. A variety of approaches have been developed to algorithmically detect communities in temporal networks [72]. These approaches include statistical inference [67, 79, 88], optimization of various objective functions [2, 51, 54], non-negative matrix and tensor factorization [17, 24, 74], information-theoretic methods (such as

those that minimize description length) [67, 75, 78], local methods [34, 37, 47], and other applications.

In the present chapter, we study community detection in temporal networks using statistical inference [66]. As we discussed in Section 2.3, in an inferential approach, one uses a generative model to algorithmically detect a desired type of network structure. There are statistical-inference approaches to detect many types of mesoscale structures in networks [68, 81, 89], and there are a particularly large number of such methods for community structure [20, 66]. Investigations of community structure using statistical inference have led to insights into a wealth of applications, including causal inference in social networks [59], identification of epidemic sources [39], role detection in bicycle-sharing networks [9], and international migration [13]. The statistical-inference methods that we consider use a Bayesian approach to sample from a posterior distribution and obtain a network’s community structure [65].

As we discussed in Section 2.3, it is desirable to base the generative models that one employs to statistically infer community structure on realistic assumptions [66]. A generative model that relies on unrealistic assumptions can have a detrimental impact on the accurate detection of communities in networks. However, it is not always straightforward to avoid unrealistic assumptions, and many generative models that appear to make reasonable choices include such assumptions. One example of a problematic assumption involves the probability distribution of the number of nodes in a community. It is common for generative models for temporal community structure to use a discrete-time Markov process to assign communities [1, 25, 50, 88]. However, due to the choices in many such models, the community-size distributions become increasingly localized over time. To demonstrate this behavior, we consider the Markov-process models of Yang et al. [88] and Bazzi et al. [1]. We show that

the community-size distributions for these models become increasingly localized over time. Therefore, at later times of a temporal network, the probability of generating a small or large community is much smaller than the probability of generating a community of moderate size. Real-world temporal networks can have communities of many sizes [61], including ones that are small or large, so it is desirable that generative models of networks are able to successfully infer small and large communities.

To mitigate the issue of generative models producing networks that are biased against small and large communities, we introduce a novel community-evolution approach that yields community-size distributions with substantially less localization than those from Markov-process models. Our approach generates the community assignments of a network at a given time using the community assignments of all nodes at the previous time, rather than updating each node separately at each time. For a multilayer representation of a temporal network [40], when evolving the community assignments from one layer to the next (i.e., from one time to the next), we generate the number of nodes with community assignments that change between layers, instead of generating the community assignments of each node separately. This choice arises from the idea of “exchangeability” [6]. In our context, it signifies that we should not distinguish between nodes with the same community assignment. We demonstrate that statistical-inference methods that use our generative model perform more accurately than the Yang et al. [88] and Bazzi et al. [1] Markov-process models in networks with small or large communities.

This chapter proceeds as follows. In Section 4.2, we introduce the Markov-process methods and our statistical-inference approach in more detail. In Section 4.3, we demonstrate empirically that our approach leads to significantly less-localized community-size distributions than the Markov-process models. In Section 4.4, we

demonstrate that our approach identifies communities more accurately than those models in synthetic networks with small or large communities. In Section 4.5, we conclude and discuss future directions. In Appendix B, we derive computable expressions for various probability distributions. We need these expressions to implement the examined statistical-inference approaches.

4.2 Statistical-Inference Methods for Community Detection

In this section, we discuss statistical-inference methods for community detection. In Section 4.2.1, we discuss how the methods that we consider use a generative model to infer community structure in networks. In Sections 4.2.2 and 4.2.3, we introduce the generative models that we will consider, with a particular focus on each method’s probability distribution for community assignments.

4.2.1 Statistical Inference

In a statistical-inference method for community detection, one chooses a *generative model* for networks with community structure [66]. To do this, one creates a generates random adjacency structure A , which is an adjacency matrix for a monolayer network and is a sequence of adjacency matrices for a temporal network, according to some probability distribution $\mathbb{P}(A)$. We choose the probability distribution $\mathbb{P}(A)$ so that the randomly-generated networks have community structure. Let k denote the number of communities in a network. Given k , a generative model first generates a vector $g \in \{1, \dots, k\}^n$ (if we seek a monolayer network) or a matrix $g \in \{1, \dots, k\}^{n \times L}$ (if we seek a temporal network) of community assignments. Let $g_i \in \{1, \dots, k\}$ denote the community assignment of node i of a monolayer network,

and let $g_{(i,\ell)} \in \{1, \dots, k\}$ denote the community assignment of node-layer (i, ℓ) of a temporal network. The vector

$$g^{(\ell)} = (g_{(1,\ell)}, \dots, g_{(n,\ell)}) \quad (4.1)$$

encodes the community assignment of each node-layer in layer ℓ of a temporal network.

Using the generative model, we generate an adjacency structure A with an “expected” community structure g . We expect nodes (respectively, node-layers) with the same community assignment to be more likely to be adjacent to each other than to nodes (respectively, node-layers) with different community assignments. One commonly-employed generative model to generate a monolayer network A given a community assignment g is a *stochastic block model* (SBM) [56, 65]. In the simplest type of SBM, given a matrix $\psi \in [0, 1]^{k \times k}$, we place an edge between nodes i and j independently with probability $\psi_{g_i g_j}$ for each pair $\{i, j\}$ of distinct nodes. If the diagonal elements ψ_{rr} are larger than the off-diagonal elements ψ_{rs} (with $r \neq s$), we expect that nodes with the same community assignments are connected more densely than nodes with different community assignments. In other words, we expect that A has community structure that is specified by g .

To make our above intuition precise, let $\mathbb{P}(A|g)$ denote the probability distribution that we obtain a network (i.e., an adjacency structure) A given the community structure g . Let $\mathbb{P}(g)$ denote the probability distribution of the community assignments. This distribution describes the probability of a given community structure g independent of the observed adjacency structure A . The primary foci of the present chapter are the assumptions about this distribution and the effect of these assumptions on the performance of statistical-inference methods for community detection.

In particular, we focus on assumptions about the probability distribution of the number of nodes in a community (i.e., the size of a community). Such size assumptions also affect the behavior and performance of statistical-inference methods that infer a discrete group assignment for each node-layer. This includes the identification of both community structure and other mesoscale structures (e.g., core–periphery structure) in temporal networks. In Chapter 5, we discuss the effects of similar size assumptions on the behavior of statistical-inference algorithms for core–periphery detection.

The generative model for a random network A with community structure g is

$$\mathbb{P}(A, g) = \mathbb{P}(A|g)\mathbb{P}(g). \quad (4.2)$$

Because we fix the number k of communities, one can think of k as an input of the examined methods. We assume a fixed k throughout the entire statistical-inference process, so we omit k from the notation for each probability distribution.

We consider statistical-inference methods that use a Bayesian approach to infer the community structure g of a network A (which is either monolayer or temporal). Given $\mathbb{P}(A|g)$ and $\mathbb{P}(g)$, the posterior distribution for the inferred community structure g given the observed network A is

$$\mathbb{P}(g|A) = \frac{\mathbb{P}(A|g)\mathbb{P}(g)}{\mathbb{P}(A)}. \quad (4.3)$$

There are a variety of approaches to sample from $\mathbb{P}(g|A)$ [22, 88]. In Section 4.4, we briefly discuss the specific sampling approaches that we use in our comparisons of various statistical-inference methods. However, our focus is the effect of the choice of generative model on the results of statistical inference, so we do not include a detailed discussion of the benefits and drawbacks of different sampling approaches.

For comparisons of such sampling approaches, see [63] and [49] for a general comparison and see [22] for a discussion of the application of various methods of statistical inference of community structure.

There are also a variety of generative models $\mathbb{P}(A|g)$ that generate a network A with “expected” community structure g . (See [22] for a review of such models.) However, because our primary concerns are (1) the de facto assumptions about the community assignments that arise from different choices of community-assignment probability distributions $\mathbb{P}(g)$ and (2) the effects of these assumptions on statistical-inference results, we also limit our discussion to the generative models $P(A|g)$ that we use in our comparisons.

4.2.2 Community-Assignment Probability Distributions for Monolayer Networks

As we discussed in Section 4.2.1, our primary goals are to examine (1) the choices of community-assignment probability distributions $\mathbb{P}(g)$ in various generative models of community structure in temporal networks and (2) the influence of these choices on the results of their corresponding statistical-inference approaches for community detection. We now discuss each of the choices of $\mathbb{P}(g)$ in detail. In Sections 4.2.2.1 and 4.2.2.2, we first consider generative models of monolayer networks, as the examined generative models of temporal networks are extensions of these models. In this discussion, we focus on properties of $\mathbb{P}(g)$ that affect the performance of the corresponding statistical-inference methods. In Section 4.3, we support these claims using numerical computations.

4.2.2.1 Uniform Distributions on Community Assignments

The first class of community-assignment probability distributions $\mathbb{P}(g)$ are uniform distributions on community assignments, for which

$$\mathbb{P}(g) = \frac{1}{k^n} \tag{4.4}$$

for all community assignments $g \in \{1, \dots, k\}^n$. Equivalently, such a model generates a community assignment g by generating each node's community assignment g_i independently at random from a uniform distribution on $\{1, \dots, k\}$.

It is common to use a uniform distribution on community assignments (see, e.g., [25,28]). However, Polanco and Newman [68] noted that this distribution choice assumes implicitly that the distributions of the sizes of communities are highly localized (and specifically that the probability of generating a community structure with large or small communities is very small), which typically is an unrealistic assumption in practice.

4.2.2.2 Nodewise Community Assignments

Another type of single-layer community-assignment probability distribution is “node-wise” community assignments [88]. Employing a nodewise approach allows one to sample the community assignments of each node from a distribution that is not uniform. One inputs a parameter vector $\pi \in [0, 1]^k$ with entries that sum to 1. For each node i , one independently sets its community assignment g_i to a value r with probability π_r . We denote this procedure by

$$g_i | \pi \sim \pi. \tag{4.5}$$

Equivalently, the probability distribution of g is

$$\mathbb{P}(g|\pi) = \prod_{r=1}^k \pi_r^{n_r(g)}, \quad (4.6)$$

where $n_r(g)$ is the number of nodes i with $g_i = r$. That is, $n_r(g)$ is the size of community r .

To remove the dependence on π in (4.6), we sample π from the Dirichlet distribution with parameters $\gamma_1, \dots, \gamma_k$. That is, $\pi \sim \text{Dir}(\gamma_1, \dots, \gamma_k)$, so

$$\mathbb{P}(\pi) = \frac{\Gamma(\sum_{r=1}^k \gamma_r)}{\prod_{r=1}^k \Gamma(\gamma_r)} \prod_{r=1}^k \pi_r^{\gamma_r-1}, \quad (4.7)$$

where $\Gamma(\cdot)$ is the Gamma function. In the present thesis, we always choose $(\gamma_1, \dots, \gamma_k) = (1, \dots, 1)$, which entails that the prior distribution for π is a uniform distribution on $\{(v_1, \dots, v_k)^T \mid \sum_{r=1}^k v_r = 1\} \cap [0, 1]^k$. For notational convenience, let Δ^{k-1} denote the set

$$\left\{ (v_1, \dots, v_k)^T \mid \sum_{r=1}^k v_r = 1 \right\} \cap [0, 1]^k. \quad (4.8)$$

The choice $(\gamma_1, \dots, \gamma_k) = (1, \dots, 1)$ in the distribution (4.7) is equivalent to taking a uniform distribution on community sizes [57]. In this uniform distribution, one first chooses the sizes n_1, \dots, n_k of communities $1, \dots, k$ uniformly at random from the set of ordered pairs of k non-negative integers that sum to n . One then chooses the community assignment g uniformly at random from the set of all community assignments with n_r nodes in community r (for $r \in \{1, \dots, k\}$).

Although this class of community-assignment probability distributions is perhaps less intuitive than using uniform distributions on community assignments, we show in Section 4.3.2 that it greatly mitigates localization issues in community-size distributions.

4.2.3 Community-Assignment Probability Distributions for Temporal Networks

We now build on our discussion of choices of community-assignment probability distributions $\mathbb{P}(g)$ in generative models of monolayer networks (see Section 4.2.2) to examine these choices in temporal networks.

In the present discussion of $\mathbb{P}(g)$, we omit closed-form expressions for some choices of $\mathbb{P}(g)$ because deriving them is cumbersome. For those probability distributions, we instead provide a procedure to sample from $\mathbb{P}(g)$. In Appendix B, we include derivations of the closed-form expressions of $\mathbb{P}(g)$ that are required to sample from the posterior distributions $\mathbb{P}(g|A)$. In Section 4.3.1, we discuss these sampling methods in detail.

4.2.3.1 Uniform Distributions on Community Assignments

As with monolayer networks (see Section 4.2.2.1), one can employ a uniform distribution on community assignments in the study of temporal networks. In contrast to the monolayer setting, because a uniform distribution avoids correlation between the community assignments in different layers (which is generally desirable), Researchers avoid this choice for inference applications.

Analogously to (4.4), we have

$$\mathbb{P}(g) = \frac{1}{k^{nL}} \tag{4.9}$$

for all community assignments $g \in \{1, \dots, k\}^{n \times L}$. As with monolayer networks, it is equivalent to generate a community assignment g by generating the community assignment $g_{(i,\ell)}$ of each node-layer independently at random from a uniform distri-

bution on $\{1, \dots, k\}$.

As with monolayer networks, this community-assignment probability distribution implicitly makes the unrealistic assumption that the community-size distributions are highly localized.

4.2.3.2 Discrete-Time Markov-Process Models

Another class of community-assignment probability distributions use a discrete-time Markov process to generate community assignments [1, 25, 50, 88].

One uses a nodewise approach (see Section 4.2.2.2) to generate the community assignments for the first layer of a temporal network. That is, to sample the community assignment $g_{(i,1)}$, one follows the procedure

$$\begin{aligned} \pi &\sim \text{Dir}(\gamma), \\ g_{(i,1)} | \pi &\sim \pi, \end{aligned} \tag{4.10}$$

where we recall that $\gamma = (1, \dots, 1)$. In this thesis, when we provide a sequence of samples from probability distributions (i.e., a “sampling procedure”), such as in (4.10), we perform the sampling in sequence from top to bottom. For the second and subsequent layers, one generates the community assignments of each node-layer using a process that depends only on the community assignment of the same node in the previous layer. To do this, one inputs a “transition kernel” $K = (K^{(2)}, \dots, K^{(L)})$, where each $K^{(\ell)}$ is a matrix whose rows sum to 1, and “laziness parameters” $\alpha_\ell \in [0, 1]$. The ℓ th laziness parameter α_ℓ is the probability that one copies the community assignment from that of the previous layer when generating the community assignment for a node in layer ℓ . For notational convenience, we write $\alpha = (\alpha_2, \dots, \alpha_L)$.

One then generates $g_{(i,\ell)}$ according to the conditional distribution

$$\mathbb{P}(g_{(i,\ell)} = r \mid g_{(i,\ell-1)} = s, \alpha, K) = \alpha_\ell \delta_{rs} + (1 - \alpha_\ell) K_{sr}^{(\ell)}, \quad (4.11)$$

where δ_{ij} is the Kronecker delta. Sampling the community assignments $g_{i,\ell}$ according to the conditional distributions (4.11) is equivalent to using

$$\{g_{(i,\ell)}\}_{\ell=2}^L \mid \alpha, K \sim \text{Markov}(\{\alpha_\ell I + (1 - \alpha_\ell) K^{(\ell)}\}),$$

where $\text{Markov}(\{K^{(\ell)}\})$ denotes a discrete-time Markov process with transition kernels $\{K^{(\ell)}\}$. We thus can write

$$\begin{aligned} \pi &\sim \text{Dir}(\gamma), \\ g_{(i,1)} \mid \pi &\sim \pi, \\ \{g_{(i,\ell)}\}_{\ell=2}^L \mid \alpha, K &\sim \text{Markov}(\{\alpha_\ell I + (1 - \alpha_\ell) K^{(\ell)}\}). \end{aligned} \quad (4.12)$$

To complete the specification of the model, we need to provide priors on the laziness parameters α and the transition kernel K . As we discussed in Section 4.1, many common choices of priors on α and K lead to community-size distributions that become increasingly localized over time. In the present chapter, we consider two representative choices of priors for these parameters.

The first choice of priors is from the method of Yang et al. [88]. In this method, the laziness parameters α are $(0, 0, \dots, 0)$. To choose the transition kernel K , one first assumes that $K^{(\ell)} = \tilde{K}$ for each layer $\ell \in \{2, \dots, L\}$. One then imposes an independent Dirichlet prior with parameters $(\mu_{s_1}, \dots, \mu_{s_k})$ on each row of \tilde{K} . That is,

$$\tilde{K}_{s*} \sim \text{Dir}(\mu_{s*}), \quad (4.13)$$

where $\tilde{K}_{s*} = (\tilde{K}_{s1}, \dots, \tilde{K}_{sk})$ and $\mu_{s*} = (\mu_{s1}, \dots, \mu_{sk})$. In the present chapter, we assume that $\mu_{s*} = (1, \dots, 1)$ for each $s \in \{1, \dots, k\}$.

The second choice of priors is from the method of Bazzi et al. [1]. In this method, one chooses each entry α_ℓ of α uniformly at random from $[0, 1]$. To choose the transition kernel K , one takes each probability vector $K_{s*}^{(\ell)}$ to be the same for all $s \in \{1, \dots, k\}$. That is, $K_{s*}^{(\ell)} = \kappa^{(\ell)}$. One then imposes an independent Dirichlet prior with parameters $\mu^{(\ell)} = (\mu_1^{(\ell)}, \dots, \mu_k^{(\ell)})$ on $\kappa^{(\ell)}$ for each layer $\ell \in \{2, \dots, L\}$. In summary, one takes

$$\begin{aligned}\alpha_\ell &\sim \text{Unif}(0, 1), \\ \kappa^{(\ell)} &\sim \text{Dir}(\mu^{(\ell)}), \\ K_{s*}^{(\ell)} &= \kappa^{(\ell)},\end{aligned}$$

where $K_{s*}^{(\ell)} = (K_{s1}^{(\ell)}, \dots, K_{sk}^{(\ell)})$. In the present chapter, we assume that $\mu^{(\ell)} = (1, \dots, 1)$ for each layer $\ell \in \{2, \dots, L\}$.

The Yang et al. [88] and Bazzi et al. [1] Markov-process approaches both have community-size distributions with less localization than those for a uniform distribution on community assignments (see Section 4.2.3.1). However, in both approaches, the single-layer community-size distributions are more localized for later layers than for earlier layers, which leads to moderate localization in the community-size distributions of a temporal network (see Section 4.3.3). Because many Markov-process approaches (e.g., [35, 50]) have similar choices of prior distributions to the Yang et al. [88] and Bazzi et al. [1] approaches, we also expect increasing localization over time in other Markov-process models.

4.2.3.3 Exchangeability-Based Approach

As we discussed in Section 4.2.3.2, the employed methods for a discrete-time Markov-process approach to community assignment yield increasingly localized single-layer community-size distributions for later layers, and they thus in turn yield more localized community-size distributions in a temporal network. To mitigate this undesirable feature, we introduce a novel approach for community evolution. As we will see in Section 4.3.3, localization of the single-layer community-size distributions increases less over time in this approach than in discrete-time Markov-process approaches. Consequently, the overall community-size distribution of our novel approach is much less localized than those of the discrete-time Markov-process approaches.

The intuition behind our approach is the notion of *exchangeability* [6], which in our context entails that community assignments in a given layer of a temporal network do not distinguish between nodes that were not distinguished by their community assignments in previous layers. With exchangeability, one obtains a uniform distribution on community assignments in monolayer networks (see Section 4.2.2.2). Before generating the community assignments of each node, there is nothing to distinguish the nodes from one another. Because nodes are indistinguishable, the only way to distinguish between different nodes i is to assign them to communities with different sizes n_r . Therefore, we first generate the community sizes uniformly at random, and we then choose a random labeling of the nodes by selecting the community assignment g uniformly at random from the set of all community assignments with the desired community sizes. This is precisely the procedure that we described in Section 4.2.2.2 to generate community assignments with a uniform distribution on community sizes. Additionally, recall that this method has less localized community-size

distributions than a uniform distribution on community assignments (see Sections 4.2.2.1 and 4.3.3), motivating our choice to consider exchangeability.

We now use the concept of exchangeability in temporal networks to motivate our statistical-inference approach. As with the discrete-time Markov-process approaches in Section 4.2.3.2, we generate the community assignments one layer at a time. Because the nodes in the first layer are indistinguishable before one determines their community assignments, we use a nodewise approach for the first layer. Therefore, as before (see (4.10)), we write

$$\begin{aligned} \pi &\sim \text{Dir}(\gamma), \\ g_{(i,1)}|\pi &\sim \pi, \end{aligned} \tag{4.14}$$

where $\gamma = (1, \dots, 1)$. For each layer $\ell \geq 2$, we then treat nodes as indistinguishable if they are in the same community in layer $\ell - 1$. This clearly does not need to be true in practice (because two nodes can certainly have the same community assignment in layer $\ell - 1$ but different community assignments in a layer $\ell' < \ell - 1$), but this simplification allows us to formulate a computationally tractable approach. When generating the community assignments for layer ℓ , we partition the set of nodes into k sets, where the r th set (with $r \in \{1, \dots, k\}$) consists of the nodes in community r in layer $\ell - 1$. We treat the nodes in each set as indistinguishable from each other. In a similar fashion as our procedure to generate community assignments in the first layer, for each set, our community-assignment procedure consists of first choosing the community sizes and then randomly labeling the nodes in the set by selecting the community assignments uniformly at random from the set of all community assignments with the desired community sizes.

To make the above approach scientifically rigorous, we begin by introducing no-

tation for weak compositions. A *weak composition* of a positive integer n with k parts is an ordered sequence (n_1, \dots, n_k) of non-negative integers that satisfies the constraint $\sum_{r=1}^k n_r = n$. We use \mathcal{C}_n^k to denote the set of all weak compositions of n with k parts, and we write $\text{Unif}(\mathcal{C}_n^k)$ for the uniform distribution on \mathcal{C}_n^k .

Given a weak composition $c = (c_1, \dots, c_k) \in \mathcal{C}_n^k$ and a set $\mathcal{G} = \{1, \dots, n\}$ of nodes, let $\mathcal{G}(c)$ denote the set of all community assignments $g = (g_1, \dots, g_n) \in \{1, \dots, k\}^{\mathcal{G}}$ such that

$$|\{g_i = r \mid i \in \{1, \dots, k\}\}| = c_r$$

for all $r \in \{1, \dots, k\}$. We write $\text{Unif}(\mathcal{G}(c))$ to refer to the uniform distribution on $\mathcal{G}(c)$.

Using this notation, we can now describe our approach for community evolution in detail. In particular, we provide the process to generate the community assignments $g^{(\ell)}$ in layer ℓ given the community assignments $g^{(\ell-1)}$ in layer $\ell - 1$. Let $n_r(g) = |\{g_i = r \mid i \in \{1, \dots, k\}\}|$ be the number of times that r appears in g , let $n(g) = (n_1(g), \dots, n_k(g))$, and let

$$\mathcal{G}_r^{(\ell)} = \{i \in \{1, \dots, n\} \mid g_{(i,\ell)} = r\} \quad (4.15)$$

be the r th community in $g^{(\ell)}$. It follows from the definition (4.15) that $|\mathcal{G}_r^{(\ell)}| = n_r(g^{(\ell)})$.

