2018

Statistical methods for topology inference, denoising, and bootstrapping in networks

BOSTON UNIVERSITY

GRADUATE SCHOOL OF ARTS AND SCIENCES

Dissertation

STATISTICAL METHODS FOR TOPOLOGY INFERENCE, DENOISING, AND BOOTSTRAPPING IN NETWORKS

by

XINYU KANG

B.S., Hong Kong Baptist University, 2010
M.S., University of Illinois at Urbana-Champaign, 2011

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2018

Approved by

First Reader

Eric D. Kolaczyk, PhD
Professor of Mathematics and Statistics

Second Reader

Daniel Sussman, PhD
Assistant Professor of Mathematics and Statistics

Third Reader

Luis Carvalho, PhD
Associate Professor of Mathematics and Statistics

Acknowledgments

I would like to express my greatest gratitude to my advisor Prof. Eric D. Kolaczyk, without whose guidance and endless support, it is impossible for me to be at this stage. I have always been inspired by his remarkable dedication to his craft and benefited from his great vision.

Thank you to Prakash Balachandran, Apratim Ganguly, Yaonan Zhang, Heather Shappell and Jun Li, the best group members and colleagues I could have asked for, and thank you Ian Johnston and Shuyang Bai, for being my roommates here at BU. My thanks also goes to other professors, friends and staffs in the math department.

Finally, I would like to give special thanks to my parents and my family for their support. Specially thanks to my wife Hui Xu, for her encouragement and unconditional love.

STATISTICAL METHODS FOR TOPOLOGY INFERENCE, DENOISING, AND BOOTSTRAPPING IN NETWORKS

(Order No.)

XINYU KANG

Boston University, Graduate School of Arts and Sciences, 2018

Major Professor: Eric D. Kolaczyk, Professor of Mathematics and Statistics

ABSTRACT

Quite often, the data we observe can be effectively represented using graphs. The underlying structure of the resulting graph, however, might contain noise and does not always hold constant across scales. With the right tools, we could possibly address these two problems. This thesis focuses on developing the right tools and provides insights in looking at them. Specifically, I study several problems that incorporate network data within the multi-scale framework, aiming at identifying common patterns and differences, of signals over networks across different scales. Additional topics in network denoising and network bootstrapping will also be discussed.

The first problem we consider is the connectivity changes in dynamic networks constructed from multiple time series data. Multivariate time series data is often non-stationary. Furthermore, it is not uncommon to expect changes in a system across multiple time scales. Motivated by these observations, we in-

corporate the traditional Granger-causal type of modeling within the multi-scale framework and propose a new method to detect the connectivity changes and recover the dynamic network structure.

The second problem we consider is how to denoise and approximate signals over a network adjacency matrix. We propose an adaptive unbalanced Haar wavelet based transformation of the network data, and show that it is efficient in approximation and denoising of the graph signals over a network adjacency matrix. We focus on the exact decompositions of the network, the corresponding approximation theory, and denoising signals over graphs, particularly from the perspective of compression of the networks. We also provide a real data application on denoising EEG signals over a DTI network.

The third problem we consider is in network denoising and network inference. Network representation is popular in characterizing complex systems. However, errors observed in the original measurements will propagate to network statistics and hence induce uncertainties to the summaries of the networks. We propose a spectral-denoising based resampling method to produce confidence intervals that propagate the inferential errors for a number of Lipschitz continuous network statistics. We illustrate the effectiveness of the method through a series of simulation studies.

Contents

List of Tables

List of Figures

List of Abbreviations

DLPFC	.....	Dorsolateral prefrontal cortex
DTI	.....	Diffusion tensor imaging
EEG	.....	Electroencephalography
EM	.....	Expectation maximization (algorithm)
FEF	.....	Frontal eye fields
MEG	.....	Magnetoencephalography
fMRI	.....	Functional magnetic resonance imaging
MT+	.....	Middle temporal +
RDP	.....	Recursive dyadic partition
ROI	.....	Region of interest
RP	.....	Recursive partition
rP-VAR(p)	.....	Restricted piecewise-vectorautoregression model of order p
SPL	.....	Superior parietal lobule
TGUH	.....	Tail-greedy unbalanced Haar
VAR	.....	Vector autoregression
VIP	.....	Vental intraparietal sulcus
V3a	.....	Third visual cortex

Chapter 1

Introduction

1.1 Network science and network data

Network science or network theory studies the relations and interactions among entities of interest. Formally, a network is defined as a graph $G = (V, E)$, where V is the set of vertices or nodes, and E is the set of edges or links that describe the connections between nodes. The edges can be directed or undirected, weighted and unweighted, depending on the context and the nature of the connections.