As we discussed previously, we consider each set $\mathcal{G}_r^{(\ell)}$ separately and choose the community sizes when restricted to each individual set. For notational convenience, we write $\mathbf{c}_r^{(\ell)} = (c_{r1}^{(\ell)}, \dots, c_{rk}^{(\ell)})$, where $c_{rs}^{(\ell)}$ is the number of nodes in community r in layer $\ell - 1$ that are in community s in layer ℓ . In contrast to monolayer networks, we desire that temporal networks bias community assignments toward remaining unchanged between layers, as we want our generative model to generate communities

whose memberships remain somewhat stable over time. To do this, we bias the number $c_{rr}^{(\ell)}$ of nodes that stay in community r toward the value $n_r(g_{(\ell-1)})$, which is the size of community r in layer $\ell - 1$, by sampling it from a geometric distribution. Specifically, let $\overline{\text{Geom}}(n, p)$ be the distribution with probability mass function

$$\mathbb{P}(m) = p^{n-m} \frac{1-p}{1-p^{n+1}}, \quad (4.16)$$

where $m \in \{0, 1, \dots, n\}$. We sample $c_{rr}^{(\ell)}$ from $\overline{\text{Geom}}(n_r(g_{(\ell-1)}), p_{r,\ell})$ and impose independent uniform priors over $[0, 1]$ on $p_{r,\ell}$ for each $r \in \{1, \dots, k\}$ and $\ell \in \{1, \dots, L\}$. Let $\mathbf{c}_{r,-r}^{(\ell)} = (c_{r1}^{(\ell)}, \dots, c_{r,r-1}^{(\ell)}, c_{r,r+1}^{(\ell)}, \dots, c_{rk}^{(\ell)})$ denote the vector $\mathbf{c}_r^{(\ell)}$ with the r th component removed. We do not want our method to be biased for or against any community other than community r , so we sample $\mathbf{c}_{r,-r}^{(\ell)}$ uniformly at random from all weak compositions of $n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)}$ into $k - 1$ parts. Given $\mathbf{c}_r^{(\ell)}$, we sample $\mathbf{g}'_{r,\ell}$ from a uniform distribution on $\mathcal{G}_r^{(\ell-1)}(\mathbf{c}_r^{(\ell)})$. Finally, we concatenate $\mathbf{g}'_{r,\ell} \in \{1, \dots, k\}^{\mathcal{G}_r^{(\ell-1)}}$ over $r \in \{1, \dots, k\}$ to obtain the community assignments $g_{(\ell)} \in \{1, \dots, k\}^n$ in layer ℓ . To make our notation compact, we write $g_{(\ell)} = \bigoplus_{r=1}^k \mathbf{g}'_{r,\ell}$. In summary, for our approach, given the layer- $(\ell - 1)$ communities $g_{(\ell-1)}$, we generate the layer- ℓ communities $g_{(\ell)}$ by first sampling $\mathbf{g}'_{r,\ell}$ for each $r \in \{1, \dots, k\}$ according to the following procedure:

$$\begin{aligned} p_{r,\ell} &\sim \text{Unif}(0, 1), \\ c_{rr}^{(\ell)} \mid p_{r,\ell} &\sim \overline{\text{Geom}}(n_r(g_{(\ell-1)}), p_{r,\ell}), \\ \mathbf{c}_{r,-r}^{(\ell)} \mid c_{rr}^{(\ell)} &\sim \text{Unif}\left(\mathcal{C}_{n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)}}^{k-1}\right), \\ \mathbf{g}'_{r,\ell} \mid \mathbf{c}_r^{(\ell)} &\sim \text{Unif}(\mathcal{G}_r^{(\ell-1)}(\mathbf{c}_r^{(\ell)})), \end{aligned} \quad (4.17)$$

where we recall that $\gamma = (1, \dots, 1)$. We then set $g_{(\ell)} = \bigoplus_{r=1}^k \mathbf{g}'_{r,\ell}$.

4.3 Localization of Community-Size Distributions

We now numerically compute community-size distributions for each of the generative models in Sections 4.2.2 and 4.2.3. We obtain the following results:

- Monolayer-network generative models:
 - The community-size distribution of a uniform distribution on community assignments (see Section 4.2.2.1) is highly localized.
 - The community-size distribution of nodewise community assignments (see Section 4.2.2.2) is much less localized than the community-size distribution of a uniform distribution on community assignments.
- Temporal-network generative models:
 - The community-size distribution of a uniform distribution on community assignments (see Section 4.2.3.1) is highly localized.
 - The community-size distributions of the Yang et al. [88] and Bazzi et al. [1] discrete-time Markov-process approaches (see Section 4.2.3.2) are much less localized than that of a uniform distribution on community assignments. However, for both the Yang et al. and Bazzi et al. approaches,, the single-layer community-size distributions of later layers are more localized than those of earlier layers.
 - In our approach (see Section 4.2.3.3), the localization of the single-layer community-size distributions increases much more slowly than it does for the Yang et al. [88] and Bazzi et al. [1] approaches. Consequently, the

overall community-size distribution is much less localized than those of the Yang et al. and Bazzi et al. approaches.

In Section 4.3.1, we discuss the methodology that we use to verify these claims. In Section 4.3.2, we verify the above claims for monolayer-network generative models. In Section 4.3.3, we verify the above claims for temporal-network generative models. Finally, in Section 4.3.4, we present a conjecture for the behavior of the single-layer community-size distributions in our approach (see Section 4.2.3.3) in the limit of infinitely many layers.

4.3.1 Methodology

To compute the community-size distributions for each of the generative models in Sections 4.2.2 and 4.2.3, we begin by using them to generate $M = 10^6$ instantiations of community assignments g .

For monolayer networks, we report the “empirical distribution” (i.e., histogram) of the size of community 1. We let $g^{(m)}$ be the m th community-assignment instantiation, and we plot the observed frequencies P_i for each $i \in \{0, \dots, n\}$, where P_i is defined by the map

$$i \mapsto P_i := \frac{1}{M} \sum_{m=1}^M \delta_{n_1(g^{(m)}), i}. \quad (4.18)$$

By symmetry, we choose the community label “1” without loss of generality. In Figure 4.1, we show an example of such a histogram.

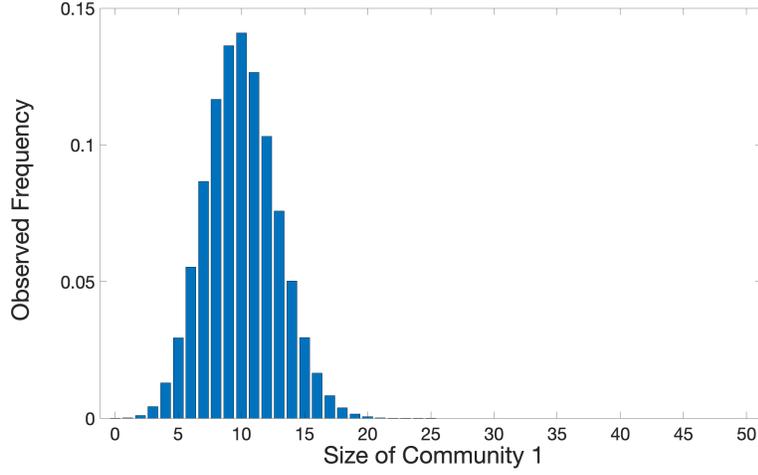


Figure 4.1: An example of the community-size frequency histogram for a monolayer network. The horizontal axis indicates the number of nodes in the network with community assignment 1, and the vertical axis indicates the observed frequency of community assignments with that number of nodes in community 1.

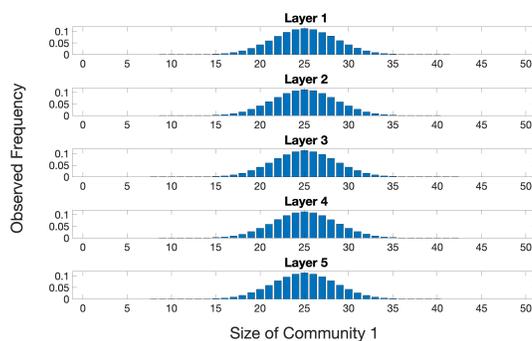
For temporal networks, we first count the number of node-layers (i, ℓ) in layer ℓ that are in community 1, for each layer $\ell \in \{1, \dots, L\}$. We also count the total number of node-layers that are in community 1 across a whole network. For each layer ℓ , we then generate a histogram for each layer of the observed frequencies P_i for each $i \in \{0, \dots, n\}$, where P_i is defined by the map

$$i \mapsto P_i := \frac{1}{M} \sum_{m=1}^M \delta_{n_1(g^{(m)}), i} \quad (4.19)$$

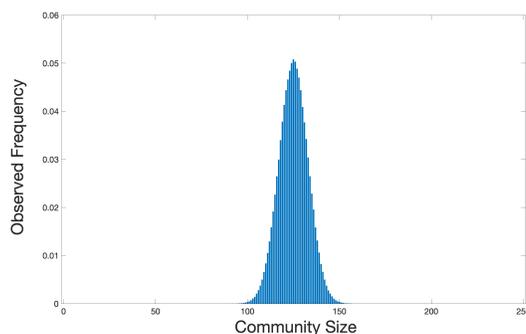
in that layer. We also generate a histogram of the observed frequencies P_i for each $i \in \{0, \dots, n\}$, where P_i is defined by the map

$$i \mapsto P_i := \frac{1}{M} \sum_{m=1}^M \delta_{n_1(g^{(m)}), i} \quad (4.20)$$

for the overall network. In Figure 4.2, we show an example of such a set of histograms.



(a) Community-size histograms for each layer



(b) Community-size histograms for the overall network

Figure 4.2: An example of the community-size histograms for a temporal network with 5 layers. In both panels (a) and (b), the horizontal axis indicates the number of node-layers with community assignment 1, and the vertical axis indicates the observed frequency of community assignments with that number of node-layers in community 1. For the histograms in (a), we consider only the node-layers in the indicated layer when counting the number of nodes in community 1. In the histogram in (b), we consider all of the node-layers in the network when counting the number of nodes in community 1.

In addition to using histograms to qualitatively compare the amount of localization of community-size distributions between different generative models, we calculate the inverse participation ratio (IPR) [43] to quantify the amount of localization in each community-size distribution. The IPR of a community-size distribution is the squared L^2 -norm of the distribution. That is,

$$\text{IPR} = \sum_{i=0}^n P_i^2, \quad (4.21)$$

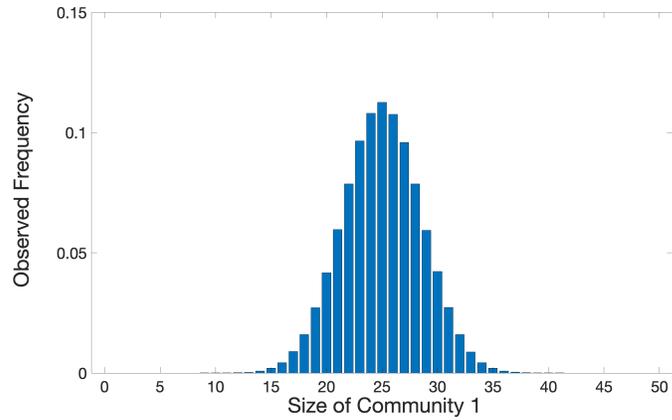
where P_i is one of (4.18), (4.19), or (4.20), depending on which type of community-size distribution we are considering. Larger values of the IPR indicate more localized distributions. The minimum value of the IPR is $\frac{1}{n+1}$ and is attained when $P_i = \frac{1}{n+1}$ for each i . The maximum value of the IPR is 1 and is attained when $P_j = 1$ for some j and $P_i = 0$ for all $i \neq j$.

In the following subsections, we examine the community-size histograms and calculate their IPRs to quantify and compare the amount of localization in the examined generative models for community detection.

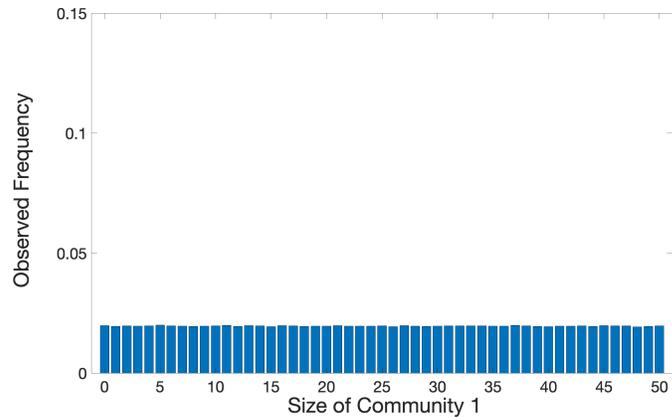
4.3.2 Community-Size Distributions of Generative Models of Monolayer Networks

We consider two community-assignment probability distributions for monolayer networks: a uniform distribution on community assignments (see Section 4.2.2.1) and nodewise community assignments (see Section 4.2.2.2). Using the approach in Section 4.3.1, we generate a community-size histogram for each of the two community-assignment probability distributions. We set the number of nodes to $n = 50$ and generate two community-size histograms for each of the two community-assignment probability distributions. One distribution has $k = 2$ communities, and the other

has $k = 5$ communities. We show the resulting community-size histograms for $k = 2$ in Figure 4.3 and the histograms for $k = 5$ in Figure 4.4. We compute the IPR for the community-size distribution of each example and compile our results in Table 4.1.

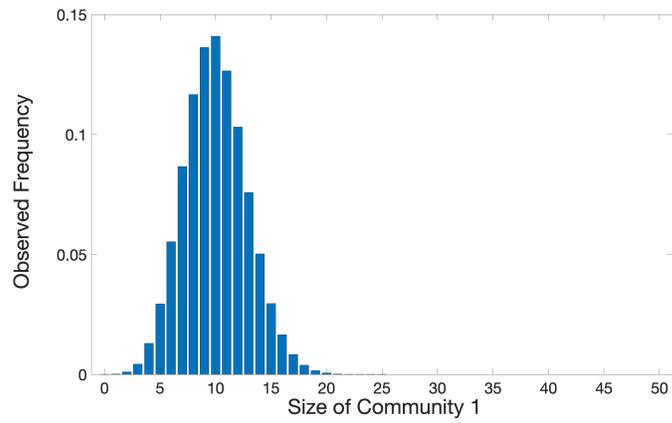


(a) Uniform distribution on community assignments

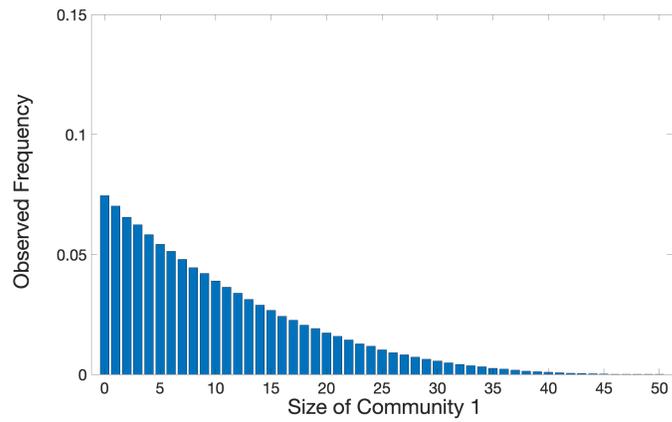


(b) Nodewise community assignments

Figure 4.3: Community-size histograms for monolayer networks for $k = 2$ communities for (a) a uniform distribution on community assignments and (b) nodewise community assignments.



(a) Uniform distribution on community assignments



(b) Nodewise community assignments

Figure 4.4: Community-size histograms for monolayer networks for $k = 5$ communities for (a) a uniform distribution on community assignments and (b) nodewise community assignments.

| Method | IPR for $k = 2$ | IPR for $k = 5$ |
|---|-----------------|-----------------|
| Uniform distribution on community assignments | 0.0796 | 0.0998 |
| Nodewise community assignments | 0.0196 | 0.0436 |

Table 4.1: Values of the inverse participation ratio (IPR) for single-layer community-size distributions.

We begin our analysis of our results by restating our claims (see Sections 4.2.2 and 4.3) for monolayer networks:

- The community-size distribution for a uniform distribution on community assignments (see Section 4.2.2.1) is highly localized.
- The community-size distribution for nodewise community assignments (see Section 4.2.2.2) is much less localized than that for a uniform distribution on community assignments (see Section 4.2.2.1).

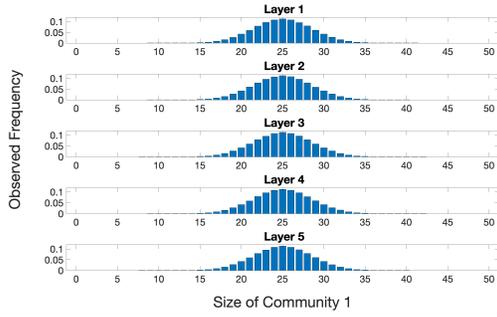
In Figures 4.3 and 4.4, we see that both claims appear to hold for both $k = 2$ communities and $k = 5$ communities. The histograms for a uniform distribution on community assignments appear to be highly localized, whereas the histograms for nodewise community assignments are far less localized.

These qualitative observations are confirmed by the IPRs of the community-size distributions in Table 4.1. For example, for $k = 2$, the IPR for the uniform distribution on community assignments is 0.0796, which is much larger than the IPR value of 0.0196 for nodewise community assignments. Recall that larger IPR values indicate more localization of a distribution. Therefore, for $k = 2$, the community-

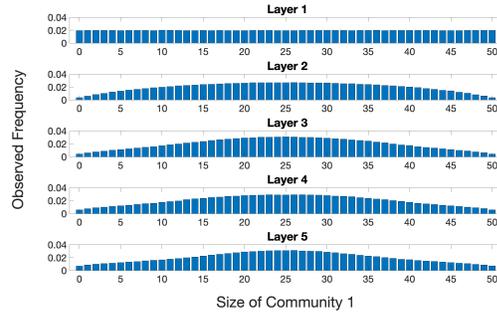
size distribution for nodewise community assignments is much less localized than the community-size distribution for a uniform distribution on community assignments. We observe the same behavior for $k = 5$.

4.3.3 Community-Size Distributions of Generative Models of Temporal Networks

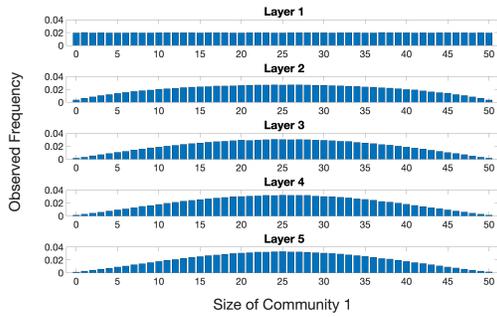
We consider the following community-assignment probability distributions for temporal networks: a uniform distribution on community assignments (see Section 4.2.3.1), the Yang et al. [88] and Bazzi et al. [1] discrete-time Markov-process approaches (see Section 4.2.3.2), and our exchangeability-based approach (see Section 4.2.3.3). As with monolayer networks, we use the approach in Section 4.3.1 to generate a set of community-size histograms (one for each layer and one for the overall network) for each of the four generative models. We again set the number of nodes to $n = 50$ and again consider $k = 2$ and $k = 5$ communities. We show the histogram for each layer for $k = 2$ in Figure 4.5, the histogram for the overall network for $k = 2$ in Figure 4.6, the histogram for each layer for $k = 5$ in Figure 4.7, and the histogram for the overall network for $k = 5$ in Figure 4.8. We compute the IPR for the community-size distribution for each example, and we compile the IPR values for each layer in Table 4.2 and the IPR values for the overall network in Table 4.3.



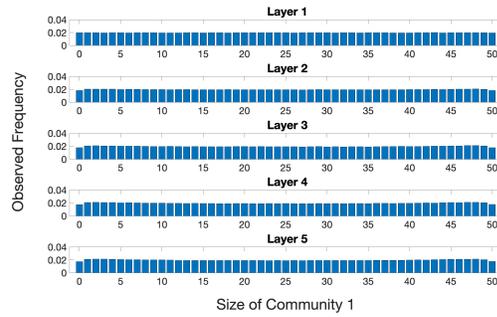
(a) Uniform distribution on community assignments



(b) Yang et al. approach

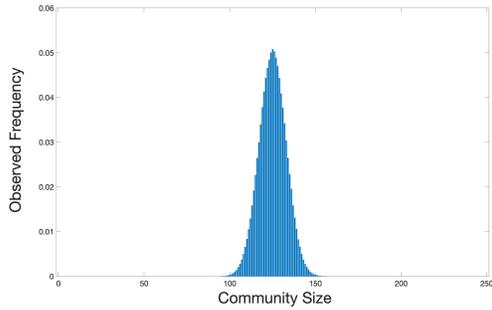


(c) Bazzi et al. approach

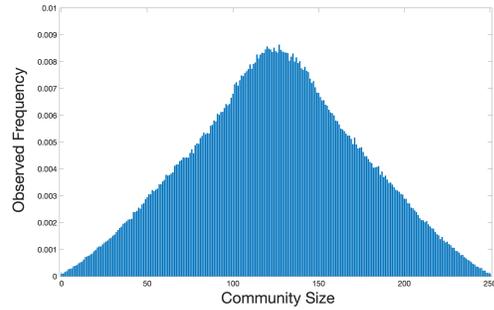


(d) Our approach

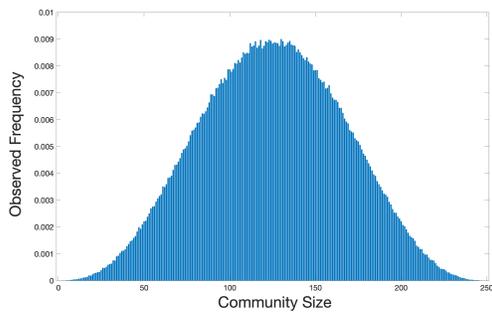
Figure 4.5: Community-size histograms for each layer of a temporal network for $k = 2$ communities for (a) a uniform distribution on community assignments, (b) the Yang et al. approach [88], (c) the Bazzi et al. approach [1], and (d) our exchangeability-based approach.



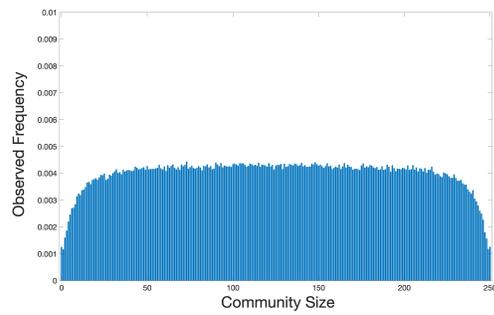
(a) Uniform distribution on community assignments



(b) Yang et al. approach

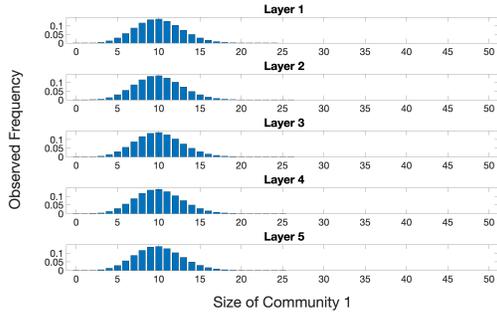


(c) Bazzi et al. approach

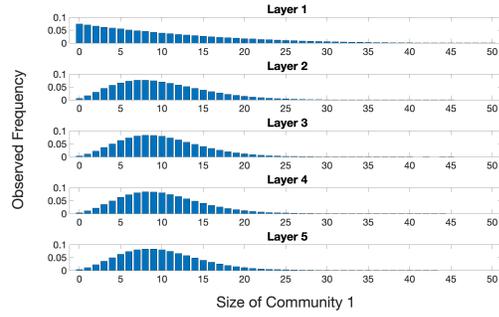


(d) Our approach

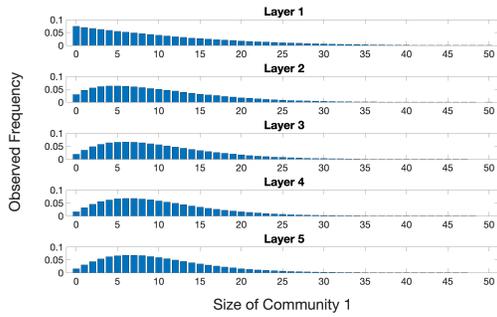
Figure 4.6: Community-size histograms for the overall network for $k = 2$ communities for (a) a uniform distribution on community assignments, (b) the Yang et al. approach, (c) the Bazzi et al. approach, and (d) our exchangeability-based approach.



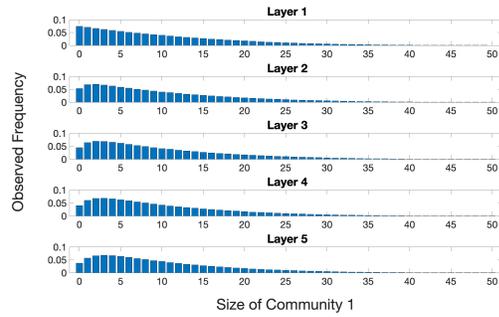
(a) Uniform distribution on community assignments



(b) Yang et al. approach

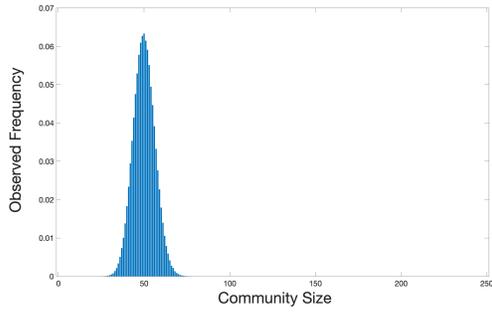


(c) Bazzi et al. approach

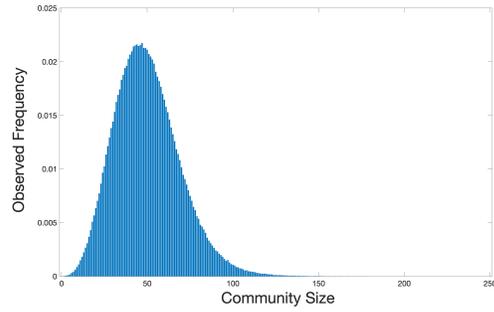


(d) Our approach

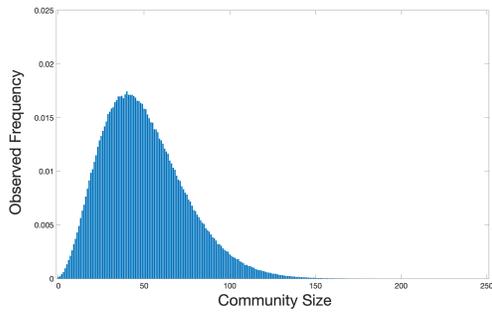
Figure 4.7: Community-size histograms for each layer of a temporal network for $k = 5$ communities for (a) a uniform distribution on community assignments, (b) the Yang et al. approach, (c) the Bazzi et al. approach, and (d) our exchangeability-based approach.



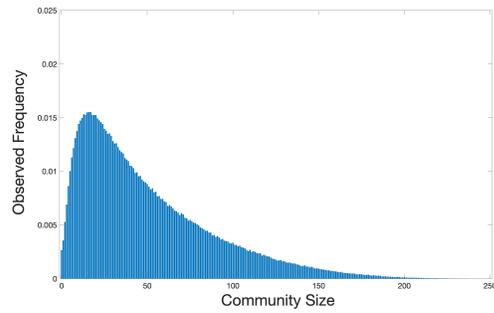
(a) Uniform distribution on community assignments



(b) Yang et al. approach



(c) Bazzi et al. approach



(d) Our approach

Figure 4.8: Community-size histograms for the overall network for $k = 5$ communities for (a) a uniform distribution on community assignments, (b) the Yang et al. approach, (c) the Bazzi et al. approach, and (d) our exchangeability-based approach.

| Method | Layer | IPR for $k = 2$ | IPR for $k = 5$ |
|---|---------|-----------------|-----------------|
| Uniform distribution on community assignments | Layer 1 | 0.0795 | 0.0997 |
| | Layer 2 | 0.0794 | 0.0998 |
| | Layer 3 | 0.0798 | 0.0997 |
| | Layer 4 | 0.0795 | 0.0997 |
| | Layer 5 | 0.0797 | 0.0998 |
| Yang et al. [88] | Layer 1 | 0.0196 | 0.0436 |
| | Layer 2 | 0.0220 | 0.0545 |
| | Layer 3 | 0.0232 | 0.0595 |
| | Layer 4 | 0.0225 | 0.0597 |
| | Layer 5 | 0.0226 | 0.0597 |
| Bazzi et al. [1] | Layer 1 | 0.0196 | 0.0435 |
| | Layer 2 | 0.0220 | 0.0448 |
| | Layer 3 | 0.0239 | 0.0471 |
| | Layer 4 | 0.0247 | 0.0481 |
| | Layer 5 | 0.0250 | 0.0485 |
| Our exchangeability-based approach | Layer 1 | 0.0196 | 0.0435 |
| | Layer 2 | 0.0196 | 0.0437 |
| | Layer 3 | 0.0196 | 0.0440 |
| | Layer 4 | 0.0196 | 0.0442 |
| | Layer 5 | 0.0196 | 0.0443 |

Table 4.2: IPR values of the community-size distribution of each layer for each examined generative model for community assignments in temporal networks.

| Method | IPR for $k = 2$ | IPR for $k = 5$ |
|---|-----------------|-----------------|
| Uniform distribution on community assignments | 0.0357 | 0.0446 |
| Yang et al. [88] | 0.0057 | 0.0153 |
| Bazzi et al. [1] | 0.0066 | 0.0122 |
| Our exchangeability-based approach | 0.0041 | 0.0095 |

Table 4.3: IPR values of the community-size distribution of the overall network for each examined generative model of community assignments in temporal networks.

For convenience, we restate our previous claims (see the introduction of Section 4.3) about temporal networks:

- The community-size distribution of a uniform distribution on community assignments (see Section 4.2.3.1) is highly localized.
- The community-size distributions of the Yang et al. [88] and Bazzi et al. [1] discrete-time Markov-process approaches (see Section 4.2.3.2) are much less localized than that of a uniform distribution on community assignments. However, for both the Yang et al. and Bazzi et al. approaches, the single-layer community-size distributions of later layers are more localized than those of earlier layers.
- In our approach (see Section 4.2.3.3), the localization of the single-layer community-size distributions increases much more slowly than it does for the Yang et al. [88] and Bazzi et al. [1] approaches. Consequently, the overall community-size distribution is much less localized than those of the Yang et al. and Bazzi et al. approaches.

In Figures 4.5, 4.6, 4.7, and 4.8, we see that all of these claims appear to hold for both $k = 2$ communities and $k = 5$ communities. In the example in Figure 4.6, we see for $k = 2$ that the overall community-size histogram is much more localized for a uniform distribution on community assignments than for the three other methods. We also observe that the community-size histogram for the overall network appears to be less localized in our approach than in those of the Yang et al. [88] or Bazzi et al. [1] approaches.

Our qualitative observations are also confirmed by the IPR values in Tables 4.2 and 4.3. For example, for $k = 2$, the IPR of the overall community-size distribution is 0.0357 for a uniform distribution on community assignments, 0.0057 for the Yang et al. method, 0.0066 for the Bazzi et al. method, and 0.0041 for our exchangeability-based approach, confirming that the overall community-size distribution is much more localized for a uniform distribution on community assignments than for the three other methods and that the community-size distribution for our approach is less localized those of the Yang et al. and Bazzi et al. approaches. Additionally, for $k = 2$, the IPRs of the single-layer community-size distributions for the Yang et al. approach increase from 0.0196 in layer 1 to 0.0226 in layer 5, illustrating that the single-layer community-size distributions become more localized over time.

4.3.4 Asymptotic Behavior of Our Exchangeability-Based Approach

Recall from Section 4.2.3.3 that our exchangeability-based approach generates the layer- ℓ communities $g_{(\ell)}$ given the layer- $(\ell - 1)$ communities $g_{(\ell-1)}$ by sampling $\mathbf{g}'_{r,\ell}$

according to the procedure given in (4.17). Namely, we take

$$\begin{aligned}
\pi &\sim \text{Dir}(\gamma), \\
g_{(i,1)} | \pi &\sim \pi, \\
p_{r,\ell} &\sim \text{Unif}(0, 1), \\
c_{rr}^{(\ell)} | p_{r,\ell} &\sim \overline{\text{Geom}}(n_r(g_{(\ell-1)}), p_{r,\ell}), \\
\mathbf{c}_{r,-r}^{(\ell)} | c_{rr}^{(\ell)} &\sim \text{Unif}\left(\mathcal{C}_{n_r(g_{(\ell-1)})-c_{rr}^{(\ell)}}^{k-1}\right), \\
\mathbf{g}'_{r,\ell} | \mathbf{c}_r^{(\ell)} &\sim \text{Unif}(\mathcal{G}_r^{(\ell-1)}(\mathbf{c}_r^{(\ell)})),
\end{aligned}$$

where we recall that $\gamma = (1, \dots, 1)$. We then set $g_{(\ell)} = \bigoplus_{r=1}^k \mathbf{g}'_{r,\ell}$. We previously showed empirically (see Section 4.3.3) that the localization of the single-layer community-size distributions increases much more slowly for this approach than it does for the Yang et al. [88] and Bazzi et al. [1] approaches.

We now discuss a conjecture about the behavior of the single-layer community-size distributions of a similar community-assignment probability distribution in the limit of infinitely many layers. The community-assignment probability distribution that we consider generates the layer- ℓ communities $g_{(\ell)}$ given the layer- $(\ell-1)$ communities $g_{(\ell-1)}$ by sampling $\mathbf{g}'_{r,\ell}$ according to the following procedure:

$$\begin{aligned}
\pi &\sim \text{Dir}(\gamma), \\
g_{(i,1)} | \pi &\sim \pi, \\
c_{rr}^{(\ell)} | p &\sim \overline{\text{Geom}}(n_r(g_{(\ell-1)}), p), \\
\mathbf{c}_{r,-r}^{(\ell)} | c_{rr}^{(\ell)} &\sim \text{Unif}\left(\mathcal{C}_{n_r(g_{(\ell-1)})-c_{rr}^{(\ell)}}^{k-1}\right), \\
\mathbf{g}'_{r,\ell} | \mathbf{c}_r^{(\ell)} &\sim \text{Unif}(\mathcal{G}_r^{(\ell-1)}(\mathbf{c}_r^{(\ell)})), \tag{4.22}
\end{aligned}$$

where $\gamma = (1, \dots, 1)$ and $p \in (0, 1)$ is a specified parameter. The procedure (4.22) is the same as the procedure (4.17) except that we choose $p_{r,\ell} = p$ instead sampling $p_{r,\ell}$ independently from a uniform distribution on $(0, 1)$. We conjecture the following result about the single-layer community-size distributions of the community-assignment probability distribution that generates communities according to the procedure (4.22):

Conjecture 1. *As the layer $\ell \rightarrow \infty$, the single-layer community-size distributions satisfy*

$$P_i \rightarrow \frac{(1 - p^{i+1})(1 - p^{n-i+1})}{\sum_{j=0}^n (1 - p^{j+1})(1 - p^{n-j+1})}, \quad (4.23)$$

where P_i is defined in (4.19).

To demonstrate that Conjecture 1 is plausible, we generate $M = 10^6$ instantiations of community assignments using the community-assignment probability distribution that samples according to (4.22). We set the number L of layers to be 50, the number n of nodes to 50, the number k of communities to 2, and the parameter p to 0.6. In Figure 4.9, we plot the empirical distribution P_i of the size of community 1 in layer 50 and the conjectured distribution from (4.23).

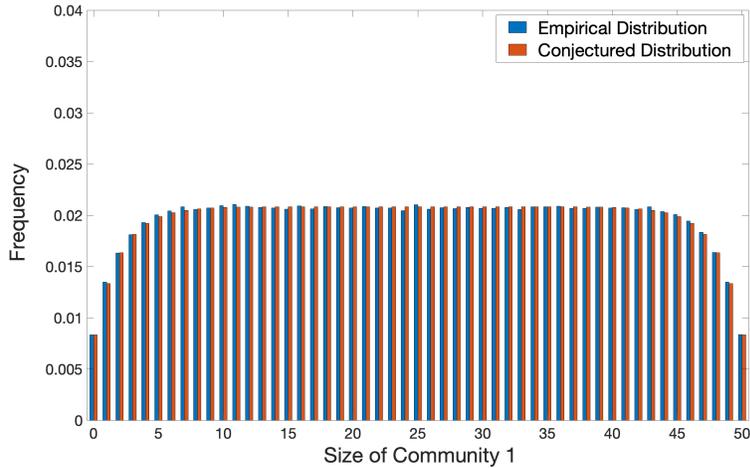


Figure 4.9: A comparison of the community-size histogram for layer 50 of a temporal network with $k = 2$ groups for a community-assignment probability distribution that samples using the procedure (4.22) (blue) and the conjectured distribution from (4.23) (red).

In Figure 4.9, we see that the conjectured distribution is almost the same as the empirical distribution P_i for each community size i , which suggests that Conjecture 1 is plausible. If we are able to prove Conjecture 1, it would further support the claim that our exchangeability-based approach has less-localized community-size distributions than Markov-process models (such as the Bazzi et al. [1] and Yang et al. [88] methods), because the conjectured distribution in (4.23) is almost uniform for community sizes i away from the boundary (i.e., very close to 0 or very close to the number n of nodes in the network).

4.3.5 Summary

Our numerical computations confirm our claims that the choice of generative model for community assignments has a significant effect on the localization of community-size distributions. We observed that a uniform distribution on community assignments yields substantially more localized community-size distributions than the other examined methods for temporal community structure. We also observed that the single-layer community-size distributions of the discrete-time Markov-process approaches become more localized over time, whereas the localization of the single-layer community-size distributions increases much more slowly with time for our exchangeability-based approach. Accordingly, in temporal networks, we conclude that our exchangeability-based approach has less localized overall community-size distributions than discrete-time Markov-process models do.

4.4 Comparison of Community-Detection Performance of Statistical-Inference Approaches

As we mentioned in Section 4.1, we expect that statistical-inference methods that employ generative models with more-localized community-size distributions will have poorer performance in networks with large or small communities than methods with less-localized community-size distributions. In this section, we demonstrate that this is indeed the case.

4.4.1 Posterior-Sampling Approaches

We first describe the algorithms that we use to sample from the posterior distributions $\mathbb{P}(g|A)$ for each choice of community-assignment probability distribution $\mathbb{P}(g)$. Recall from equation (4.2) that the generative models of networks with community structure that we consider take the form

$$\mathbb{P}(A, g) = \mathbb{P}(A|g)\mathbb{P}(g),$$

where $\mathbb{P}(A|g)$ is the probability of generating a network with adjacency structure A given community assignments g and the term $\mathbb{P}(g)$ is the probability of the community assignment g . We seek to examine the impact of localization of a community-size distribution of a community-assignment generative model $\mathbb{P}(g)$, so we vary only the choice of $\mathbb{P}(g)$ in the examined methods. All examined methods use the same $\mathbb{P}(A|g)$.

We consider independent SBMs in each layer, so

$$\mathbb{P}(A|g, \omega) = \prod_{\ell=1}^L \mathbb{P}(A^{(\ell)}|g^{(\ell)}, \omega^{(\ell)}) = \prod_{\ell=1}^L \prod_{1 \leq i < j \leq n} \left(\omega_{g^{(\ell)}(i), g^{(\ell)}(j), \ell}^{(\ell)} \right)^{A_{ij}^{(\ell)}} \left(1 - \omega_{g^{(\ell)}(i), g^{(\ell)}(j), \ell}^{(\ell)} \right)^{1 - A_{ij}^{(\ell)}}, \quad (4.24)$$

where $\omega^{(\ell)} \in [0, 1]^{k \times k}$ (for each $\ell \in \{1, \dots, L\}$) is a matrix whose entries $\omega_{rs}^{(\ell)}$ control the probability of edges between nodes in communities r and s in layer ℓ . We set the prior distributions for each $\omega_{rs}^{(\ell)}$ to be independent uniform distributions on $[0, 1]$ for each r and s such that $1 \leq r \leq s \leq k$. We set $\omega_{rs}^{(\ell)} = \omega_{sr}^{(\ell)}$ if $r > s$ (i.e., we force $\omega^{(\ell)}$ to be symmetric). Integrating¹ $\mathbb{P}(A|g, \omega)$ with respect to the probability measure that is induced by $\mathbb{P}(\omega)$ yields the posterior distribution

$$\mathbb{P}(A|g) = \prod_{\ell=1}^L \prod_{1 \leq r \leq s \leq k} \frac{m_{rs}^{(\ell)}!(t_{rs}^{(\ell)} - m_{rs}^{(\ell)})!}{(t_{rs}^{(\ell)} + 1)!}, \quad (4.25)$$

¹See [68] for the computation of similar integrals.

where

$$t_{rs}^{(\ell)} := \sum_{1 \leq i, j \leq n; i \neq j} \left(\delta_{g(i, \ell), r} \delta_{g(j, \ell), s} \right),$$

$$m_{rs}^{(\ell)} := \sum_{1 \leq i, j \leq n; i \neq j} A_{ij}^{(\ell)} \left(\delta_{g(i, \ell), r} \delta_{g(j, \ell), s} \right)$$

for $r \neq s$ and

$$t_{rr}^{(\ell)} := \frac{1}{2} \sum_{1 \leq i, j \leq n; i \neq j} \delta_{g(i, \ell), r} \delta_{g(j, \ell), r},$$

$$m_{rr}^{(\ell)} := \frac{1}{2} \sum_{1 \leq i, j \leq n; i \neq j} A_{ij}^{(\ell)} \left(\delta_{g(i, \ell), r} \delta_{g(j, \ell), r} \right).$$

That is, for $r \neq s$, the quantity $t_{rs}^{(\ell)}$ is the number of pairs of distinct node-layers (i, ℓ) and (j, ℓ) for which (i, ℓ) is in community r and (j, ℓ) is in community s . Similarly, $m_{rs}^{(\ell)}$ is the number of such pairs that are connected directly by an edge. The expressions for $t_{rr}^{(\ell)}$ and $m_{rr}^{(\ell)}$ have an additional factor of $1/2$ to avoid double-counting.

We consider three choices of $\mathbb{P}(g)$: a uniform distribution on community assignments (see Section 4.2.3.1), the discrete-time Markov-process approach of Bazzi et al. [1] (see Section 4.2.3.2), and our exchangeability-based approach (see Section 4.2.3.3). None of these three probability distributions include additional parameters beyond the number k (which we fix) of communities. Recall that each of the examined generative models has the form

$$\mathbb{P}(A, g) = \mathbb{P}(A|g)\mathbb{P}(g). \tag{4.26}$$

As we discussed in Section 4.2.1, given a network with adjacency structure A , we perform statistical inference using a generative model to determine the community assignment g . We sample from the posterior distribution $\mathbb{P}(g|A)$, which we recall

from (4.3) has the form

$$\mathbb{P}(g|A) = \frac{\mathbb{P}(A|g)\mathbb{P}(g)}{\mathbb{P}(A)}. \quad (4.27)$$

However, due to the relatively complicated nature of the expressions for $\mathbb{P}(A|g)$ and $\mathbb{P}(g)$, directly sampling from the posterior distribution is not tractable. Therefore, we use Gibbs sampling [63] to approximately sample from the posterior distribution $\mathbb{P}(g|A)$. In one iteration of Gibbs sampling, we loop over each node-layer (i, ℓ) and sample $g_{(i,\ell)}$ from

$$\mathbb{P}(g_{(i,\ell)}|\tilde{g}, A), \quad (4.28)$$

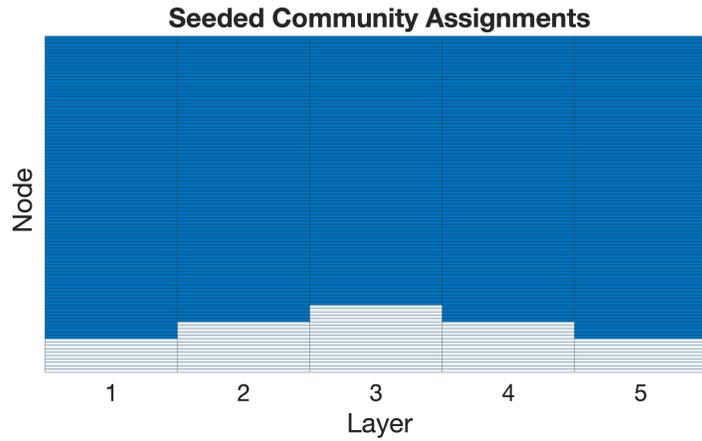
where \tilde{g} is the set of current community assignments aside from $g_{(i,\ell)}$. In our Gibbs-sampling procedure, we begin by sampling from the posterior (4.28) to generate $g_{(1,1)}, g_{(1,2)}, \dots, g_{(1,L)}$ in that order. We then continue sampling in the same fashion for each node $i \in \{2, \dots, n\}$ in order.

In practice, we sample from the distribution $\mathbb{P}(g_{(i,\ell)}|\tilde{g}, A)$ by computing

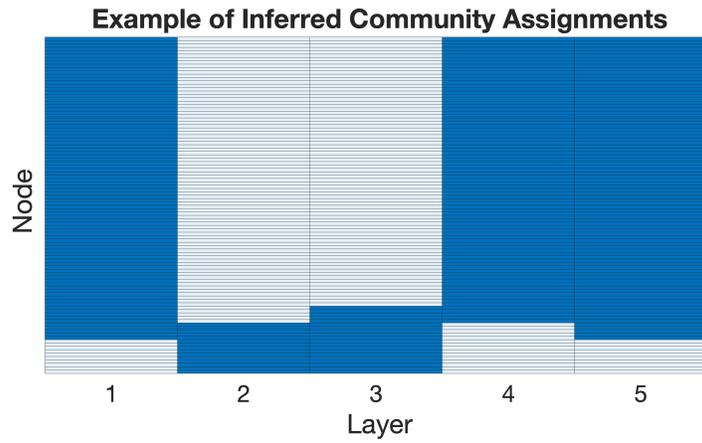
$$\mathbb{P}(g_{(i,\ell)} = r|\tilde{g}, A) = \frac{\mathbb{P}(A, g_{(i,\ell)} = r, \tilde{g})}{\sum_{s=1}^k \mathbb{P}(A, g_{(i,\ell)} = s, \tilde{g})}, \quad (4.29)$$

where $\mathbb{P}(A, g_{(i,\ell)} = r, \tilde{g})$ is $\mathbb{P}(A, g)$ evaluated for the specified adjacency structure A and for a community assignment g that includes all current community assignments aside from $g_{(i,\ell)}$, which we take to be r .

Unfortunately, applying this naive Gibbs-sampling approach often does not lead to correct community identifications in temporal networks. The reason is that the posterior distribution $\mathbb{P}(g|A)$, which we approximate via Gibbs sampling, has many local maxima that cause the Gibbs-sampling algorithm to often become stuck at suboptimal maxima. In Figure 4.10, we illustrate common patterns that we observe at such suboptimal maxima.



(a) Seeded community structure



(b) Example of local extremum

Figure 4.10: Heat maps of (a) an example of actual community structure and (b) an illustration of the permuted community structure that we observe commonly at local maxima of $\mathbb{P}(g|A)$ in a 100-node network with 5 layers. Each rectangle in a heat map corresponds to one node-layer (i, ℓ) . Dark blue rectangles signify the community assignment $g_{(i, \ell)} = 1$, and light blue rectangles signify the community assignment $g_{(i, \ell)} = 2$.

In this example, the inferred community structure for each layer is accurate (as either community can be assigned the label “1”), but the overall community structure is not accurate. To mitigate this issue, we augment the Gibbs-sampling procedure with multi-node moves. With probability p , instead of sampling $g_{(i,\ell)}$ according to the Gibbs-sampling procedure that we discussed above, we choose $r, s \in \{1, \dots, k\}$ and $\ell \in \{1, \dots, L\}$ uniformly at random and propose a move that permutes two community labels r and s for all layers at or beyond layer ℓ . The proposed community assignment g^* satisfies

$$g_{(i,\ell')}^* = \begin{cases} s, & \ell' \geq \ell \text{ and } g_{(i,\ell')} = r \\ r, & \ell' \geq \ell \text{ and } g_{(i,\ell')} = s \\ g_{(i,\ell')}, & \text{otherwise.} \end{cases}$$

With Metropolis–Hastings acceptance probability

$$\min \left\{ 1, \frac{\mathbb{P}(g^*|A)}{\mathbb{P}(g|A)} \right\}, \quad (4.30)$$

we accept the move and change the current community assignment to the proposed community assignment g^* . Otherwise, we reject the move and keep the previous community assignment g . In either case, we then proceed with the previously-discussed Gibbs-sampling procedure. If we do not accept the move, then we do not change the community assignment. By (4.26) and (4.27), the acceptance probability (4.30) is equivalent to

$$\min \left\{ 1, \frac{\mathbb{P}(A, g^*)}{\mathbb{P}(A, g)} \right\}.$$

In Section 4.4.3, we demonstrate that using these multi-node moves significantly improves the performance of both the Bazzi et al. [1] approach and our exchangeability-based approach.

We now discuss in detail how we compute $\mathbb{P}(A, g)$ for a uniform distribution on community assignments, the Bazzi et al. [1] approach, and our exchangeability-based approach. It suffices to discuss how we compute $\mathbb{P}(g)$ for each of our three choices, as

$$\mathbb{P}(A, g) = \mathbb{P}(A|g)\mathbb{P}(g)$$

from (4.26) and $\mathbb{P}(A|g)$ is specified in (4.25) as

$$\mathbb{P}(A|g) = \prod_{\ell=1}^L \prod_{1 \leq r \leq s \leq k} \frac{m_{rs}^{(\ell)}!(t_{rs}^{(\ell)} - m_{rs}^{(\ell)})!}{(t_{rs}^{(\ell)} + 1)!},$$

which is relatively simple to compute. To compute $\mathbb{P}(g)$, it is sufficient to compute an expression that is directly proportional to it because (4.29) remains unchanged when all terms are multiplied by the same constant.

We begin by discussing the case of a uniform distribution on community assignments (see Section 4.2.2.1). From (4.9), we have

$$\mathbb{P}(g) = \frac{1}{k^{nL}} \propto 1.$$

That is, for this case, we do not need to consider $\mathbb{P}(g)$ when computing $\mathbb{P}(A, g)$.

We now consider the Bazzi et al. discrete-time Markov-process approach (see Section 4.2.3.2). In Appendix B.1, we derive the expression

$$\mathbb{P}(g^{(\ell)}|g^{(\ell-1)}) = \int \cdots \int_{[0,1] \times \Delta^{n-1}} \prod_{i=1}^n \left(\alpha_{\ell} \delta_{g^{(\ell)}, g^{(\ell-1)}} + (1 - \alpha_{\ell}) \kappa_{g^{(\ell)}}^{(\ell)} \right) d\mu(\alpha_{\ell}, \kappa^{(\ell)}), \quad (4.31)$$

where μ is the product measure of a uniform measure on $[0, 1]$ and a uniform measure on $\Delta^{k-1} \cap [0, 1]^k$. Recall from (4.8) that $\Delta^{k-1} = \{(v_1, \dots, v_k) | \sum_{i=1}^k v_i = 1\} \cap [0, 1]^k$. There does not exist a simple closed form for the integral (4.31), so we approximate it

using Monte-Carlo integration with 1000 sample points. For community assignments g and g' that differ only in the community assignment of the node-layer (k, ℓ) , we have

$$\mathbb{P}(g_{(m)}|g_{(m-1)}) = \mathbb{P}(g'_{(m)}|g'_{(m-1)})$$

for any $m \notin \{\ell, \ell + 1\}$. In Appendix B.1, we derive

$$\mathbb{P}(g) = \mathbb{P}(g_{(1)}) \prod_{\ell=2}^L \mathbb{P}(g_{(\ell)}|g_{(\ell-1)}). \quad (4.32)$$

From (4.32), varying the community assignment of node-layer (k, ℓ) with all other community assignments fixed yields

$$\mathbb{P}(g) \propto \begin{cases} \mathbb{P}(g_{(1)})\mathbb{P}(g_{(2)}|g_{(1)}), & \ell = 1 \\ \mathbb{P}(g_{(\ell+1)}|g_{(\ell)})\mathbb{P}(g_{(\ell)}|g_{(\ell-1)}), & 2 \leq \ell \leq L - 1 \\ \mathbb{P}(g_{(L)}|g_{(L-1)}), & \ell = L, \end{cases}$$

which reduces the number of times that one needs to use Monte-Carlo integration to compute the integral (4.31) from $L - 1$ to at most 2.

Finally, we discuss our exchangeability-based approach (see Section 4.2.3.2). In Appendix B.2, we derive

$$\mathbb{P}(g_{(\ell)}|g_{(\ell-1)}) = \prod_{r=1}^k \left[\frac{1}{\binom{n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)} + k - 2}{n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)}} \binom{n_r(g_{(\ell-1)})}{c_{r,1}, \dots, c_{r,k}}} \times J(n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)}, n_r(g_{(\ell-1)})) \right], \quad (4.33)$$

where

$$J(k_1, k_2) = \int_0^1 x^{k_1} \frac{x - 1}{x^{k_2+1} - 1} dx.$$

To minimize computational cost we precompute

$$J(k_1, k_2) = \int_0^1 x^{k_1} \frac{x - 1}{x^{k_2+1} - 1} dx$$

for all k_1 and k_2 such that $0 \leq k_1 \leq k_2 \leq n$ using the procedure in Section B.2.1. This precomputation allows us to avoid needing to repeatedly recompute $J(k_1, k_2)$ when computing (4.33).² For g and g' that differ only in the community assignment of node-layer (k, ℓ) , recall that

$$\mathbb{P}(g_{(m)}|g_{(m-1)}) = \mathbb{P}(g'_{(m)}|g'_{(m-1)}) \quad (4.34)$$

for any $m \notin \{\ell, \ell + 1\}$. In Appendix B.2, we derive

$$\mathbb{P}(g) = \mathbb{P}(g_{(1)}) \prod_{\ell=2}^L \mathbb{P}(g_{(\ell)}|g_{(\ell-1)}). \quad (4.35)$$

Therefore, if we vary the community assignment of node-layer (k, ℓ) and fix all other community assignments, we obtain

$$\mathbb{P}(g) \propto \begin{cases} \mathbb{P}(g_{(1)})\mathbb{P}(g_{(2)}|g_{(1)}), & \ell = 1 \\ \mathbb{P}(g_{(\ell+1)}|g_{(\ell)})\mathbb{P}(g_{(\ell)}|g_{(\ell-1)}), & 2 \leq \ell \leq L - 1 \\ \mathbb{P}(g_{(L)}|g_{(L-1)}), & \ell = L, \end{cases} \quad (4.36)$$

which reduces the number of times that we need to compute (4.33) from $L - 1$ to at most 2.

To help ensure that any observed differences in performance are not due to our choice of generative model $\mathbb{P}(A|g)$ or our choice of posterior-sampling algorithm, we

²If n is sufficiently large, the integral $J(k_1, k_2)$ can become very small, which causes finite-precision issues and thereby leads to inaccurate results when computing $J(k_1, k_2)$ using (B.14). To mitigate this problem, we use the approximation

$$\frac{J(k_1 + 1, k_2)}{J(k_1, k_2)} \approx 1$$

for large k_1 and k_2 . In particular, for fixed k_2 , we set the computed values of $J(k_1, k_2)$ to $\exp(-16)$ for all $k_1 \geq k'_1$, where k'_1 is the smallest k'_1 such that $J(k'_1, k_2) < \exp(-16)$.

also compare the performance of the three methods to the performance of another method that uses both a different generative model $\mathbb{P}(A|g)$ and a different approach to approximately sample from the posterior $\mathbb{P}(g|A)$. For this comparison, we consider the method of Yang et al. [88]. As with the other three approaches (see (4.24)), $\mathbb{P}(A|g, \omega)$ consists of an independent SBM for each layer (see (4.24)). Namely,

$$\mathbb{P}(A|g, \omega) = \prod_{\ell=1}^L \mathbb{P}(A^{(\ell)}|g^{(\ell)}, \omega^{(\ell)}) = \prod_{\ell=1}^L \prod_{1 \leq i < j \leq n} \left(\omega_{g^{(\ell)}(i), g^{(\ell)}(j)}^{(\ell)} \right)^{A_{ij}^{(\ell)}} \left(1 - \omega_{g^{(\ell)}(i), g^{(\ell)}(j)}^{(\ell)} \right)^{1 - A_{ij}^{(\ell)}}.$$

However, the Yang et al. approach uses a different prior distribution $\mathbb{P}(\omega)$ than the one in the other three approaches. First, the Yang et al. approach assumes that $\omega_{rs} := \omega_{rs}^{(1)} = \dots = \omega_{rs}^{(L)}$ (i.e., the SBM parameters do not change between layers) and that $\omega_{rs} = \omega_{sr}$. Second, Yang et al. used

$$\omega_{rs} \sim \text{Beta}(\alpha_{rs}, \beta_{rs})$$

for all $r \geq s$, where α_{rs} and β_{rs} are user-provided parameters. In our implementation of their approach, we let $\alpha_{rs} = \beta_{rs} = 1$ for all $r \geq s$, which implies that

$$\omega_{rs} \sim \text{Uniform}(0, 1)$$

for all $r \geq s$. As we previously discussed Yang et al.'s approach for sampling from the community-evolution probability distribution $\mathbb{P}(g|\omega)$ in Section 4.2.3.2, we do not discuss it again here.

To sample from the posterior distribution $\mathbb{P}(g|A)$, Yang et al. [88] used Gibbs sampling with simulated annealing. This choice differs from our Gibbs-sampling approach because it uses the target distributions

$$\exp\{\log \mathbb{P}(g|A)/T_m\},$$

where T_m is the m th temperature, which is a parameter that impacts the sensitivity of the sampler to the distribution, and \log denotes the base- e logarithm. When T_m is large, the sampler is sensitive to coarser variations; when T_m is small, the sampler is sensitive to finer variations. In our implementation, we use the temperature sequence $\{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$ that was employed by Yang et al. [88]. For each temperature T_m , we perform N_m Gibbs-sampling iterations. We also use the iteration-number sequence $\{20, 10, 10, 10, 10, 10, 10, 5, 5, 5\}$ of Yang et al. Our implementations of these methods are available at <https://github.com/tfaust0196/TemporalCommunityComparison>.

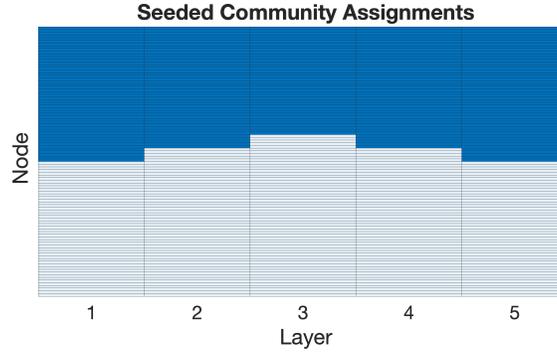
4.4.2 Setup of our Comparison

To compare the behavior of the four approaches on networks with different community sizes, we generate several networks with known community structure and different community sizes. We begin by considering $k = 2$ communities in temporal networks with $n = 100$ nodes and $L = 5$ layers. We then choose the parameters $\omega \in [0, 1]^{k \times k}$ (which encode the strength of the community structure) and $q \in \{0, \dots, 100\}$ (which is the size of community 1). We use the parameter q to assign the seeded community structure g as follows. For each layer $\ell \in \{1, \dots, L\}$, we set $g_{(i,\ell)} = 1$ for nodes $i \in \{1, \dots, q + \tau_\ell\}$ and $g_{(i,\ell)} = 2$ for all other nodes. The quantity $\tau = (\tau_1, \dots, \tau_L)$ is a vector of “offsets” for each layer that we use to avoid having the same seeded community structure for each layer. In Figure 4.11, we show examples of the seeded community structure for $q = 50$ and $q = 90$, with $\tau = (0, -5, -10, -5, 0)$ in each case. We use g to generate the networks $A^{(\ell)}$ via

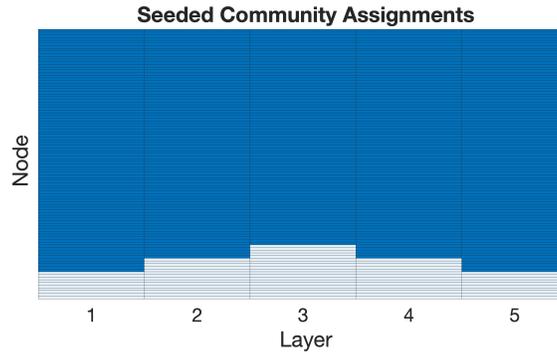
independent SBMs with parameters

$$\omega^{(\ell)} = \begin{pmatrix} 0.25 & 0.1 \\ 0.1 & 0.25 \end{pmatrix}$$

for each layer $\ell \in \{1, \dots, L\}$.



(a) $q = 50$



(b) $q = 90$

Figure 4.11: Heat maps of the seeded community structure that we use to generate the adjacency structure A for (a) community-1 size $q = 50$ and (b) community-1 size $q = 90$. Each rectangle in a heat map corresponds to one node-layer (i, ℓ) . Dark blue rectangles signify the community assignment $g_{(i, \ell)} = 1$, and light blue rectangles signify the community assignment $g_{(i, \ell)} = 2$.

In our experiments, we consider $q \in \{50, 60, 70, 80, 90\}$. For each choice of q , we run 500 instantiations of each approach and record the inferred community assignments g' of each instantiation of each approach. To quantitatively measure the performance of each approach, we compute normalized mutual information (NMI) $\text{NMI}(g'; g)$ [14], which is a commonly employed similarity measure for analyzing the performance of a classification, between the inferred community structure g' and the seeded community structure g for each instantiation and approach. The formula for the NMI is

$$\text{NMI}(g'; g) = \frac{I_0(g'; g)}{\frac{1}{2}[I_0(g; g) + I_0(g'; g')]}, \quad (4.37)$$

where

$$\begin{aligned} I_0(g'; g) &= nL \sum_{1 \leq r, s \leq k} p_{rs}^{(gg')} \log \left(\frac{p_{rs}^{(gg')}}{p_r^{(g')} p_s^{(g)}} \right), \\ p_{rs}^{(gg')} &= \frac{1}{nL} \sum_{i=1}^n \sum_{\ell=1}^L \delta_{g'_{(i,\ell)}, r} \delta_{g_{(i,\ell)}, s}, \\ p_r^{(g)} &= \frac{1}{nL} \sum_{i=1}^n \sum_{\ell=1}^L \delta_{g_{(i,\ell)}, r}. \end{aligned}$$

The NMI quantifies the similarity between the inferred community structure g' and the seeded community structure g . The maximum NMI value of 1 occurs when g' and g coincide up to a permutation of the community labels.

4.4.3 Results

We begin by comparing the performances of the Bazzi et al. [1] approach and our exchangeability-based approach with and without multi-node moves (see Section 4.4.1). We set the probability of a multi-node move to 3×10^{-3} . In Figure 4.12, we plot the mean of $\text{NMI}(g'; g)$ over each of the posterior samples g' for community-1

sizes $q \in \{50, 60, 70, 80, 90\}$.

We obtain larger mean NMI values for both the Bazzi et al. approach and our exchangeability-based approach when we use multi-node moves, indicating that they are less likely to become stuck at local extrema (of the type in Figure 4.10) than when we do not include multi-node moves. We use hypothesis testing to make this observation statistically rigorous. The distributions of the NMI values are not normal distributions, so we compute the p -values for a one-sided Mann–Whitney U test [15] for each layer ℓ , community-1 size q , and community-structure strength ω . For both approaches, we see that the p -values are significant for all values of q except $q = 50$. Therefore, we conclude that employing multi-node moves improves the performance of both our exchangeability-based approach and the Bazzi et al. approach.

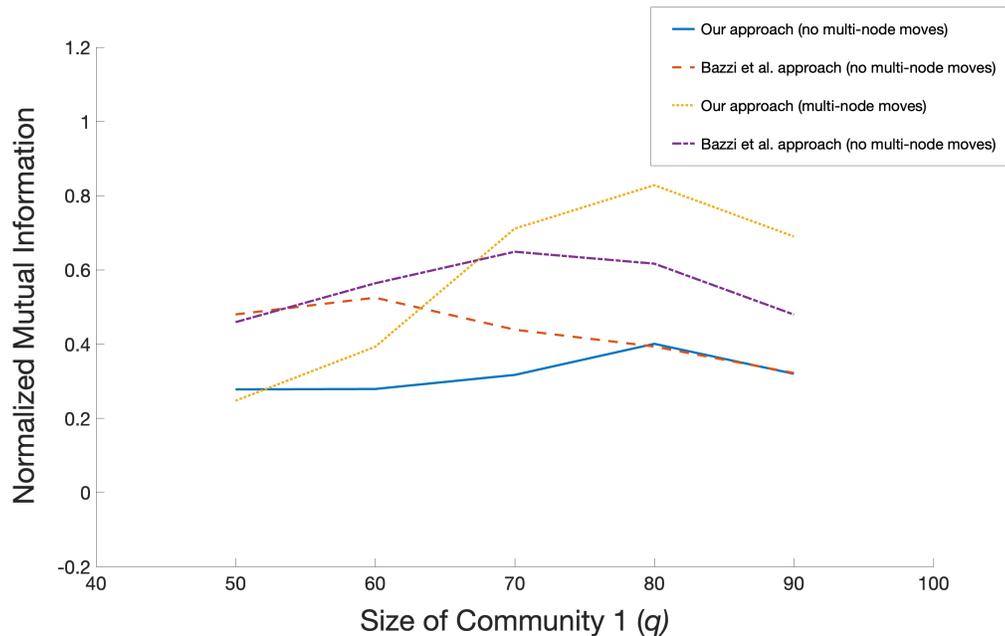


Figure 4.12: The mean NMI for our exchangeability-based approach and the Bazzi et al. approach with and without multi-node moves for several values of the community-1 size q .

| | $q = 50$ | $q = 60$ | $q = 70$ | $q = 80$ | $q = 90$ |
|-----------------|----------|----------|----------------------|----------------------|----------------------|
| $\omega = 0.28$ | 0.56 | 0.017 | 0.45 | 9.1×10^{-5} | 0.96 |
| $\omega = 0.25$ | 0.12 | 0.063 | 9.4×10^{-6} | 3.3×10^{-6} | 3.4×10^{-7} |

Table 4.4: The p -values for our comparison of our exchangeability-based approach with and without multi-node moves.

| | $q = 50$ | $q = 60$ | $q = 70$ | $q = 80$ | $q = 90$ |
|-----------------|----------------------|----------------------|----------------------|-----------------------|-------------------------|
| $\omega = 0.25$ | 0.015 | 6.5×10^{-4} | 3.3×10^{-6} | 3.1×10^{-16} | 7.8×10^{-5} |
| $\omega = 0.28$ | 2.4×10^{-3} | 0.40 | 8.9×10^{-9} | 5.9×10^{-15} | 2.5000×10^{-6} |

Table 4.5: The p -values for our comparison of the Bazzi et al. [1] approach with and without multi-node moves.

We also compare the performance of the four approaches — a uniform distribution on community assignments, the Yang et al. [88] and Bazzi et al. [1] discrete-time Markov-process approaches, and our exchangeability-based approaches — when we incorporate multi-mode moves. We again consider community-1 sizes $q \in \{50, 60, 70, 80, 90\}$ and plot the overall-network NMI for each of the four approaches in Figure 4.13.

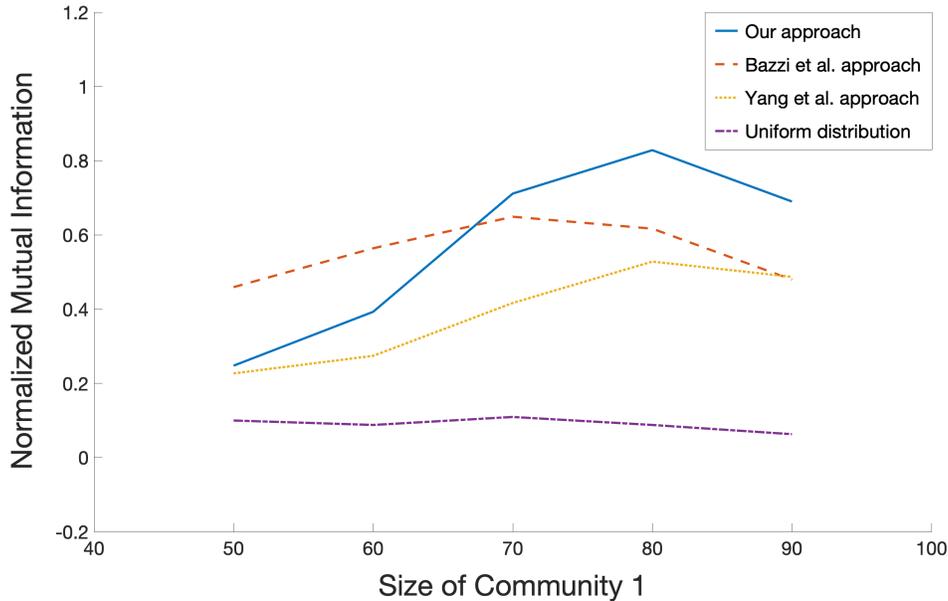


Figure 4.13: The mean NMI for each of the four examined approaches for generating community assignments for various choices of 1-community sizes q .

Our exchangeability-based approach performs worse than the Bazzi et al. approach for community-1 sizes of $q = 50$ and $q = 60$. However, our approach outperforms the Bazzi et al. approach for $q = 70$, $q = 80$, and $q = 90$. Our approach outperforms the Yang et al. approach for all values of q . Using Mann–Whitney U tests, we see that these results are significant (with p -value $p < 0.01$) for all but one value of q (this value is $q = 80$ for the Yang et al. method) when our exchangeability-based approach performs better than the other methods.

In Figures 4.12 and 4.13, the mean NMIs for each method are smaller for community-1 size $q = 90$ than for $q = 80$. We believe that this is an artifact of our choice to use NMI to measure the similarity between the inferred community structure and

the seeded community structure.³ When we use the number of correct community assignments to measure such similarity, we find that the mean numbers of correct community assignments for community-1 sizes $q = 80$ and $q = 90$ are almost identical for each method.

4.5 Conclusions and Discussion

When applying statistical inference to community detection, it is important important to use a generative model that is based on realistic assumptions [66]. One such assumption is that the generative model is not biased against communities with large or small numbers of nodes. In this chapter, we showed that many statistical-inference models that generate community assignments via either a uniform distribution on-community assignments or discrete-time Markov processes are biased against generating communities with large or small numbers of nodes. We then formulated a generative model with an exchangeability-based community-assignment approach that mitigates this bias. We observed in tests on synthetic networks with small and large community sizes that our generative model outperforms existing generative models at statistical inference. We focused in this chapter on analyzing the effect of group-size biases in the setting of community structure. Similar group-size assumptions also arise in statistical-inference methods for identifying other mesoscale structures. For example, in Chapter 5, we show that using discrete-time Markov-process models for detecting core-periphery structure causes associated statistical-inference methods to underestimate the number of groups in networks.

³For a detailed discussion of biases in NMI, see [36].

There are many viable ways to build on our work. First, we considered the performance of a small number of representative approaches, and it is certainly worthwhile to also study other methods (e.g., [35]) with similar community-assignment probability distributions to those in Sections 4.2.3.1 and 4.2.3.2. Second, in our exchangeability-based approach (see Section 4.2.3.3), we assumed that the community assignments for a given layer depend only on those in the previous layer. Relaxing this assumption and allowing community assignments to depend on additional previous layers may lead to improved community-detection performance. Third, it is beneficial to derive bounds on the amount of localization of the community-size distributions for both our approach and other approaches. In contrast to the situation for monolayer networks, where community-size distributions tend to have a relatively simple form, there often is not a simple closed-form expression for the single-layer community-size distributions in temporal networks. Consequently, we used numerical simulations to examine these community-size distributions instead of obtaining analytical results (such as Conjecture 1) about the single-layer community-size distributions in the limit of infinitely many layers. Deriving such analytical results can provide important insights into the behavior of community-assignment approaches, which in turn can lead to the development of better-performing and more efficient methods.

CHAPTER 5

Detection of Hierarchical Core–Periphery Structure in Temporal Networks

5.1 Introduction

As we discussed in Section 2.1, it is common in network analysis to study various types of mesoscale structures [56]. In this chapter, we examine core–periphery structure, in which well-connected nodes (so-called “core” nodes) are connected densely to each other and potentially densely connected to other nodes (so-called “peripheral” nodes), which are connected sparsely to other nodes [12, 71, 87]. Core–periphery structure has been studied in many time-independent networks in the past few decades, leading to insights into topics such as social networks [26, 84], academic networks [16, 86], economic networks [76, 85], transportation networks [45], and many other areas. In many situations — including in the analysis of the spread of diseases through face-to-face contacts [21], transportation systems [53], and legislation cosponsorships [46, 55] — it is important to consider relationships and/or interactions that change with time. In such situations, one can study a temporal network [30–32]. In a temporal network, the entities in the network and/or the ties between them can change with time. As we discussed in Section 2.2, one can study such a temporal network using a multilayer-network representation [40]. Many approaches have been

developed to study community structure in time-dependent networks [23, 68, 71, 89] (as we mentioned in Section 4.1), but there are only a few studies of core–periphery structure in multilayer networks. For example, Bergermann et al. used spectral methods to identify core–periphery structure in multilayer networks [4, 5], Nie et al. used a rich-club approach to identify core–periphery structure in multiplex networks [58], and Hashemi and Behrouz generalized the concept of k -cores to multiplex networks [27].

In the present chapter, we study a hierarchical notion of core–periphery structure in temporal networks (which we represent as multilayer networks). We generalize the hierarchical core–periphery structure of Polanco and Newman [68] from ordinary networks (i.e., graphs) to temporal networks. This notion of hierarchical core–periphery structure encompasses a rich variety of possible mesoscale structures, including ones that are not nested. In a nested (i.e., onion-like) core–periphery structure [23], each node of a network is part of exactly one group, with higher-numbered groups signifying nodes that are deeper into a core. In such a setting, the probability that there is an edge between two nodes depends on the lower of the two group assignments of those nodes. By contrast, in the hierarchical core–periphery structure in [68], each node of a network can be in several groups simultaneously. One determines the edge probability between two nodes using the highest-numbered group that includes both nodes. This hierarchical formulation allows the generation of networks with nested structure, tree-like structure (where any two groups must either be disjoint or have one be a strict subset of the other), and general non-nested mesoscale structures (where the group assignments of nodes do not have to be nested in any way).¹

¹For illustrations of the types of structures that one can generate using this hierarchical core–periphery structure, see Figure 1 of [68].

To identify hierarchical core–periphery structure in temporal networks, we use statistical inference. This makes our approach more computationally costly than spectral methods to detect core–periphery structure, but it also yields several important benefits [66]. For example, the use of statistical inference guarantees that posterior-sampling methods yield samples precisely from the specified posterior distribution in the limit as the number of nodes of a network goes to infinity. Indeed, in this chapter, we prove in this limit that the stable distribution of a slightly-modified version of the Markov-chain Monte Carlo (MCMC) approach that we use for statistical inference is the same as the desired posterior distribution. Additionally, in contrast to many other approaches, such as non-inferential optimization approaches, statistical-inference methods avoid identifying mesoscale structures in completely random networks, which (by construction) arise from models without any mesoscale structure [66].

After we discuss our approach and give reasons for various choices that we make, we use it to study two real-world temporal networks. These two example networks are a network of ties between terrorist organizations in the Indian states of Jammu and Kashmir [77] and a network of co-appearances in the *Luke Gospel* [29]. For each example, we show that the identified core–periphery structure is plausible.

This chapter proceeds as follows. In Section 5.2, we present the details of the generative model that we use for statistical inference of hierarchical core–periphery structure. In Section 5.3, we present the MCMC approach that we use to identify such structure. In Section 5.4, we discuss the consequences of the choice of generative model on the performance of our MCMC approach. In Section 5.5, we use our statistical-inference approach to identify hierarchical core–periphery structure in two real-world temporal networks. In Section 5.6, we conclude and discuss several future

directions.

5.2 A Hierarchical Generative Model for Core–Periphery Structure in Temporal Networks

As in [68], we think of core–periphery structure as a hierarchy with groups of nodes that do not need to be nested. We allow each node-layer to be a member of each of k groups, which we label with the indices $0, 1, \dots, k - 1$. In our hierarchical core–periphery setting, the probability of an edge between two nodes is determined by the highest-numbered group that includes both nodes [68]. By contrast, in a nested core–periphery setting [23], researchers typically require each node of a network to be a member of exactly one group with indices in $0, 1, \dots, k - 1$. In the nested core–periphery setting, the probability that there is an edge between two nodes depends on the lower of the two group assignments of the nodes.

To ensure that the hierarchical structure has a “base level”, we assume (as in [68]) that every node is in group 0. We define a set of indicator variables $g_{(i,\ell)}^r$, where $g_{(i,\ell)}^r = 1$ if node-layer (i, ℓ) is in group r and $g_{(i,\ell)}^r = 0$ otherwise.

To generate a temporal network A given group assignments g , we use the model in [68] independently for each layer. For each pair of node-layers $(i, \ell), (j, \ell)$ in the same layer, we place an edge between them independently with probability $\omega_{h((i,\ell),(j,\ell))}^{(\ell)} \in [0, 1]$, where $h((i, \ell), (j, \ell))$ is the highest common group that includes both nodes (i.e., the largest r such that $g_{(i,\ell)}^r = g_{(j,\ell)}^r = 1$). Applying the argument in equations

(1)–(3) from Section II of [68], we then have

$$P(A|\omega, k, g) = \prod_{\ell=1}^L \prod_{r=0}^{k-1} \left[(\omega_r^{(\ell)})^{m_r^{(\ell)}} (1 - \omega_r^{(\ell)})^{t_r^{(\ell)} - m_r^{(\ell)}} \right],$$

where

$$t_r^{(\ell)} = \sum_{1 \leq i < j \leq n} \delta_{r, h((i, \ell), (j, \ell))}$$

is the number of node-layer pairs $(i, \ell), (j, \ell)$ in layer ℓ that have highest common group r and

$$m_r^{(\ell)} = \sum_{1 \leq i < j \leq n} A_{ij}^{(\ell)} \delta_{r, h((i, \ell), (j, \ell))}$$

is the number of such pairs that are adjacent to each other.

To avoid a dependence on the parameters $\omega_r^{(\ell)}$, we follow the approach in [68]. For all r and ℓ , we assume that there is a uniform prior $\mathbb{P}(\omega_r^{(\ell)}) = 1$. The computations in equation (4) of [68] show that marginalizing according to these choices of prior distributions yields

$$\mathbb{P}(A|k, g) = \prod_{\ell=1}^L \prod_{r=0}^{k-1} \frac{m_r^{(\ell)}! (t_r^{(\ell)} - m_r^{(\ell)})!}{(t_r^{(\ell)} + 1)!}. \quad (5.1)$$

We break the discussion of our model into two parts. In Section 5.2.1, we consider a variant of our model in which we fix the number k of groups. In Section 5.2.2, we remove this assumption and introduce our main model, in which the number k of groups is unspecified.

5.2.1 Fixed Number of Groups

Suppose that we fix the number k of groups. To perform Bayesian inference on a temporal network A to yield a posterior distribution $\mathbb{P}(g|A)$, we need a prior distribution

$\mathbb{P}(g|k)$ for the group assignments of nodes. There are many possible choices, such as a uniform distribution over group assignments (i.e., $\mathbb{P}(g|k) \propto 1$) for such a prior distribution. However, we showed in Section 4.3 in the context of community structure that many common choices for this prior distribution, such as a uniform distribution and generating g via a discrete-time Markov process, make it prohibitively unlikely to obtain large or small groups, which typically is an undesirable situation. This, in turn, impacts the accuracy of statistical inference of small and large groups.

To mitigate the group-size bias, we determine group assignments using a similar approach to the exchangeability-based community-evolution approach in Section 4.2.3.3. We start by selecting the group sizes for the first layer uniformly at random. Given these group sizes, we choose the group assignments of nodes uniformly at random from all group assignments with the chosen group sizes. In other words, we choose the group assignments $g_{(1)}$ of the nodes in the first layer according to the probability distribution

$$P(g_{(1)}|k) = \prod_{r=1}^{k-1} \frac{(n_1(g_{(1)}^r))! \times (n - n_1(g_{(1)}^r))!}{(n+1)!}, \quad (5.2)$$

where $n_1(g_{(1)}^r)$ is the number of nodes in layer 1 that have group- r indicator variables of 1.

To generate each layer beyond the first one, we use the group assignments from

the previous layer to generate those in the next layer via the probability distribution

$$\mathbb{P}(g_{(\ell)}^r | g_{(\ell-1)}^r) = \prod_{s=0}^1 \left[\frac{1}{\binom{n_s(g_{(\ell-1)}^r)}{c_{r;ss}^{(\ell)}}} \times \int_0^1 p_{r;s,\ell}^{n_s(g_{(\ell-1)}^r) - c_{r;ss}^{(\ell)}} \frac{p_{r;s,\ell} - 1}{p_{r;s,\ell}^{n_s(g_{(\ell-1)}^r) + 1} - 1} dp_{r;s,\ell} \right], \quad (5.3)$$

where $g_{(\ell)}^r$ is the set of indicator variables of group r for all nodes in layer ℓ , the quantity $n_s(g_{(\ell-1)}^r)$ is the number of nodes in layer $\ell - 1$ that have group- r indicator variables of s , the quantity $c_{r;ss}^{(\ell)}$ is the number of nodes i such that $g_{(i,\ell-1)}^r = s$ and $g_{(i,\ell)}^r = s$, and where each $p_{r;s,\ell}$ is the parameter of an independent geometric distribution $\overline{\text{Geom}}(p_{r;s,\ell})$ (see equation (4.16)).² For notational convenience, we define

$$J(k_1, k_2) = \int_0^1 x^{k_1} \frac{x - 1}{x^{k_2 + 1} - 1} dx,$$

as in (B.14). Using this notation, we can write (5.3) as

$$\mathbb{P}(g_{(\ell)}^r | g_{(\ell-1)}^r) = \prod_{s=0}^1 \left[\frac{1}{\binom{n_s(g_{(\ell-1)}^r)}{c_{r;ss}^{(\ell)}}} \times J(n_s(g_{(\ell-1)}^r) - c_{r;ss}^{(\ell)}, n_s(g_{(\ell-1)}^r)) \right]. \quad (5.4)$$

Finally, we set

$$\mathbb{P}(g|k) = \mathbb{P}(g_{(1)}) \prod_{\ell=2}^L \prod_{r=1}^{k-1} \mathbb{P}(g_{(\ell)}^r | g_{(\ell-1)}^r), \quad (5.5)$$

²For a full discussion of the approach that we use to generate $g_{(\ell)}^r$ from $g_{(\ell-1)}^r$, see Section 4.2.3.3 and Appendix B.2.

and we thereby obtain the prior distribution $\mathbb{P}(g|k)$. Using the notation

$$F(g|k) = \prod_{\ell=2}^L \prod_{r=1}^{k-1} \mathbb{P}(g_{(\ell)}^r | g_{(\ell-1)}^r), \quad (5.6)$$

we write (5.5) as

$$\mathbb{P}(g|k) = \mathbb{P}(g_{(1)}|k)F(g|k). \quad (5.7)$$

By Bayes' rule,

$$\mathbb{P}(g|A, k) = \frac{\mathbb{P}(A|g, k)\mathbb{P}(g|k)}{\mathbb{P}(A|k)}. \quad (5.8)$$

Because we have expressions for $\mathbb{P}(A|g, k)$ and $\mathbb{P}(g|k)$ (and because $\mathbb{P}(A|k) \propto 1$), we can use (5.8) to sample from the posterior distribution $\mathbb{P}(g|A, k)$.

5.2.2 Main Model: Unspecified Number of Groups

In our main model for hierarchical core–periphery structure, we suppose that the number k of groups is unspecified. For this case, as in [68], we use a Poisson prior distribution on k with mean 1. Namely,

$$\mathbb{P}(k) = \frac{e^{-1}}{(k-1)!}. \quad (5.9)$$

We then have

$$\mathbb{P}(g, k|A) = \mathbb{P}(k)\mathbb{P}(g|A, k). \quad (5.10)$$

Using the expression (5.10), in Section 5.3, we derive our main MCMC algorithm for sampling from the posterior distribution (5.10) and prove that the stationary distribution of a slightly-modified version of this algorithm is the same as (5.10). In Section 5.3.1.3, we discuss the required modifications of our main MCMC algorithm.

5.3 Statistical-Inference Approach

To sample from the posterior distribution (5.10), we use an MCMC method that is similar to the ones in [68].

In Sections 5.3.1, 5.3.2, and 5.3.3, we introduce our MCMC method (see Algorithm 4). To prove that the stable distribution of a slightly-modified version of Algorithm 4 is the same as the desired posterior distribution (5.10), we introduce versions of Algorithm 4 that do not include certain types of moves and prove intermediate results involving their stable distributions. In Section 5.3.4, we consider Algorithm 6, which is an MCMC algorithm for a fixed number of groups. In Section 5.3.5, we consider Algorithm 7, which is similar to Algorithm 4 but (1) allows the number of groups to vary and (2) does not allow one class of moves that we consider in Algorithm 4. In Section 5.3.6, we prove that the stable distribution of a slightly-modified version of Algorithm 4 is the same as the desired posterior distribution (5.10).

5.3.1 Our Main MCMC Algorithm

We begin by presenting the three types of MCMC moves in Algorithm 4.

5.3.1.1 Standard Moves

The first type of MCMC move, which we call a *standard move* (see Algorithm 1), is a move that adds a node-layer to a group, removes a node-layer from a group, or removes an empty group (i.e., a group that has no nodes). In this type of move, we first uniformly randomly choose a group r and a layer ℓ . The specific move that we

propose is different for the first layer $\ell = 1$ and the other layers $\ell \geq 2$.

First consider layer $\ell = 1$. With probability $1/2$, we propose a move that adds a node-layer in layer 1 to group r . With probability $1/2$, we propose a move that removes a node-layer in layer 1 from group r . If we choose to add a node-layer to a group, we select a node-layer $(i, 1)$ in layer 1 that is not in group r (i.e., $g_{(i,1)}^r = 0$) uniformly at random from the set of all node-layers $(i, 1)$ in layer 1 that are not in group r . If all node-layers in layer 1 are already in group r , then we do nothing. Analogously, if we choose to remove a node-layer from a group, we select a node-layer $(i, 1)$ in layer 1 that is currently in group r (i.e., $g_{(i,1)}^r = 1$) uniformly at random from the set of all node-layers $(i, 1)$ in layer 1 that are in group r . If no node-layers are in group r , we reduce the number k of groups by 1 by removing group r and shifting the labels of all groups above r down by 1. If group r has no node-layers in layer 1 but at least one node-layer in another layer, then we do nothing.

If layer $\ell \geq 2$, we choose a node i uniformly at random. If node-layer (i, ℓ) is in group r , then we propose a move that removes it from group r . If node-layer (i, ℓ) is not in group r , then we propose a move that adds it to group r . For $\ell \geq 2$, a proposed move cannot remove a group.

Algorithm 1 Proposal of a standard move

1. Choose a layer ℓ uniformly at random from $1, \dots, L$.
 2. Choose a group r uniformly at random from $1, \dots, k - 1$.
 3. If $\ell = 1$:
 - (a) With probability $1/2$, we choose to add a node-layer in layer 1 to group r . With probability $1/2$, we choose to remove a node-layer in layer 1 from group r .
 - (b) If we choose to add a node-layer in layer 1 to group r :
 - i. If all node-layers $(i, 1)$ in layer 1 are in group r :
 - A. Do nothing.
 - ii. Otherwise (i.e., if there is at least one node-layer in layer 1 that is not in group r):
 - A. Choose a node-layer $(i, 1)$ that is not in group r uniformly at random from all node-layers $(i, 1)$ that are not in group r .
 - B. Propose a move that adds $(i, 1)$ to group r .
-

Algorithm 1 Proposal of a standard move (continued)

3. (c) If we choose to remove a node-layer in layer 1 from group r :
 - i. If no node-layers (i, ℓ) are in group r :
 - A. Propose a move that removes group r and shifts the labels of all groups above r down by 1.
 - ii. Otherwise, if no node-layers $(i, 1)$ in layer 1 are in group r but group r has at least one node-layer from another layer:
 - A. Do nothing.
 - iii. Otherwise (i.e., if group r has at least one node-layer from layer 1):
 - A. Choose a node-layer $(i, 1)$ from group r uniformly at random from all node-layers $(i, 1)$ in group r .
 - B. Propose a move that removes $(i, 1)$ from group r .
 4. Otherwise (i.e., if $\ell \geq 2$):
 - (a) Choose a node i uniformly at random from $1, \dots, n$.
 - i. If (i, ℓ) is in group r :
 - A. Propose a move that removes (i, ℓ) from group r .
 - ii. If (i, ℓ) is not in group r :
 - A. Propose a move that adds (i, ℓ) to group r .
-

5.3.1.2 Group-Addition Moves

Our second type of MCMC move is a *group-addition move* (see Algorithm 2). This type of move increases the number k of groups by 1.

Algorithm 2 Proposal of a group-addition move

1. Choose a group r uniformly at random from $1, \dots, k$.
 2. Propose a move that increases the labels of all groups r and higher by 1, creates a new empty group with the label r , and increases the value of k by 1.
-

5.3.1.3 Multi-Node Moves

Our third type of MCMC move is a *multi-node move* (see Algorithm 3). This type of move changes more than one group assignment at a time. For a layer ℓ and two subsets \mathcal{G}_1 and \mathcal{G}_2 of the groups $\{1, \dots, k-1\}$, we propose a new group assignment g' that satisfies

$$(g')_{(i,\ell)}^r = \begin{cases} \delta_{\mathcal{G}_2,r}, & \ell = \ell \text{ and } \delta_{\mathcal{G}_1,(i,\ell)} = 1 \\ \delta_{\mathcal{G}_1,r}, & \ell = \ell \text{ and } \delta_{\mathcal{G}_2,(i,\ell)} = 1 \\ g_{(i,\ell)}^r, & \text{otherwise,} \end{cases} \quad (5.11)$$

where

$$\delta_{\mathcal{G},r} = \begin{cases} 1, & r \in \mathcal{G} \\ 0, & \text{otherwise} \end{cases}$$

and

$$\delta_{\mathcal{G},(i,\ell)} = \begin{cases} 1, & g_{(i,\ell)}^r = \delta_{\mathcal{G},r} \text{ for all } r \in \{1, \dots, k-1\} \\ 0, & \text{otherwise.} \end{cases}$$

Algorithm 3 Proposal of a multi-node move

1. Select subsets \mathcal{G}_1 and \mathcal{G}_2 uniformly at random from the groups $\{1, \dots, k-1\}$.
 2. Choose a layer ℓ uniformly at random from $1, \dots, L$.
 3. Propose a move with a new group assignment g' using (5.11).
-

As we discussed in Section 5.2.2, we need to slightly modify our main MCMC algorithm (see Algorithm 4) to attain equality of the stationary distribution and the desired posterior distribution (5.10). Specifically, we need to modify Algorithm 3 by choosing the layer ℓ uniformly from $2, \dots, L$ instead of from $1, \dots, L$. In Section 5.3.6, we prove that the stationary distribution of the modified version of Algorithm 4 coincides with the desired posterior distribution (5.10) and discuss why we nevertheless use the unmodified version of Algorithm 4 to infer core–periphery structure.

5.3.2 Acceptance Probability

In Sections 5.3.1.1, 5.3.1.2, and 5.3.1.3, we discussed three types of MCMC moves. In each step of our MCMC algorithm (see Algorithm 4), we propose a move of one of these three types. After proposing a move from g, k to g', k' , we need to determine whether to accept or reject the move. Due to the design of our MCMC algorithm, using the standard Metropolis–Hastings acceptance [70]

$$\min \left\{ 1, \frac{\mathbb{P}(A|g', k')\mathbb{P}(g'|k')}{\mathbb{P}(A|g, k)\mathbb{P}(g|k)} \right\} \quad (5.12)$$

would cause the stationary distribution of our Markov chain to differ from the desired posterior distribution (5.10). Therefore, we slightly modify (5.12) and instead use

the acceptance probability

$$\alpha(g, k \rightarrow g', k') = \min \left\{ 1, \frac{\mathbb{P}(A|g', k')F(g'|k')}{\mathbb{P}(A|g, k)F(g|k)} \right\}, \quad (5.13)$$

where we recall that $F(g|k)$ is given by (5.6).

5.3.3 Statement of our Main MCMC Algorithm

Now that we have discussed the three types of MCMC moves (see Sections 5.3.1.1, 5.3.1.2, and 5.3.1.3) and the acceptance probability (see Section 5.3.2) of a move, we now state Algorithm 4, which gives the MCMC approach that we use to identify hierarchical core–periphery structure in temporal networks.

Let $p \in [0, 1]$ denote the probability that a move is a multi-node move. We use Algorithm 4 to sample from the posterior distribution (5.10) and identify hierarchical core–periphery structure.

Algorithm 4 Main MCMC algorithm

1. Propose a move:
 - (a) With probability p , propose a multi-node move (see Algorithm 3).
 - (b) Otherwise, with probability $1 - \frac{1}{2k(n+1)}$, propose a standard move (see Algorithm 1).
 - (c) Otherwise, propose a group-addition move (see Algorithm 2).
2. Accept the proposed move from g, k to g', k' with the acceptance probability

$$\alpha(g, k \rightarrow g', k') := \min \left\{ 1, \frac{\mathbb{P}(A|g', k')F(g'|k')}{\mathbb{P}(A|g, k)F(g|k)} \right\}.$$

3. Otherwise, reject the proposed move.
-

5.3.4 MCMC Algorithm for a Fixed Number of Groups

As we discussed in the introduction of Section 5.3, we prove that the stationary distribution of a slightly-modified version of Algorithm 4 is the same as the desired posterior distribution (5.10) by proving similar results for intermediate algorithms that build up to the main result. In this section, we consider a fixed number k of groups. In this case, we use only standard moves (see Section 5.3.1.1) in our MCMC algorithm.

Because we fix k , we slightly modify our algorithm for proposing a standard move (see Algorithm 1). In this modified version of the algorithm, if we choose to remove a node-layer in layer 1 and no node-layers are in group r , we do nothing (instead of removing group r , as we did before). In Algorithm 5, we give our modified algorithm

for proposing a standard move.

Algorithm 5 Proposal of a standard move when we fix the number of groups

1. Choose a layer ℓ uniformly at random from $1, \dots, L$.
 2. Choose a group r uniformly at random from $1, \dots, k - 1$.
 3. If $\ell = 1$:
 - (a) With probability $1/2$, add a node-layer in layer 1 to group r . With probability $1/2$, remove a node-layer in layer 1 from group r .
 - (b) If we choose to add a node-layer in layer 1 to group r :
 - i. If all node-layers $(i, 1)$ in layer 1 are in group r :
 - A. Do nothing.
 - ii. Otherwise (i.e., if there is at least one node-layer in layer 1 that is not in group r):
 - A. Choose a node-layer $(i, 1)$ that is not in group r uniformly at random from all node-layers $(i, 1)$ that are not in group r .
 - B. Propose a move that adds $(i, 1)$ to group r .
-

Algorithm 5 Proposal of a standard move when we fix the number of groups (continued)

3. (c) If we choose to remove a node-layer in layer 1 from group r :
 - i. If no node-layers $(i, 1)$ in layer 1 are in group r :
 - A. Do nothing.
 - ii. Otherwise (i.e., if group r has at least one node-layer from layer 1):
 - A. Choose a node-layer $(i, 1)$ that is in group r uniformly at random from all node-layers $(i, 1)$ that are in group r .
 - B. Propose a move that removes $(i, 1)$ from group r .
 4. Otherwise (i.e., if $\ell \geq 2$):
 - (a) Choose a node $i \in \{1, \dots, n\}$.
 - i. If (i, ℓ) is in group r :
 - A. Propose a move that removes (i, ℓ) from group r .
 - ii. If (i, ℓ) is not in group r :
 - A. Propose a move that adds (i, ℓ) to group r .
-

Because we fix the number k of groups, we slightly modify the acceptance probability (5.13). The acceptance probability of a move from g to g' for this algorithm is

$$\alpha(g \rightarrow g') = \min \left\{ 1, \frac{\mathbb{P}(A|g', k)F(g'|k)}{\mathbb{P}(A|g, k)F(g|k)} \right\}. \quad (5.14)$$

In Algorithm 6, we give this algorithm for sampling from the posterior distribution (5.8).

Algorithm 6 MCMC algorithm for the case of a fixed number of groups

1. Propose a move:

(a) Propose a standard move (see Algorithm 5).

2. Accept the proposed move from g to g' with the acceptance probability

$$\alpha(g \rightarrow g') := \min \left\{ 1, \frac{\mathbb{P}(A|g', k)F(g'|k)}{\mathbb{P}(A|g, k)F(g|k)} \right\} .$$

3. Otherwise, reject the proposed move.

We now show that the equilibrium distribution of this MCMC process is the same as the desired posterior distribution $\mathbb{P}(g|A, k)$ that we stated in (5.8). To show this, it is sufficient to show that the MCMC process satisfies ergodicity and detailed balance [70]. To prove ergodicity, we need to show that one can access every state in the system from every other state using a finite sequence of moves. To prove detailed balance, we need to show that the mean rate of $g \rightarrow g'$ moves equals the mean rate of $g' \rightarrow g$ moves at equilibrium. That is, we need to verify that

$$\mathbb{P}(g|A, k)\mathbb{P}(g \rightarrow g') = \mathbb{P}(g'|A, k)\mathbb{P}(g' \rightarrow g) . \quad (5.15)$$

In Algorithm 6, ergodicity clearly holds because we can first remove all node-layers from all groups and then re-add node-layers to attain any desired group assignment g . We thus only need to prove that the MCMC process satisfies detailed balance.

To prove detailed balance, we need to verify (5.15). We write $\mathbb{P}(g \rightarrow g') = \pi(g \rightarrow g')\alpha(g \rightarrow g')$, where π is the probability of proposing a move and α is the probability

of accepting it. This implies that equation (5.15) is equivalent to

$$\frac{\mathbb{P}(g'|A, k)}{\mathbb{P}(g|A, k)} = \frac{\pi(g \rightarrow g')\alpha(g \rightarrow g')}{\pi(g' \rightarrow g)\alpha(g' \rightarrow g)}. \quad (5.16)$$

Because $\mathbb{P}(g|A, k) = \frac{\mathbb{P}(A|g, k)\mathbb{P}(g|k)}{\mathbb{P}(A|k)} = \frac{\mathbb{P}(A|g, k)F(g|k)\mathbb{P}(g_{(1)}|k)}{\mathbb{P}(A|k)}$ by (5.8) and (5.7) and $\alpha(g \rightarrow g') = \min \left\{ 1, \frac{\mathbb{P}(A|g', k)F(g'|k)}{\mathbb{P}(A|g, k)F(g|k)} \right\}$ by (5.13), equation (5.16) is equivalent to

$$\frac{\mathbb{P}(A|g', k)F(g'|k)\mathbb{P}(g'_{(1)}|k)}{\mathbb{P}(A|g, k)F(g|k)\mathbb{P}(g_{(1)}|k)} = \frac{\pi(g \rightarrow g')\mathbb{P}(A|g', k)F(g'|k)}{\pi(g' \rightarrow g)\mathbb{P}(A|g, k)F(g|k)},$$

which in turn is equivalent to

$$\frac{\mathbb{P}(g'_{(1)}|k)}{\mathbb{P}(g_{(1)}|k)} = \frac{\pi(g \rightarrow g')}{\pi(g' \rightarrow g)}. \quad (5.17)$$

We now show that (5.17) holds for a standard move $g \rightarrow g'$ that adds node-layer $(i, 1)$ to group r . First, because $\mathbb{P}(g_{(1)}|k) = \prod_{s=1}^{k-1} \frac{n_s^{(1)}!(n-n_s^{(1)})!}{(n+1)!}$, we have

$$\frac{\mathbb{P}(g'_{(1)}|k)}{\mathbb{P}(g_{(1)}|k)} = \prod_{s=1}^{k-1} \frac{n'_s{}^{(1)}!(n-n'_s{}^{(1)})!}{n_s^{(1)}!(n-n_s^{(1)})!}.$$

By adding a node-layer in layer 1 to group r , we obtain $n'_r{}^{(1)} = n_r^{(1)} + 1$ and $n'_s{}^{(1)} = n_s^{(1)}$, which then yields

$$\frac{\mathbb{P}(g'_{(1)}|k)}{\mathbb{P}(g_{(1)}|k)} = \frac{n_r^{(1)} + 1}{n - n_r^{(1)}}.$$

From Algorithms 5 and 6, the probability of proposing the move $g \rightarrow g'$ is

$$\pi(g \rightarrow g') = \frac{1}{L} \times \frac{1}{k-1} \times \frac{1}{2} \times \frac{1}{n - n_r^{(1)}}.$$

Similarly, $\pi(g' \rightarrow g) = \frac{1}{L} \times \frac{1}{k-1} \times \frac{1}{2} \times \frac{1}{n_r^{(1)}+1}$. Therefore,

$$\frac{\pi(g \rightarrow g')}{\pi(g' \rightarrow g)} = \frac{n_r^{(1)} + 1}{n - n_r^{(1)}} = \frac{\mathbb{P}(g'_{(1)}|k)}{\mathbb{P}(g_{(1)}|k)}.$$

The detailed-balance equation (5.17) is thus satisfied in this instance.

We now verify (5.17) for a standard move $g \rightarrow g'$ that adds node-layer (i, ℓ) to group r when $\ell \geq 2$. First, $n_r^{(1)} = n_r^{(1)}$ for all r (because adding a node-layer in layer $\ell \geq 2$ does not affect the group sizes in layer 1). This yields

$$\frac{\mathbb{P}(g'_{(1)}|k)}{\mathbb{P}(g_{(1)}|k)} = 1.$$

From Algorithms 5 and 6, the probability of proposing the move $g \rightarrow g'$ is

$$\pi(g \rightarrow g') = \frac{1}{L} \times \frac{1}{k-1} \times \frac{1}{n}.$$

Similarly, $\pi(g' \rightarrow g) = \frac{1}{L} \times \frac{1}{k-1} \times \frac{1}{n}$, so

$$\frac{\pi(g \rightarrow g')}{\pi(g' \rightarrow g)} = 1 = \frac{\mathbb{P}(g'_{(1)}|k)}{\mathbb{P}(g_{(1)}|k)}.$$

Consequently, the detailed-balance equation (5.17) is satisfied.

We omit the proofs of (5.17) for the cases where $g \rightarrow g'$ removes a node-layer in layer $\ell \geq 1$, as they are extremely similar to the arguments above. Now that we have verified both ergodicity and detailed balance, we see that the equilibrium distribution of Algorithm 6 is the same as the desired posterior distribution (5.8).

5.3.5 Intermediate Algorithm for a Variable Number of Groups

We now allow the number k of groups to vary, and we consider an MCMC algorithm for sampling from the desired posterior distribution (5.10). In contrast to our main algorithm (see Algorithm 4), this algorithm does not use multi-node moves (see Section 5.3.1.3). The number of groups is allowed to vary, so we use our main algorithms for proposing a standard move (see Algorithm 1) and for proposing a

group-addition move (see Algorithm 2). We also again use the acceptance probability

$$\alpha(g, k \rightarrow g', k') = \min \left\{ 1, \frac{\mathbb{P}(A|g', k')F(g'|k')}{\mathbb{P}(A|g, k)F(g|k)} \right\}$$

that we specified previously in (5.13). In Algorithm 7, we give our MCMC algorithm that samples from the desired posterior distribution (5.10) without using multi-node moves.

Algorithm 7 MCMC algorithm with no multi-node moves for the case of a variable number of groups

1. Propose a move:
 - (a) With probability $1 - \frac{1}{2k(n+1)}$, propose a standard move (see Algorithm 1).
 - (b) Otherwise, propose a group-addition move (see Algorithm 2).
2. Accept the proposed move from g, k to g', k' with acceptance probability

$$\alpha(g, k \rightarrow g', k') := \min \left\{ 1, \frac{\mathbb{P}(A|g', k')F(g'|k')}{\mathbb{P}(A|g, k)F(g|k)} \right\}.$$

3. Otherwise, reject the proposed move.
-

Algorithm 7 is the same as Algorithm 4 except that it does not allow multi-node moves.

We now show that the equilibrium distribution of this MCMC process is the same as the desired posterior distribution $\mathbb{P}(g, k|A)$ (see (5.10)). As in the fixed- k case, ergodicity clearly holds for this MCMC procedure. Therefore, it suffices to prove detailed balance. However, in contrast to the fixed- k case, detailed balance holds only in the limit $n \rightarrow \infty$. Following similar logic to the logic for (5.15)–(5.17) for the

fixed- k case, to prove detailed balance for the variable- k case, we need to show that

$$\frac{\mathbb{P}(g'_{(1)}, k')}{\mathbb{P}(g_{(1)}, k)} = \frac{\pi(g, k \rightarrow g', k')}{\pi(g', k' \rightarrow g, k)}. \quad (5.18)$$

First suppose that $k = k'$, which occurs when $g, k \rightarrow g', k'$ is a standard move that does not change the number of groups. Algorithms 5 and 1 have the same probability of proposing a specific standard move, so the proposed-move probabilities $\pi(g, k \rightarrow g', k')$ and $\pi(g', k' \rightarrow g, k)$ in Algorithm 7 are (aside from a factor of $1 - \frac{1}{2k(n+1)}$) the same as the probabilities $\pi(g \rightarrow g')$ and $\pi(g' \rightarrow g)$, respectively, in Algorithm 6. Additionally, by the definition of conditional probability, $\mathbb{P}(g, k) \propto \mathbb{P}(g|k)$ and $\mathbb{P}(g', k') \propto \mathbb{P}(g'|k')$ with the same proportionality constant. The detailed-balance result (5.17) for the fixed- k case then implies that

$$\frac{\mathbb{P}(g'_{(1)}, k')}{\mathbb{P}(g_{(1)}, k)} = \frac{\pi(g, k \rightarrow g', k')}{\pi(g', k' \rightarrow g, k)},$$

so the detailed-balance equation (5.18) holds for this case.

Now suppose that $k' \neq k$. We will show that detailed balance holds when $g, k \rightarrow g', k'$ is a group-addition move. Because $\mathbb{P}(g_{(1)}|k) = \prod_{r=1}^{k-1} \frac{n_r^{(1)}!(n-n_r^{(1)})!}{(n+1)!}$ and $\mathbb{P}(k) = \frac{e^{-1}}{(k-1)!}$, we obtain

$$\frac{\mathbb{P}(g'_{(1)}, k')}{\mathbb{P}(g_{(1)}, k)} = \frac{(k-1)! \prod_{r=1}^k \frac{n'_r{}^{(1)}!(n-n'_r{}^{(1)})!}{(n+1)!}}{k! \prod_{r=1}^{k-1} \frac{n_r^{(1)}!(n-n_r^{(1)})!}{(n+1)!}} = \frac{1}{k(n+1)}.$$

From Algorithms 1, 2, and 7, the proposal probabilities $\pi(g, k \rightarrow g', k')$ and $\pi(g', k' \rightarrow g, k)$ are

$$\begin{aligned} \pi(g, k \rightarrow g', k') &= \frac{1}{L} \times \frac{1}{2k(n+1)} \times \frac{1}{k}, \\ \pi(g', k' \rightarrow g, k) &= \frac{1}{L} \times \left(1 - \frac{1}{2(k+1)(n+1)}\right) \times \frac{1}{k} \times \frac{1}{2}, \end{aligned}$$

where the expression for $\pi(g', k' \rightarrow g, k)$ follows from the fact that $g', k' \rightarrow g, k$ must be a standard move that removes a group. Therefore,

$$\frac{\pi(g, k \rightarrow g', k')}{\pi(g', k' \rightarrow g, k)} = \frac{1}{k(n+1)} + \Theta\left(\frac{1}{n^2}\right).$$

Because $\frac{\mathbb{P}(g', k')}{\mathbb{P}(g, k)} = \frac{1}{k(n+1)}$, we have that

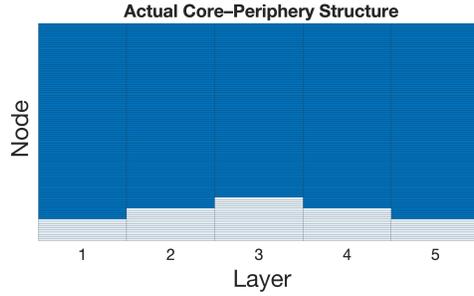
$$\frac{\mathbb{P}(g', k')}{\mathbb{P}(g, k)} = \frac{\pi(g, k \rightarrow g', k')}{\pi(g', k' \rightarrow g, k)}$$

as $n \rightarrow \infty$. Therefore, the detailed-balance equation (5.18) holds in the limit $n \rightarrow \infty$.

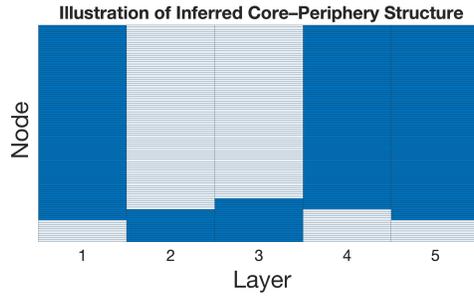
As with the fixed- k case (see Section 5.3.4), we omit the proof of detailed balance when $g, k \rightarrow g', k'$ is a standard move that removes a group, as it is extremely similar to the proof of detailed balance when $g, k \rightarrow g', k'$ is a group-addition move.

5.3.6 Revisiting our Main MCMC Algorithm

Although we proved in Section 5.3.5 that the stationary distribution of the MCMC approach in Algorithm 7 is the same as the desired posterior distribution (5.10) in the limit $n \rightarrow \infty$, in practice, this approach often does not lead to the correct identification of core-periphery structure in temporal networks. This occurs because the posterior distribution $\mathbb{P}(g|A)$ from which we sample using this approach has many local maxima, so the algorithm often becomes stuck at suboptimal maxima. In Figure 4.1, we illustrate the behavior that we commonly see at such suboptimal maxima.



(a) Actual core-periphery structure



(b) Illustration of a local maximum

Figure 5.1: Heat maps of (a) an example of actual core-periphery structure and (b) an illustration of the permuted group structure that we commonly observe at local maxima of $\mathbb{P}(g|A)$ for a 100-node network with 5 layers. Each rectangle in a heat map corresponds to one node-layer (i, ℓ) . The dark blue rectangles signify the value $g_{(i, \ell)}^1 = 1$, and the light blue rectangles signify the value $g_{(i, \ell)}^1 = 0$.

In this example, we consider a network with 100 nodes and 5 layers. We observe that the group structure of the local maximum is permuted from the actual core-periphery structure in some layers. Specifically, for the nodes in layers $\ell = 2$ and $\ell = 3$, the group assignment g coincides with the group assignment of the actual core-periphery structure if we swap the values 0 and 1 of the group-assignment indicator variables $g_{(i, \ell)}^1$. If $g_{(i, \ell)}^1$ is initially equal to 1, then $g_{(i, \ell)}^1$ becomes 0 after the swap; if

$g_{(i,\ell)}^1 = 0$, then $g_{(i,\ell)}^1$ becomes 1 after the swap.

To mitigate this issue, we incorporate multi-node moves into our main MCMC algorithm (see Algorithm 4). In Section 4.4.3, we showed that including multi-node moves in posterior sampling methods for community detection in temporal networks greatly reduces the frequency at which such methods get stuck at suboptimal local maxima where the inferred community assignments become permuted from the correct community assignments in some layers. This observation motivates our choice to include multi-node moves in our main MCMC algorithm.

We now prove that the stationary distribution of our main algorithm (see Algorithm 4) coincides with the desired posterior distribution (5.10). Recall that Algorithm 4 is the same as Algorithm 7 except for the addition of multi-node moves. Including this additional type of move does not impact the detailed-balance calculations in Section 5.3.5 (aside from an additional factor of $1 - p$ in all terms), so it suffices to show the detailed-balance equation

$$\frac{\mathbb{P}(g'_{(1)}, k')}{\mathbb{P}(g_{(1)}, k)} = \frac{\pi(g, k \rightarrow g', k')}{\pi(g', k' \rightarrow g, k)} \quad (5.19)$$

when $g, k \rightarrow g', k'$ and $g', k' \rightarrow g, k$ are multi-node moves. From Algorithms 3 and 4, we have

$$\begin{aligned} \pi(g, k \rightarrow g', k') &= \frac{1}{L} \times p \times \frac{1}{2^{k-1}} \times \frac{1}{2^{k-1}}, \\ \pi(g', k' \rightarrow g, k) &= \frac{1}{L} \times p \times \frac{1}{2^{k'-1}} \times \frac{1}{2^{k'-1}}. \end{aligned}$$

A multi-node move does not change the number of groups, so $k' = k$. Therefore, $\pi(g, k \rightarrow g', k') = \pi(g', k' \rightarrow g, k)$. Recall that we define g' by equation (5.11) for a multi-node move $g, k \rightarrow g', k'$. From this definition, we see for $\ell \in \{2, \dots, L\}$ that g

and g' have the same group assignments in the first layer, so

$$\mathbb{P}(g'_{(1)}, k') = \mathbb{P}(g_{(1)}, k).$$

Therefore, the detailed-balance equation (5.19) holds for multi-node moves $g, k \rightarrow g', k'$ for $\ell \in \{2, \dots, k\}$. However, because a multi-node move for $\ell = 1$ nearly always changes the group sizes in layer 1, the detailed-balance equation (5.19) does not hold for such a move. If we choose to restrict Algorithm 3 to allow multi-node moves only in layers $\ell \in \{2, \dots, k\}$ (as we discussed previously in Section 5.3.1.3), the stationary distribution of our main algorithm (see Algorithm 4) would be the same as the desired posterior distribution (5.10). However, this choice would cause our MCMC algorithm to become stuck at local extrema with permuted group assignments in the first layer. This would significantly decrease the performance of our approach. Therefore, we do not do this, and we instead allow multi-node moves in all layers.

5.3.7 Computation of Acceptance Probability

We now discuss how we compute the acceptance probability (5.13). Recall from (5.13) that we must compute

$$\alpha(g, k \rightarrow g', k') = \min \left\{ 1, \frac{\mathbb{P}(A|g', k)F(g'|k)}{\mathbb{P}(A|g, k)F(g|k)} \right\}.$$

It is straightforward to compute $\mathbb{P}(A|g, k)$ from (5.1). To compute $F(g|k)$, recall from (5.6) and (5.4) that

$$\mathbb{P}(g_{(\ell)}^r | g_{(\ell-1)}^r) = \prod_{s=0}^1 \left[\frac{1}{\binom{n_{r;s}^{(\ell-1)}}{c_{r;ss}^{(\ell)}}} \times J(n_s(g_{(\ell-1)}^r) - c_{r;ss}^{(\ell)}, n_s(g_{(\ell-1)}^r)) \right],$$

where

$$F(g|k) = \prod_{\ell=2}^L \prod_{r=1}^{k-1} \mathbb{P}(g_{(\ell)}^r | g_{(\ell-1)}^r)$$

and

$$J(k_1, k_2) = \int_0^1 x^{k_1} \frac{x-1}{x^{k_2+1}-1} dx.$$

As in Section 4.4.1, to minimize computational cost, we precompute

$$J(k_1, k_2) = \int_0^1 x^{k_1} \frac{x-1}{x^{k_2+1}-1} dx$$

for all k_1 and k_2 such that $0 \leq k_1 \leq k_2 \leq n$ using the procedure in Section B.2.1. This precomputation allows us to avoid needing to repeatedly recompute $J(k_1, k_2)$ when computing (4.33).³ We also use analogous arguments to those in equations (4.34)–(4.36) to avoid directly calculating terms in (5.6) that will cancel out in the expression (5.13) for the acceptance probability $\alpha(g, k \rightarrow g', k')$. This reduces the

³If n is sufficiently large, the integral $J(k_1, k_2)$ can become very small, which causes finite-precision issues and thereby leads to inaccurate results when computing $J(k_1, k_2)$ using (B.14). To mitigate this problem, we use the approximation

$$\frac{J(k_1+1, k_2)}{J(k_1, k_2)} \approx 1$$

for large k_1 and k_2 . In particular, for fixed k_2 , we set the computed values of $J(k_1, k_2)$ to $\exp(-16)$ for all $k_1 \geq k'_1$, where k'_1 is the smallest k'_1 such that $J(k'_1, k_2) < \exp(-16)$.

number of times that we need to evaluate $J(k_1, k_2)$. Our implementation of Algorithm 4 is available at <https://github.com/tfaust0196/mhCorePeriphery>.

5.4 Discussion of our Group-Assignment Approach

We are equipped to discuss an additional rationale behind our choice of group-assignment probability distribution $\mathbb{P}(g|k)$ now that we have introduced Algorithm 4 (see Section 5.3.3, which we use to sample from the desired posterior distribution $\mathbb{P}(g, k|A)$ in (5.10). Recall from Algorithm 4 and Section 5.3.2 that the acceptance probability $\alpha(g, k \rightarrow g', k')$ is

$$\alpha(g, k \rightarrow g', k') = \min \left\{ 1, \frac{\mathbb{P}(A|g', k')F(g', k')}{\mathbb{P}(A|g, k)F(g, k)} \right\}. \quad (5.20)$$

By (5.2) and (5.5), we generate the group assignments for each group independently. Therefore, if we propose a group-addition move that adds an empty group r , the acceptance probability becomes

$$\mathbb{P}(g_{(i,\ell)}^r = 0 \text{ for all } i \in \{1, \dots, n\} \text{ and } \ell \in \{1, \dots, L\}).$$

Consequently, if a generative model is biased against groups with few node-layers, then the acceptance probability (5.20) is very small. This, in turn, causes an associated statistical-inference method to underestimate the number of groups.

For a monolayer network (i.e., an ordinary graph), one can resolve the above issue by generating the group sizes uniformly at random. With this choice, the probability of generating an empty group is $\frac{1}{n+1}$. However, for a temporal network, it is difficult to choose a generative model that is not biased against groups with few node-layers. In Section 4.3, we showed that the commonly-used approach (see, e.g., [1,25,50,88]) of

evolving group assignments via a discrete-time Markov process (where one bases the group assignment of a node in a given layer on its group assignment in the previous layer) leads to an increased bias against small and large groups in later layers. This, in turn, leads to a strong bias of a generative model against small and large groups in a network. In our context of hierarchical core–periphery structure, this bias makes the acceptance probability (5.20) of a group-addition move prohibitively small, leading to an underestimation in the number of groups.

Our novel group-assignment approach is much less biased against small and large groups than such Markov-process approaches. This is the case because our approach generates the group assignments for a layer based on all of the previous group assignments in the previous layer, instead of using a discrete-time Markov process to evolve the group assignments of each node individually. (See Sections 4.2.3.3 and 4.3 for more information.) Therefore, our method is much less likely to underestimate the number of groups in a network.

This discussion illustrates the practical importance of carefully considering potentially undesirable assumptions of the generative models in statistical-inference methods. See Chapter 4 for further discussion and examples in the context of community structure.

5.5 Application to Real-World Networks

In this section, we apply our hierarchical core–periphery detection method to two real-world networks and discuss the structures that we identify in these networks. In Section 5.5.1, we consider a network of terrorists in the Indian states of Jammu and Kashmir. In Section 5.5.2, we consider a network of co-appearances in the *Luke*

Gospel.

5.5.1 Jammu–Kashmir Terrorist Network

We first apply our method to a network of links between terrorist organizations in the Indian states of Jammu and Kashmir from 2000 to 2003 [77]. This temporal network consists of $n = 34$ nodes (which represent terrorist organizations) and $L = 4$ layers (which represent years). Applying our hierarchical core–periphery detection method to this network yields $k = 2$ groups and the core–periphery structure in Figure 5.2. To generate the core–periphery structure in Figure 5.2, we perform 5 runs of 10^6 steps of our main MCMC algorithm (see Algorithm 4) with the number k of groups initialized to 4, the probability p of a multi-node move set to 10^{-3} , and the initial group assignments selected uniformly at random for each run. We then save the output group assignments every 10^4 steps. For each node-layer (i, ℓ) , we set the group assignment for (i, ℓ) to be the most frequent group assignment among each of the saved group assignments (across all runs).

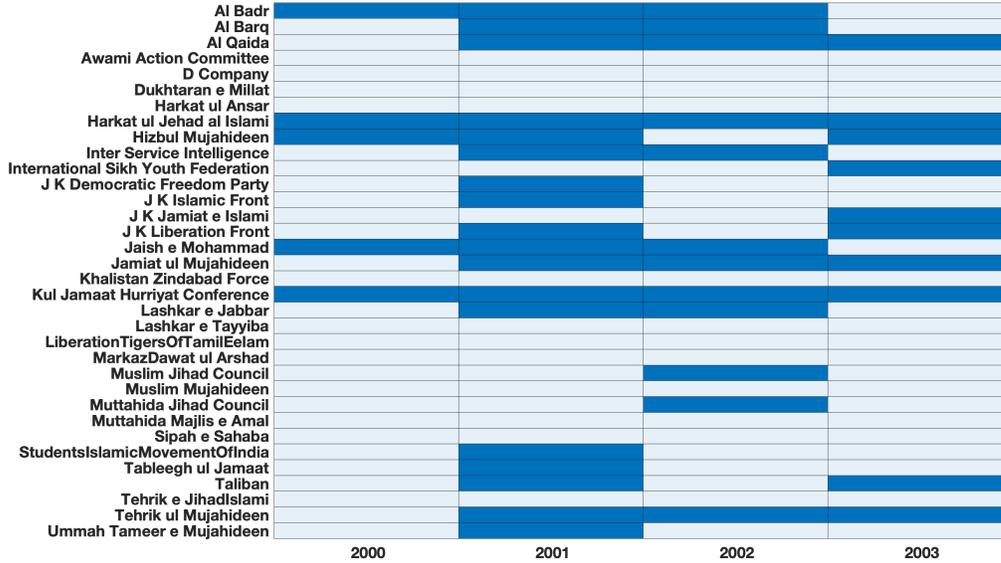


Figure 5.2: The inferred core-periphery structure in the Jammu-Kashmir terrorist network. The dark blue rectangles signify the value $g_{(i,\ell)}^1 = 1$, and light blue rectangles signify the value $g_{(i,\ell)}^1 = 0$. The horizontal axis indicates years and the vertical axis indicates terrorist organizations.

To demonstrate that the detected core-periphery structure is reasonable, we plot heat maps (see Figure 5.3) of the adjacency matrices $A^{(\ell)}$ for each layer ℓ . In these heat maps, we permute the rows and columns so that all node-layers (i, ℓ) with $g_{(i,\ell)}^1 = 1$ preferentially occur earlier. We also include a dividing line between the node-layers with $g_{(i,\ell)}^1 = 1$ and the node-layers with $g_{(i,\ell)}^1 = 0$. In Figure 5.3, we show these heat maps, and we can see that the node-layers with $g_{(i,\ell)}^1 = 1$ (i.e., the core node-layers) are densely connected to other nodes with $g_{(i,\ell)}^1 = 1$ and that the node-layers with $g_{(i,\ell)}^1 = 0$ (i.e., the peripheral node-layers) are sparsely connected to

all other node-layers.

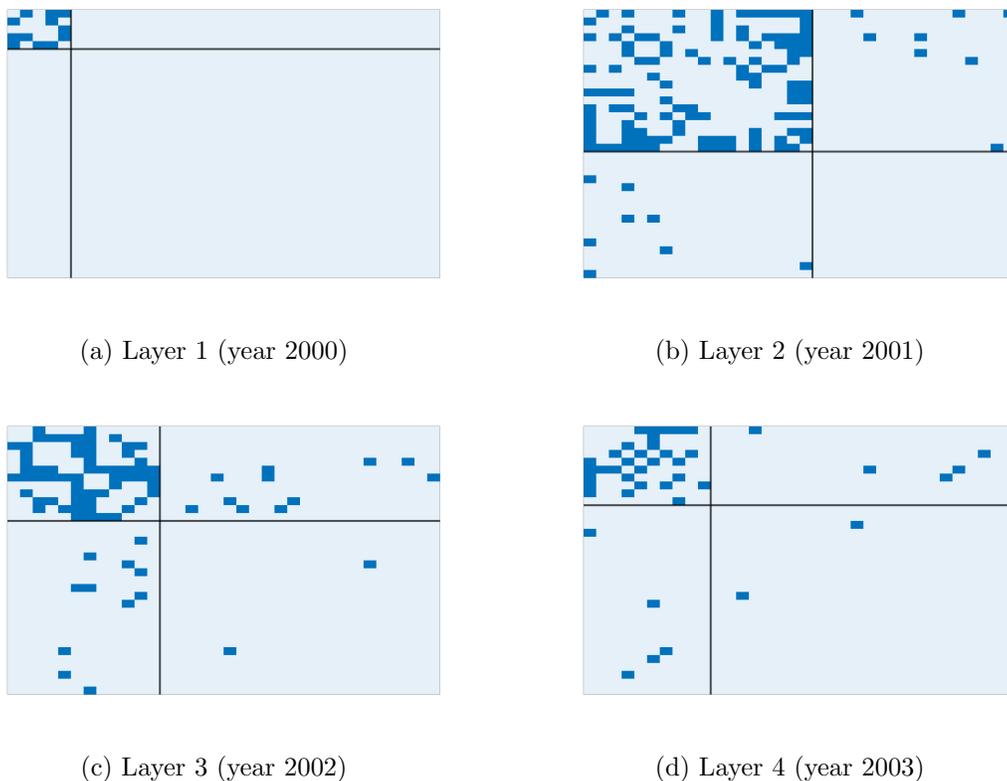


Figure 5.3: Permutated adjacency matrices to illustrate the inferred core-periphery structure in the Jammu-Kashmir terrorist network. We show (a) layer 1 (the year 2000), (b) layer 2 (the year 2001), (c) layer 3 (the year 2002), and (d) layer 4 (the year 2003).

5.5.2 Literary Co-Appearance Network

We also apply our method to a network of co-appearances in the *Luke Gospel* [29]. This temporal network consists of $n = 76$ nodes (which represent characters) and

$L = 5$ layers (which represent non-overlapping ranges of 4 consecutive chapters). We place an edge between two nodes in a given layer if the two associated characters encounter each other in those chapters. As in Section 5.5.1, we perform 5 runs of 10^6 steps of our main MCMC algorithm (see Algorithm 4) with the number k of groups initialized to 4, the probability p of a multi-node move set to 10^{-3} , and the initial group assignments selected uniformly at random for each run. We then save the output group assignments every 10^4 steps. In contrast to the results from Section 5.5.1, the inferred group assignments differ greatly between runs. Therefore, unlike for the Jammu–Kashmir terrorist network, we cannot reasonably choose to set the group assignment for each node-layer (i, ℓ) to be the most frequent group assignment across all runs.

One of the reasons that different runs yield different group assignments is that group assignments are permuted between layers in two of the five runs. In Figure 5.4, we illustrate this with an example of such a run.



Figure 5.4: The inferred core–periphery structure from one run of Algorithm 4 applied to the *Luke Gospel* literary co-appearance network. The light blue rectangles signify the value $g^1_{(i,\ell)} = 1$, and the dark blue rectangles signify the value $g^1_{(i,\ell)} = 0$. The horizontal axis indicates non-overlapping ranges of 4 consecutive chapters and the vertical axis indicates characters.

Recall from Section 5.3.6 that we commonly observe permutations of group assignments between layers in local maxima of the MCMC algorithm without multi-node moves. This suggests that multi-node moves are less effective at avoiding such extrema for the *Luke Gospel* literary co-appearance network than for the Jammu–Kashmir terrorist network. We do not have an explanation for why multi-node moves are less effective for this example. However, the inferred core–periphery structure for each of the layers appears to be plausible. When we permute the adjacency matri-

ces according to the core–periphery structure in Figure 5.4 (as in Section 5.5.1), the densities of the edges between node-layers depend on the groups that the node-layers are in.

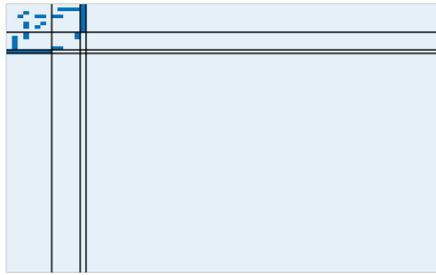
In runs where group assignments are not permuted between layers, we detect reasonable core–periphery structure. In particular, one of the runs of our core–periphery detection method on the *Luke Gospel* literary co-appearance network yields $k = 3$ groups and the core–periphery structure in Figure 5.5. Observe that the group assignments are not permuted between layers (except possibly for the last layer).



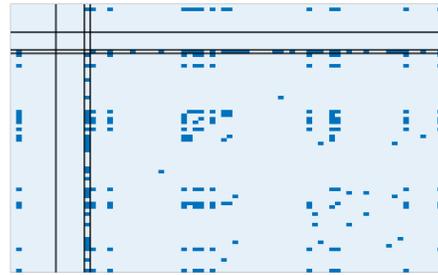
Figure 5.5: The inferred core–periphery structure from one run of Algorithm 4 applied to the *Luke Gospel* literary co-appearance network. The light blue rectangles signify the values $g_{(i,\ell)}^1 = 0$ and $g_{(i,\ell)}^2 = 0$, the dark blue rectangles signify the values $g_{(i,\ell)}^1 = 1$ and $g_{(i,\ell)}^2 = 0$, the light purple rectangles signify the values $g_{(i,\ell)}^1 = 0$ and $g_{(i,\ell)}^2 = 1$, and the dark purple rectangles signify the values $g_{(i,\ell)}^1 = 1$ and $g_{(i,\ell)}^2 = 1$. The horizontal axis indicates non-overlapping ranges of 4 consecutive chapters and the vertical axis indicates characters.

To demonstrate that the detected core–periphery structure in Figure 5.5 is reasonable, we plot heat maps (see Figure 5.6) of the adjacency matrices $A^{(\ell)}$ for each layer ℓ . In these heat maps, we permute the rows and columns so that all node-layers (i, ℓ) with $g_{(i,\ell)}^1 = 1$ and $g_{(i,\ell)}^2 = 1$ appear earliest, followed by node-layers (i, ℓ) with $g_{(i,\ell)}^1 = 0$ and $g_{(i,\ell)}^2 = 1$, node-layers (i, ℓ) with $g_{(i,\ell)}^1 = 1$ and $g_{(i,\ell)}^2 = 0$, and finally

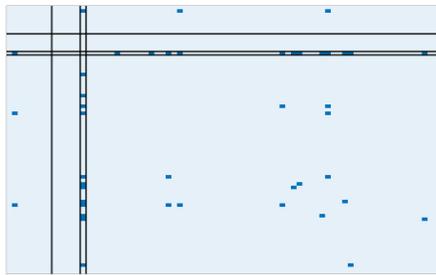
node-layers (i, ℓ) with $g_{(i,\ell)}^1 = 0$ and $g_{(i,\ell)}^2 = 0$. We observe that the densities of the edges between node-layers depend on the groups that the node-layers are in. For example, in layers 2–4, node-layers with group assignments $g_{(i,\ell)}^1 = 1$ and $g_{(i,\ell)}^2 = 0$ tend to be densely connected to node-layers with group assignments $g_{(i,\ell)}^1 = 0$ and $g_{(i,\ell)}^2 = 0$ but sparsely connected to node-layers with group assignments $g_{(i,\ell)}^1 = 1$ and $g_{(i,\ell)}^2 = 1$. However, the validity of the inferred core–periphery structure is less evident than the inferred core–periphery structure of the Jammu–Kashmir terrorist network. We hypothesize that this is because the *Luke Gospel* literary co-appearance network is less dense than the Jammu–Kashmir terrorist network.



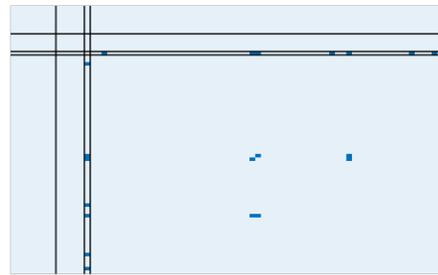
(a) Layer 1



(b) Layer 2



(c) Layer 3



(d) Layer 4



(e) Layer 5

Figure 5.6: Permutated adjacency matrices to illustrate the inferred core-periphery structure in the *Luke Gospel* literary co-appearance network. We show (a) layer 1, (b) layer 2, (c) layer 3, (d) layer 4, and (e) layer 5.

5.6 Conclusions and Discussion

We proposed a method to identify hierarchical core–periphery structure in temporal networks. We applied this method to two real-world networks and obtained reasonable inferred structures when our MCMC method converged.

By using the group-evolution probability distribution from Section 4.2.3.3 (instead of a group-evolution probability distribution that is based on a discrete-time Markov process), our Markov-chain Monte Carlo (MCMC) approach for statistical inference is able to mitigate the underestimation of the number of groups in networks in the detection of hierarchical core–periphery structure. Additionally, by using multi-node moves, we sped up the convergence to the inferred core–periphery structure.

There are a variety of ways to build on our work. The primary weakness of our approach is the high computational cost of each iteration in our MCMC approach, which (despite the use of multi-node moves to reduce the number of iterations to achieve convergence) makes it prohibitive to apply our method to networks with many nodes or layers. Therefore, it is important to develop and implement more computationally efficient methods to identify hierarchical core–periphery structure in temporal networks and more generally in multilayer networks.

We hypothesized in Section 5.5.2 that our method identifies hierarchical core–periphery structure less accurately for sparse networks than for dense networks. It is important to examine the performance of our method on synthetic networks with different densities to better understand the effect of density on it. Finally, we emphasize that there have been very few studies of core–periphery structure in temporal and multilayer networks — a situation that contrasts with the voluminous analysis

of temporal and multilayer community structure [33, 72] — and such efforts deserve more attention. We expect that such studies will yield both theoretical and practical insights.

CHAPTER 6

Conclusion

6.1 Summary

One of the fundamental areas of network science is the algorithmic detection of mesoscale structures, which involve groups of nodes that are larger than a single node but smaller than an overall network [56]. There have been a variety of approaches have been developed to detect mesoscale structures, including optimization of objective functions [2, 51, 54], non-negative matrix and tensor factorization [17, 24, 74], information-theoretic methods (such as ones that minimize description length) [67, 75, 78], local methods [34, 37, 47], and spectral methods [4, 5]. In this thesis, we analyzed techniques for statistical inference of mesoscale structures [68, 81, 89]. Statistical-inference approaches have a variety of favorable properties, including convergence guarantees and better abilities than other methods at avoiding overfitting [66].

When applying statistical inference to network applications, it is important to use a generative model that is based on realistic assumptions [66]. In this thesis, we illustrated the effects of unrealistic generative models on the detection of mesoscale structures, and we developed an approach that mitigates them. We demonstrated (see Section 4.4.3) that biases against communities with large or small numbers

of nodes cause methods to infer community structure less accurately in networks with small or large communities. We also demonstrated (see Section 5.4) that biases against groups without any node-layers cause our multilayer core–periphery detection method to greatly underestimate the number of mesoscale groups in a network. Such examples illustrate the importance of carefully evaluating the behavior of generative models.

When performing statistical inference in temporal networks, which change with time, using multi-node moves can yield significant improvement in the performance of inference methods. For example, in Section 4.4.3, we saw that multi-node moves greatly improve the performance of Gibbs sampling for several community-detection approaches. Analyzing the behavior at the local extrema that we obtained from unmodified Gibbs sampling allowed us to effectively employ multi-node moves. We were able to understand the behavior at local extrema by applying unmodified Gibbs sampling to networks with known ground-truth mesoscale structure. We recommend such an approach when possible, as it is much simpler to identify patterns in local extrema when they can be compared with a known ground truth.

6.1.1 Future Directions

There are many viable ways to build on our work. For example, one can improve the running time per iteration of our statistical-inference methods, modify our generative models to incorporate more previous layers when generating the community or other group assignments for a given layer, and apply similar statistical-inference methods to other types of mesoscale structure. We now discuss each of these possible future directions in more detail.

First, although multi-node moves improve the convergence behavior of both our Gibbs-sampling approach for community detection (see Section 4.4.1) and our MCMC approach for core-periphery detection (see Section 5.3.1), the iterations of each method are computationally costly. Consequently, it is time-consuming to apply our methods to networks with many nodes or many layers. An improvement in the running time of such methods — either through more efficient implementations of our existing algorithms or through the creation of new and more efficient algorithms — would make it possible to apply our approaches to a wider variety of networks.

Second, our generative models for community structure and core-periphery structure assume that the group assignments for a given layer depend only on those in the previous layer. This choice is necessary for our statistical-inference methods to be computationally tractable. However, such models are somewhat unrealistic. If one is able to devise tractable generative models that allow more general group evolution, such that group assignments in a layer depend on more than just the previous layer, it would likely lead to more realistic inference results.

Third, although this thesis focused on the application of statistical-inference methods to community detection and core-periphery detection in temporal networks, there are many other important types of mesoscale structure for which inference methods have been considered sparingly or not at all. One example of such a mesoscale structure is a “ranked community”, which is a subset of nodes in a directed network whose edge directions are biased according to some hierarchy [44]. The application of inference methods to identify this and other structures — such as role structures [73], motifs [52, 62], and others — would improve understanding of these structures and hence yield insights into a variety of complex systems, such as faculty hiring networks [82], transaction networks [41], and biological networks [48].

APPENDIX A

Table of Commonly-Used Notation

In this appendix, we give a table (see Table A.1) of commonly-used notation. Each entry of this table includes the notation, an accompanying short definition, and a location of the full definition.

Table A.1: Table of commonly-used notation.

| Notation | Short Definition | Chapter, Section, or Equation |
|----------------------------|--|-------------------------------|
| n | Number of nodes in a network | Chapter 3 |
| L | Number of layers in a temporal network | Chapter 3 |
| i, j | Indices of specific nodes | Chapter 3 |
| ℓ | Index of a specific layer | Chapter 3 |
| $(i, \ell_1), (j, \ell_2)$ | Indices of specific node-layers | Chapter 3 |
| A | Adjacency structure of a network | Chapter 3 |
| $A^{(\ell)}$ | Adjacency matrix of layer ℓ in a temporal network | Chapter 3 |
| k | Number of communities or other groups | Section 4.2.1 and Section 5.2 |
| $g_{(i,\ell)}$ | Community assignment of node-layer (i, ℓ) | Section 4.2.1 |

| Notation | Short Definition | Chapter, Section, or Equation |
|---------------------------------|--|----------------------------------|
| $g_{(i,\ell)}^r$ | Group-assignment indicator variable of node-layer (i, ℓ) and group r | Section 5.2 |
| $g_{(\ell)}$ | Set of community assignments of all node-layers in layer ℓ | Section 4.2.1 |
| $g_{(\ell)}^r$ | Set of group-assignment indicator variables of all node-layers in layer ℓ and group r | Section 5.2.1 |
| $h(((i, \ell), (j, \ell)))$ | Highest common group of node-layers (i, ℓ) and (j, ℓ) | Section 5.2 |
| $\text{Dir}(\gamma)$ | Dirichlet distribution with parameters $\gamma = (\gamma_1, \dots, \gamma_r)$ | Section 4.2.2.2 |
| $\text{Markov}(\{K^{(\ell)}\})$ | Markov process with transition kernels $\{K^{(\ell)}\}$ | Section 4.2.3.2 |
| $\overline{\text{Geom}}(n, p)$ | Modified geometric distribution with parameters n and p | Equation (4.16) |
| \mathcal{C}_n^k | Set of weak compositions of n with k parts | Section 4.2.3.3 |
| c | Vector of community sizes | Section 4.2.3.3 |
| $\mathcal{G}(c)$ | Set of all community assignments for which community r has size c_r for each $r \in \{1, \dots, k\}$ | Section 4.2.3.3 |
| $J(k_1, k_2)$ | $\int_0^1 x^{k_1} \frac{x-1}{x^{k_2+1}-1} dx$ | Equation (B.12) |
| Δ^{k-1} | $\left\{ (v_1, \dots, v_k)^T \mid \sum_{r=1}^k v_r = 1 \right\} \cap [0, 1]^k$ | Equation (4.8) |

APPENDIX B

Expressions for the Community-Assignment Probabilities

In this appendix, we derive closed-form expressions for the community-assignment probability distributions $\mathbb{P}(g)$ for the Bazzi et al. approach [1] (see Section 4.2.3.2) and our exchangeability-based approach (see Section 4.2.3.3). We use these expressions in the Gibbs-sampling procedures that we described in Section 4.4.1.

B.1 Closed-Form Expression for $\mathbb{P}(g)$ for the Bazzi et al. Approach

Recall from Section 4.2.3.2 that the Bazzi et al. approach samples community assignments g via the discrete-time Markov process (4.12). This Markov process is

$$\begin{aligned}\pi &\sim \text{Dir}(\gamma), \\ g_{(i,1)} | \pi &\sim \pi, \\ \{g_{(i,\ell)}\}_{\ell=2}^L | \alpha, K &\sim \text{Markov}(\{\alpha_\ell I + (1 - \alpha_\ell)K^{(\ell)}\}).\end{aligned}$$

where $\gamma = (1, \dots, 1)$ and $\alpha = (\alpha_2, \dots, \alpha_L)$. In the Bazzi et al. approach,

$$\begin{aligned}\alpha_\ell &\sim \text{Unif}(0, 1), \\ \kappa^{(\ell)} &\sim \text{Dir}(\mu^{(\ell)}), \\ K_{s^*}^{(\ell)} &= \kappa^{(\ell)},\end{aligned}$$

where we also assume that $\mu^{(\ell)} = (1, \dots, 1)$ for each layer $\ell \in \{2, \dots, L\}$.

Because we generate $\{g_{(i,\ell)}\}_{\ell=1}^L$ via a discrete-time Markov process, the community assignment of a node in a given layer depends only on its community assignment in the previous layer. Therefore,

$$\mathbb{P}(g) = \mathbb{P}(g_{(1)}) \prod_{\ell=2}^L \mathbb{P}(g_{(\ell)} | g_{(\ell-1)}) \quad (\text{B.1})$$

and

$$\mathbb{P}(g_{(\ell)} | g_{(\ell-1)}, K, \alpha) = \prod_{i=1}^n \mathbb{P}\left(g_{(i,\ell)} \mid g_{(i,\ell-1)}, K, \alpha\right). \quad (\text{B.2})$$

To derive a closed-form expression for $\mathbb{P}(g)$, it suffices to derive closed-form expressions for $\mathbb{P}(g_{(1)})$ and $\mathbb{P}(g_{(\ell)} | g_{(\ell-1)})$.

Because the procedure

$$\begin{aligned}\pi &\sim \text{Dir}(\gamma), \\ g_{(i,1)} | \pi &\sim \pi,\end{aligned}$$

for sampling $g_{(1)}$ is identical to the nodewise community-assignment approach for monolayer networks that we discussed in Section 4.2.2.2, it is equivalent to sample $g_{(1)}$ using a uniform distribution on community sizes. Namely, we first choose the sizes n_1, \dots, n_k of communities $1, \dots, k$ uniformly at random from the set $\{(n_1, \dots, n_k) \mid \sum_{i=1}^k n_i = n\} \cap \{0, \dots, n\}^k$ of ordered pairs of k non-negative integer

elements that sum to n , and we then choose $g_{(1)}$ uniformly at random from the set of all community assignments with n_i nodes in community i for all $i \in \{1, \dots, k\}$.

Therefore,

$$\mathbb{P}(g_{(1)}) = \frac{1}{\binom{k+n-1}{n}} \frac{n_1(g_{(1)})! \times \dots \times n_k(g_{(1)})!}{n!}, \quad (\text{B.3})$$

where we recall that $n_r(g_{(1)})$ is the number of times that r appears in $g_{(1)}$.

To derive a closed-form expression for $\mathbb{P}(g_{(\ell)}|g_{(\ell-1)})$, we recall from (4.11) that

$$\mathbb{P}(g_{(i,\ell)} = r | g_{(i,\ell-1)} = s, \alpha, K) = \alpha_\ell \delta_{rs} + (1 - \alpha_\ell) K_{sr}^{(\ell)}.$$

It thus follows directly from (B.2) that

$$\begin{aligned} \mathbb{P}(g_{(\ell)}|g_{(\ell-1)}, K, \alpha) &= \prod_{i=1}^n \mathbb{P}\left(g_{(i,\ell)} \middle| g_{(i,\ell-1)}, K, \alpha\right) \\ &= \prod_{i=1}^n \left(\alpha_\ell \delta_{g_{(i,\ell)}, g_{(i,\ell-1)}} + (1 - \alpha_\ell) K_{g_{(i,\ell-1)}g_{(i,\ell)}}^{(\ell)} \right). \end{aligned}$$

Finally, because

$$\begin{aligned} \alpha_\ell &\sim \text{Unif}(0, 1), \\ \kappa^{(\ell)} &\sim \text{Dir}(\mu^{(\ell)}), \\ K_{s*}^{(\ell)} &= \kappa^{(\ell)}, \end{aligned}$$

with $\mu^{(\ell)} = (1, \dots, 1)$ for each $\ell \in \{2, \dots, L\}$, we have

$$\mathbb{P}(g_{(\ell)}|g_{(\ell-1)}) = \int \dots \int_{[0,1] \times \Delta^{k-1}} \prod_{i=1}^n \left(\alpha_\ell \delta_{g_{(i,\ell)}, g_{(i,\ell-1)}} + (1 - \alpha_\ell) \kappa_{g_{(i,\ell)}}^{(\ell)} \right) d\mu(\alpha_\ell, \kappa^{(\ell)}), \quad (\text{B.4})$$

where we recall from (4.8) that $\Delta^{k-1} = \{(v_1, \dots, v_k) | \sum_{i=1}^k v_i = 1\} \cap [0, 1]^k$ and μ is the product measure of a uniform measure on $[0, 1]$ and a uniform measure on Δ^{k-1} .

(The measure is uniform because $\mu^{(\ell)} = (1, \dots, 1)$.) Combining equation (B.4) with (B.1) and (B.3) yields a closed-form expression for $\mathbb{P}(g)$.

B.2 Closed-Form Expression for $\mathbb{P}(g)$ for our Exchangeability-Based Approach

Recall from Section 4.2.3.3 that our exchangeability-based approach samples community assignments g via the process in (4.14) and (4.17). For each $r \in \{1, \dots, k\}$, we sample $\mathbf{g}'_{r,\ell}$ according to the following procedure:

$$\begin{aligned} \pi &\sim \text{Dir}(\gamma), \\ g_{(i,1)} | \pi &\sim \pi, \\ p_{r,\ell} &\sim \text{Unif}(0, 1), \\ c_{rr}^{(\ell)} | p_{r,\ell} &\sim \overline{\text{Geom}}(n_r(g_{(\ell-1)}), p_{r,\ell}), \\ \mathbf{c}_{r,-r}^{(\ell)} | c_{rr}^{(\ell)} &\sim \text{Unif}\left(\mathcal{C}_{n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)}}^{k-1}\right), \\ \mathbf{g}'_{r,\ell} | \mathbf{c}_r^{(\ell)} &\sim \text{Unif}(\mathcal{G}_r^{(\ell-1)}(\mathbf{c}_r^{(\ell)})), \end{aligned}$$

where we recall that $\gamma = (1, \dots, 1)$. We then set $g_{(\ell)} = \bigoplus_{r=1}^k \mathbf{g}'_{r,\ell}$. The above procedure implies that the community assignments in a given layer depend only on the community assignments in the previous layer. We thus obtain

$$\mathbb{P}(g) = \mathbb{P}(g_{(1)}) \prod_{\ell=2}^L \mathbb{P}(g_{(\ell)} | g_{(\ell-1)}), \quad (\text{B.5})$$

which takes the same form as (B.1). Additionally, because the procedure to sample $g_{(1)}$ and the choice of γ is the same as in the Bazzi et al. [1] approach, we obtain (using the same logic as in Appendix B.1) the expression

$$\mathbb{P}(g_{(1)}) = \frac{1}{\binom{k+n-1}{n}} \frac{n_1(g_{(1)})! \times \dots \times n_k(g_{(1)})!}{n!}, \quad (\text{B.6})$$

where we recall that $n_r(g_{(1)})$ is the number of times that r appears in $g_{(1)}$.

To derive a closed-form expression for $\mathbb{P}(g_{(\ell)}|g_{(\ell-1)})$, we first note that we obtain the set $g_{(\ell)}$ of community assignments for layer ℓ by concatenating $\mathbf{g}'_{r,\ell}$, which is the set of community assignments for layer ℓ when restricted to nodes in community r in layer $\ell - 1$ for each $r \in \{1, \dots, k\}$. Therefore,

$$\mathbb{P}(g_{(\ell)}|g_{(\ell-1)}) = \prod_{r=1}^k \mathbb{P}(\mathbf{g}'_{r,\ell}). \quad (\text{B.7})$$

Recall that we generate $\mathbf{g}'_{r,\ell}$ using the procedure

$$\begin{aligned} c_{rr}^{(\ell)} | p_{r,\ell} &\sim \overline{\text{Geom}}(n_r(g_{(\ell-1)}), p_{r,\ell}), \\ \mathbf{c}_{r,-r}^{(\ell)} | c_{rr}^{(\ell)} &\sim \text{Unif}\left(\mathcal{C}_{n_r(g_{(\ell-1)})-c_{rr}^{(\ell)}}^{k-1}\right), \\ \mathbf{g}'_{r,\ell} | \mathbf{c}_r^{(\ell)} &\sim \text{Unif}(\mathcal{G}_r^{(\ell-1)}(\mathbf{c}_r^{(\ell)})). \end{aligned}$$

By the definition (4.16) of the probability mass function of $\overline{\text{Geom}}(n_r(g_{(\ell-1)}), p_{r,\ell})$, we have

$$\mathbb{P}(c_{rr}^{(\ell)}|p_{r,\ell}) = p_{r,\ell}^{n_r(g_{(\ell-1)})-c_{rr}^{(\ell)}} \frac{p_{r,\ell} - 1}{p_{r,\ell}^{n_r(g_{(\ell-1)})+1} - 1}. \quad (\text{B.8})$$

Because $\mathbf{c}_{r,-r}^{(\ell)} | c_{rr}^{(\ell)} \sim \text{Unif}\left(\mathcal{C}_{n_r(g_{(\ell-1)})-c_{rr}^{(\ell)}}^{k-1}\right)$, a combinatorial argument combined with the definition of weak compositions implies that

$$\mathbb{P}(\mathbf{c}_{r,-r}^{(\ell)} | c_{rr}^{(\ell)}) = \frac{1}{\binom{n_r(g_{(\ell-1)})-c_{rr}^{(\ell)}+k-2}{n_r(g_{(\ell-1)})-c_{rr}^{(\ell)}}}. \quad (\text{B.9})$$

Recall that we sample $\mathbf{g}'_{r,\ell} | \mathbf{c}_r^{(\ell)} \sim \text{Unif}(\mathcal{G}(\mathbf{c}_r^{(\ell)}))$. It follows from a combinatorial argument along with the definition of \mathcal{G} (see Section 4.2.3.3) that

$$\mathbb{P}(\mathbf{g}'_{r,\ell} | \mathbf{c}_r^{(\ell)}) = \frac{1}{\binom{n_r(g_{(\ell-1)})}{c_{r,1}, \dots, c_{r,k}}}, \quad (\text{B.10})$$

where $\binom{n}{k_1, \dots, k_j} = \frac{n!}{k_1! k_2! \dots k_j!}$ and $\sum_{i=1}^j k_i = n$. Combining (B.8), (B.9), and (B.10) yields

$$\begin{aligned} \mathbb{P}(\mathbf{g}'_{r,\ell} | p_{r,\ell}) &= \mathbb{P}(c_{rr}^{(\ell)} | p_{r,\ell}) \mathbb{P}(\mathbf{c}_{r,-r}^{(\ell)} | c_{rr}^{(\ell)}) \mathbb{P}(\mathbf{g}'_{r,\ell} | \mathbf{c}_r^{(\ell)}) \\ &= p_{r,\ell}^{n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)}} \frac{p_{r,\ell} - 1}{p_{r,\ell}^{n_r(g_{(\ell-1)})+1} - 1} \times \frac{1}{\binom{n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)} + k - 2}{n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)}} \binom{n_r(g_{(\ell-1)})}{c_{r,1}, \dots, c_{r,k}}}. \end{aligned}$$

Because $p_{r,\ell} \sim \text{Unif}(0, 1)$, integrating $\mathbb{P}(\mathbf{g}'_{r,\ell} | p_{r,\ell})$ with respect to the probability measure induced by $\mathbb{P}(p_{r,\ell})$ yields

$$\mathbb{P}(\mathbf{g}'_{r,\ell}) = \frac{1}{\binom{n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)} + k - 2}{n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)}} \binom{n_r(g_{(\ell-1)})}{c_{r,1}, \dots, c_{r,k}}} \times \int_0^1 \frac{p_{r,\ell}^{n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)}}}{p_{r,\ell}} \frac{p_{r,\ell} - 1}{p_{r,\ell}^{n_r(g_{(\ell-1)})+1} - 1} dp_{r,\ell}. \quad (\text{B.11})$$

While we can directly compute $\mathbb{P}(\mathbf{g}'_{r,\ell})$ using equation (B.11), we make the computations more efficient by precomputing the integrals

$$\int_0^1 \frac{p_{r,\ell}^{n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)}}}{p_{r,\ell}} \frac{p_{r,\ell} - 1}{p_{r,\ell}^{n_r(g_{(\ell-1)})+1} - 1} dp_{r,\ell}.$$

For notational simplicity, we write

$$J(k_1, k_2) = \int_0^1 x^{k_1} \frac{x - 1}{x^{k_2+1} - 1} dx. \quad (\text{B.12})$$

Combining (B.7), (B.11), and (B.12) then yields

$$\mathbb{P}(g_{(\ell)} | g_{(\ell-1)}) = \prod_{r=1}^k \left[\frac{1}{\binom{n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)} + k - 2}{n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)}} \binom{n_r(g_{(\ell-1)})}{c_{r,1}, \dots, c_{r,k}}} \times J(n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)}, n_r(g_{(\ell-1)})) \right]. \quad (\text{B.13})$$

In concert, the relations (B.13), (B.5), and (B.6) yield a closed-form expression for $\mathbb{P}(g)$.

B.2.1 Computing $J(k_1, k_2)$

To compute (B.13), we need to compute $J(k_1, k_2)$ (see (B.12)). Because $n_r(g_{(\ell-1)}) - c_{rr}^{(\ell)} \leq n_r(g_{(\ell-1)})$, we are able to compute (B.13) by computing $J(k_1, k_2)$ only when $0 \leq k_1 \leq k_2$. When $0 \leq k_1 < k_2$, we use a partial-fractions expansion to obtain

$$\begin{aligned} J(k_1, k_2) &= \int_0^1 x^{k_1} \frac{x-1}{x^{k_2+1}-1} dx \\ &= \sum_{r=1}^{k_2} c_r \left[\log \left(1 - e^{\frac{2\pi i r}{k_2+1}} \right) - \log \left(-e^{\frac{2\pi i r}{k_2+1}} \right) \right], \end{aligned} \quad (\text{B.14})$$

where

$$c_r = \frac{e^{\frac{2\pi i r k_1}{k_2+1}}}{\prod_{s=1, s \neq r}^{k_2} \left(e^{\frac{2\pi i r}{k_2+1}} - e^{\frac{2\pi i s}{k_2+1}} \right)}, \quad (\text{B.15})$$

the symbol i denotes the imaginary unit, and \log denotes the complex base- e logarithm with a branch cut on the negative real line. To compute $J(k_2, k_2)$, we note that

$$\sum_{k_1=0}^{k_2} \left(\int_0^1 x^{k_2} \frac{x-1}{x^{k_2+1}-1} dx \right) = \int_0^1 \left(\sum_{k_1=0}^{k_2} x^{k_2} \frac{x-1}{x^{k_2+1}-1} \right) dx = \int_0^1 1 dx = 1.$$

We thereby obtain

$$\begin{aligned} J(k_2, k_2) &= \int_0^1 x^{k_2} \frac{x-1}{x^{k_2+1}-1} dx \\ &= 1 - \sum_{k_1=0}^{k_2-1} \int_0^1 x^{k_2} \frac{x-1}{x^{k_2+1}-1} dx \\ &= 1 - \sum_{k_1=0}^{k_2-1} J(k_1, k_2). \end{aligned} \quad (\text{B.16})$$

REFERENCES

- [1] Marya Bazzi, Lucas G. S. Jeub, Alex Arenas, Sam D. Howison, and Mason A. Porter. A framework for the construction of generative models for mesoscale structure in multilayer networks. *Physical Review Research*, 2:023100, 2020.
- [2] Marya Bazzi, Mason A. Porter, Stacy Williams, Mark McDonald, Daniel J. Fenn, and Sam D. Howison. Community detection in temporal multilayer networks, with an application to correlation networks. *Multiscale Modeling & Simulation: A SIAM Interdisciplinary Journal*, 14(1):1–41, 2016.
- [3] Punam Bedi and Chhavi Sharma. Community detection in social networks. *WIRES Data Mining and Knowledge Discovery*, 6(3):115–135, 2016.
- [4] Kai Bergermann, Martin Stoll, and Francesco Tudisco. A nonlinear spectral core–periphery detection method for multiplex networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 480(2300):20230914, 2024.
- [5] Kai Bergermann and Francesco Tudisco. Core-periphery detection in multilayer networks. arXiv preprint, 2412.04179, 2024.
- [6] José M. Bernardo. The concept of exchangeability and its applications. *Far East Journal of Mathematical Sciences*, 4:111–122, 1996.
- [7] Stephen P. Borgatti and Martin G. Everett. Models of core/periphery structures. *Social Networks*, 21(4):375–395, 2000.
- [8] Ieva Brauksa. Use of cluster analysis in exploring economic indicator differences among regions: The case of Latvia. *Journal of Economics, Business and Management*, 1(1):42–45, 2013.
- [9] Jane Carlen, Jaume de Dios Pont, Cassidy Mentus, Shyr-Shea Chang, Stephanie Wang, and Mason A. Porter. Role detection in bicycle-sharing networks using multilayer stochastic block models. *Network Science*, 10(1):46–81, 2022.
- [10] Bolin Chen, Weiwei Fan, Juan Liu, and Fang-Xiang Wu. Identifying protein complexes and functional modules—From static PPI networks to dynamic PPI networks. *Briefings in Bioinformatics*, 15(2):177–194, 2013.

- [11] Meggan E. Craft. Infectious disease transmission and contact networks in wildlife and livestock. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1669):20140107, 2015.
- [12] Peter Csermely, András London, Ling-Yun Wu, and Brian Uzzi. Structure and dynamics of core/periphery networks. *Journal of Complex Networks*, 1(2):93–123, 2013.
- [13] Valentin Danchev and Mason A. Porter. Neither global nor local: Heterogeneous connectivity in spatial network structures of world migration. *Social Networks*, 53:4–19, 2018.
- [14] Leon Danon, Albert Díaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.
- [15] Yadolah Dodge. Mann–Whitney test. In *The Concise Encyclopedia of Statistics*, pages 327–329, Heidelberg, Germany, 2008. Springer.
- [16] Patrick Doreian. Structural equivalence in a psychology journal network. *Journal of the American Society for Information Science*, 36(6):411–417, 1985.
- [17] Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data*, 5(10):1–27, 2011.
- [18] Theodore Y. Faust, Arash A. Amini, and Mason A. Porter. Community-size biases in statistical inference in temporal networks. In preparation.
- [19] Theodore Y. Faust and Mason A. Porter. A statistical-inference method for identifying hierarchical core–periphery structure in temporal networks. In preparation.
- [20] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, 2016.
- [21] Julie Fournet and Alain Barrat. Contact patterns among high school students. *PLoS ONE*, 9(9):e107878, 2014.
- [22] Thorben Funke and Till Becker. Stochastic block models: A comparison of variants and inference methods. *PLoS ONE*, 14(4):e0215296, 2019.

- [23] Ryan J. Gallagher, Jean-Gabriel Young, and Brooke Foucault Welles. A clarified typology of core-periphery structure in networks. *Science Advances*, 7(12):eabc9800, 2021.
- [24] Laetitia Gauvin, André Panisson, and Ciro Cattuto. Detecting the community structure and activity patterns of temporal networks: A non-negative tensor factorization approach. *PLoS ONE*, 9(1):e86028, 2014.
- [25] Amir Ghasemian, Pan Zhang, Aaron Clauset, Cristopher Moore, and Leto Peel. Detectability thresholds and optimal algorithms for community structure in dynamic networks. *Physical Review X*, 6:031005, 2016.
- [26] Mark Granovetter. The strength of weak ties: A network theory revisited. *Sociological Theory*, 1:201–233, 1983.
- [27] Farnoosh Hashemi and Ali Behrouz. A unified core structure in multiplex networks: From finding the densest subgraph to modeling user engagement. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pages 1028–1039, New York, NY, USA, 2024. Association for Computing Machinery.
- [28] M. B. Hastings. Community detection as an inference problem. *Physical Review E*, 74:035102, 2006.
- [29] Adriano J. Holanda, Mariane Matias, Sueli M. S. P. Ferreira, Gisele M. L. Benedites, and Osame Kinouchi. Character networks and book genre classification. *International Journal of Modern Physics C*, 30(08):1950058, 2019.
- [30] Petter Holme. Modern temporal network theory: A colloquium. *The European Physical Journal B*, 88(9):234, 2015.
- [31] Petter Holme and Jari Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012.
- [32] Petter Holme and Jari Saramäki. *Temporal Network Theory*. Springer, Cham, Switzerland, second edition, 2023.
- [33] Xinyu Huang, Dongming Chen, Tao Ren, and Dongqi Wang. A survey of community detection methods in multilayer networks. *Data Mining and Knowledge Discovery*, 35(1):1–45, 2021.

- [34] Roberto Interdonato, Andrea Tagarelli, Dino Ienco, Arnaud Sallaberry, and Pascal Poncelet. Local community detection in multilayer networks. *Data Mining and Knowledge Discovery*, 31(5):1444–1479, 2017.
- [35] Katsuhiko Ishiguro, Tomoharu Iwata, Naonori Ueda, and Joshua Tenenbaum. Dynamic infinite relational model for time-varying relational data analysis. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23, pages 919–927, Vancouver, British Columbia, Canada, 2010. Curran Associates, Inc.
- [36] Maximilian Jerdee, Alec Kirkley, and M. E. J. Newman. Normalized mutual information is a biased measure for classification and community detection. arXiv preprint, 2307.01282, 2024.
- [37] Lucas G. S. Jeub, Prakash Balachandran, Mason A. Porter, Peter J. Mucha, and Michael W. Mahoney. Think locally, act locally: Detection of small, medium-sized, and large communities in large networks. *Physical Review E*, 91:012821, 2015.
- [38] Pengsheng Ji and Jiashun Jin. Coauthorship and citation networks for statisticians. *The Annals of Applied Statistics*, 10(4):1779–1812, 2016.
- [39] S. Jalil Kazemitabar and Arash A. Amini. Approximate identification of the optimal epidemic source in complex networks. In Naoki Masuda, Kwang-Il Goh, Tao Jia, Junichi Yamanoi, and Hiroki Sayama, editors, *Proceedings of NetSci-X 2020: Sixth International Winter School and Conference on Network Science*, pages 107–125, Cham, Switzerland, 2020. Springer International Publishing.
- [40] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer Networks. *Journal of Complex Networks*, 2(3):203–271, 2014.
- [41] Dániel Kondor, Márton Pósfai, István Csabai, and Gábor Vattay. Do the rich get richer? An empirical analysis of the Bitcoin transaction network. *PLoS ONE*, 9(2):e86197, 2014.
- [42] Olivera Kostoska, Sonja Mitikj, Petar Jovanovski, and Ljupco Kocarev. Core-periphery structure in sectoral international trade networks: A new approach to an old theory. *PLoS ONE*, 15(4):e0229547, 2020.
- [43] B. Kramer and A. MacKinnon. Localization: Theory and experiment. *Reports on Progress in Physics*, 56(12):1469–1564, 1993.

- [44] Dmitriy Kunisky, Daniel A. Spielman, Alexander S. Wein, and Xifan Yu. Statistical inference of a ranked community in a directed graph. arXiv preprint, 2411.19885, 2024.
- [45] Sang Hoon Lee, Mihai Cucuringu, and Mason A. Porter. Density-based and transport-based core-periphery structures in networks. *Physical Review E*, 89:032810, 2014.
- [46] Sang Hoon Lee, José Manuel Magallanes, and Mason A. Porter. Time-dependent community structure in legislation cosponsorship networks in the Congress of the Republic of Peru. *Journal of Complex Networks*, 5(1):127–144, 2016.
- [47] XiaoMing Li, Qiang Tian, Minghu Tang, Xue Chen, and Xiaoxian Yang. Local community detection for multi-layer mobile network based on the trust relation. *Wireless Networks*, 26(8):5503–5515, 2020.
- [48] Joseph J. Luczkovich, Stephen P. Borgatti, Jeffrey C. Johnson, and Martin G. Everett. Defining and measuring trophic role similarity in food webs using regular equivalence. *Journal of Theoretical Biology*, 220(3):303–321, 2003.
- [49] Adolphus Lye, Alice Cicirello, and Edoardo Patelli. A review of stochastic sampling methods for Bayesian inference problems. In Michael Beer and Enrico Zio, editors, *Proceedings of the 29th European Safety and Reliability Conference, ESREL 2019*, pages 1866–1873, Hannover, Germany, 2019.
- [50] Catherine Matias and Vincent Miele. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(4):1119–1141, 2016.
- [51] Michele Mazza, Guglielmo Cola, and Maurizio Tesconi. Modularity-based approach for tracking communities in dynamic social networks. *Knowledge-Based Systems*, 281:111067, 2023.
- [52] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.
- [53] Ignacio Morer, Alessio Cardillo, Albert Díaz-Guilera, Luce Prignano, and Sergi Lozano. Comparing spatial networks: A one-size-fits-all efficiency-driven approach. *Physical Review E*, 101:042301, 2020.

- [54] Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A. Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.
- [55] Zachary P. Neal. A sign of the times? Weak and strong polarization in the U.S. Congress, 1973–2016. *Social Networks*, 60:103–112, 2020.
- [56] Mark E. J. Newman. *Networks*. Oxford University Press, Oxford, UK, 2nd edition, 2018.
- [57] Kai Wang Ng, Guo-Liang Tian, and Man-Lai Tang. Dirichlet–Multinomial distribution. In *Dirichlet and Related Distributions: Theory, Methods, and Applications*, Wiley Series in Probability and Statistics, pages 199–225, Hoboken, NJ, USA, 2011. John Wiley & Sons, Ltd. Chapter 6.
- [58] Jiaqi Nie, Qi Xuan, Dehong Gao, and Zhongyuan Ruan. An effective method for profiling core–periphery structures in complex networks. *Physica A: Statistical Mechanics and its Applications*, 669:130618, 2025.
- [59] Elizabeth L. Ogburn, Oleg Sofrygin, Iván Díaz, and Mark J. van der Laan. Causal inference for social network data. *Journal of the American Statistical Association*, 119(545):597–611, 2024.
- [60] Mert Ozer, Nyunsu Kim, and Hasan Davulcu. Community detection in political twitter networks using nonnegative matrix factorization methods. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 81–88, San Francisco, CA, USA, 2016. Institute of Electrical and Electronics Engineers.
- [61] Gergely Palla, Albert-László Barabási, and Tamás Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007.
- [62] Ashwin Paranjape, Austin R. Benson, and Jure Leskovec. Motifs in temporal networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM '17*, pages 601–610, New York, NY, USA, 2017. Association for Computing Machinery.
- [63] Taeyoung Park and Seunghan Lee. Improving the Gibbs sampler. *WIREs Computational Statistics*, 14(2):e1546, 2022.
- [64] Georgios A. Pavlopoulos, Maria Secier, Charalampos N. Moschopoulos, Theodoros G. Soldatos, Sophia Kossida, Jan Aerts, Reinhard Schneider, and

- Pantelis G. Bagos. Using graph theory to analyze biological networks. *BioData Mining*, 4(1):10, 2011.
- [65] Tiago P. Peixoto. Bayesian stochastic blockmodeling. In Patrick Doreian, Vladimir Batagelj, and Anuška Ferligoj, editors, *Advances in Network Clustering and Blockmodeling*, pages 289–332, Hoboken, NJ, USA, 2019. John Wiley & Sons, Ltd. Chapter 11.
- [66] Tiago P. Peixoto. *Descriptive vs. Inferential Community Detection in Networks: Pitfalls, Myths and Half-Truths*. Elements in the Structure and Dynamics of Complex Networks. Cambridge University Press, Cambridge, UK, 2023.
- [67] Tiago P. Peixoto and Martin Rosvall. Modelling sequences and temporal networks with dynamic community structures. *Nature Communications*, 8:582, 2017.
- [68] Austin Polanco and M. E. J. Newman. Hierarchical core–periphery structure in networks. *Physical Review E*, 108:024311, 2023.
- [69] Mason A. Porter, Jukka-Pekka Onnela, and Peter J. Mucha. Communities in networks. *Notices of the American Mathematical Society*, 56:1082–1097, 1164–1166, 2009.
- [70] Christian P. Robert and George Casella. The Metropolis–Hastings algorithm. In *Monte Carlo Statistical Methods*, pages 267–320, Heidelberg, Germany, 2004. Springer. Chapter 7.
- [71] Puck Rombach, Mason Porter, James Fowler, and Peter Mucha. Core-periphery structure in networks (revisited). *SIAM Review*, 59:619–646, 2017.
- [72] Giulio Rossetti and Rémy Cazabet. Community discovery in dynamic networks: A survey. *ACM Computing Surveys*, 51(2):35, 2018.
- [73] Ryan A. Rossi and Nesreen K. Ahmed. Role discovery in networks. *IEEE Transactions on Knowledge and Data Engineering*, 27(4):1112–1131, 2015.
- [74] Ryan A. Rossi, Brian Gallagher, Jennifer Neville, and Keith Henderson. Modeling dynamic behavior in large evolving graphs. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM ’13, pages 667–676, New York, NY, USA, 2013. Association for Computing Machinery.

- [75] Martin Rosvall and Carl T. Bergstrom. Mapping change in large networks. *PLoS ONE*, 5(1):e8694, 2010.
- [76] William G. Roy. The unfolding of the interlocking directorate structure of the United States. *American Sociological Review*, 48(2):248–257, 1983.
- [77] Sudhir Saxena, K. Santhanam, and Aparna Basu. Application of Social Network Analysis (SNA) to terrorist networks in Jammu & Kashmir. *Strategic Analysis*, 28(1):84–101, 2004.
- [78] Jimeng Sun, Christos Faloutsos, Spiros Papadimitriou, and Philip S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 687–696, New York, NY, USA, 2007. Association for Computing Machinery.
- [79] Yizhou Sun, Jie Tang, Jiawei Han, Manish Gupta, and Bo Zhao. Community evolution detection in dynamic heterogeneous information networks. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, MLG '10, pages 137–146, New York, NY, USA, 2010. Association for Computing Machinery.
- [80] Shazia Tabassum, Fabiola S. F. Pereira, Sofia Fernandes, and João Gama. Social network analysis: An overview. *WIREs Data Mining and Knowledge Discovery*, 8(5):e1256, 2018.
- [81] Pinghui Wang, John C.S. Lui, Don Towsley, and Junzhou Zhao. Minfer: A method of inferring motif statistics from sampled edges. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 1050–1061, 2016.
- [82] K. Hunter Wapman, Sam Zhang, Aaron Clauset, and Daniel B. Larremore. Quantifying hierarchy and dynamics in us faculty hiring and retention. *Nature*, 610(7930):120–127, 2022.
- [83] Todd Waskiewicz. Friend of a friend influence in terrorist social networks. In Hamid R. Arabnia, David de la Fuente, Elena B. Kozerenko, Peter M. LaMonica, Raymond A. Liuzzi, Jose A. Olivas, Ashu M. G. Solo, and Todd Waskiewicz, editors, *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, Las Vegas, Nevada, USA, 2012. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

- [84] Harrison C. White, Scott A. Boorman, and Ronald L. Breiger. Social structure from multiple networks. I. Blockmodels of roles and positions. *American Journal of Sociology*, 81(4):730–780, 1976.
- [85] Stephen Wyn Williams. Internal colonialism, core-periphery contrasts and devolution: An integrative comment. *Area*, 9(4):272–278, 1977.
- [86] Cecil L. Willis and Stephen J. McNamee. Social networks of science and patterns of publication in leading sociology journals, 1960 to 1985. *Knowledge*, 11(4):363–381, 1990.
- [87] Eric Yanchenko and Srijan Sengupta. Core-periphery structure in networks: A statistical exposition. *Statistics Surveys*, 17:42–73, 2023.
- [88] Tianbao Yang, Yun Chi, Shenghuo Zhu, Yihong Gong, and Rong Jin. Detecting communities and their evolutions in dynamic social networks—A Bayesian approach. *Machine Learning*, 82(2):157–189, 2011.
- [89] Xiao Zhang, Travis Martin, and M. E. J. Newman. Identification of core-periphery structure in networks. *Physical Review E*, 91:032803, 2015.