Examples of networks include but are not limited to the Internet, citation networks, food webs, protein-protein interaction networks, financial economic networks and social networks. Loosely speaking, these networks can be characterized into four classes: technological, informational, social, and biological [46][64]. Technological networks include communication network, for example the internet, transportation network, such as highway and airline route networks, and energy networks, for instance, the network for gas or heat delivery. One point of interest in technical networks is the flow across the network. Informational networks are the networks to describe the relationships between elements of information, for example, citation networks, where documents are the vertices and links from one document to other documents form the directed edges. Social networks usually characterize the interactions among a group of people or a group of animals. The study of social networks is of particular interest to researchers in social sciences

and business area. Some interesting questions are "Does similar people share similar hobby or shopping habits?" "Who is the most influential person among a social group?" "How does rumor spread among social networks?" Biological networks are the network inferred from biological entities that used to represent the system. Examples are the gene-regulatory network, where the genes are the vertices and the regulations or interactions are the edges, and the functional brain imaging such as the EEG/MEG connectivity network, where regions of interest are the vertices and the inferred associations are indicated by edges. Biological networks often contain different information at different scales. For example, in functional brain imaging, causal relations and co-activation patterns among regions of interest could be quite different at different temporal resolution. Modeling this type of network, and sometime even correctly specifying it, is a challenging problem. For more examples of networks, please refer to [46][47].

Statistical analyses of a network can provide us more critical insight on the structure, flow, dynamics and evolution of the network system. However, in the real world, networks may exhibit variations and changes, so that a single static network is not sufficient to represent the system, regardless of whether the network is observed or inferred. Also, the construction of networks can be subjective and contain inevitable noise, which induces uncertainties to the summaries of the networks. In this thesis, I study these two problems that are quite common in the network field. In particular, this thesis discusses three interesting questions. The first two questions are inspired from the study of neuro-image data, where dynamic evolution and changes of structures of the functional connectivity networks are studied. The third question we study provides a practical solution for quantifying uncertainties of the summaries of networks with simple flipped errors. Section 1.2, 1.3, and 1.4 offer reviews of previous work.

1.2 Prior work in dynamic network modeling

There are multiple ways to represent and work with a dynamic network. For example, one could study a cumulative version of the network, for which a node or edge present in $G(s)$ is also presented in all $G(t)$ for $t > s$ [54]. Another option is to work with snapshots of graphs. The temporal resolution of the system plays an important role in determining the appropriate model to use. In the case when continuous information is available, more specific models can be used[67]. In other cases, some types of smoothing across time are often desirable, either through the use of a moving window[51], or a moving average[41].

A number of works in recent years focus on modeling multivariate time series using causal network types of models. A common theme among these is to generalize the work of [60], who show that the Lasso can consistently recover the neighborhood structure of a Gaussian graphical model in high-dimensional settings under appropriate assumptions. Seminal examples of such extensions include [10], where they assume the time series are stationary and carry out variable selection using group-lasso principles; and [5], where they estimate the network Granger causality for panel data using the group-lasso. Similarly, in the work by [4], networks are defined and inferred through use of the long-run partial correlation matrix between multiple time series. For non-stationary multivariate time series processes, [57] use time-varying auto-regressive models with adaptively chosen – but fixed – windows. These latter are applied to functional MRI data.

While we make use of ideas similar to those above, our approach is significantly different from those proposed previously in the sense that we incorporate them within a multi-scale framework. Multi-resolution analysis was formally proposed by [59] and others in the late 1980's and has been known for mathematically elegant, computationally efficient and often domain-specific representations

of data that are inhomogeneous in their support. While there is by now a vast literature on the topic of multiscale statistical modeling, with literally scores of representations for standard signal and image analysis applications alone, a key representation is that of recursive dyadic partitioning. A fundamental result from [24] relates the method of recursive dyadic partitioning and the selection of a best-orthonormal basis, where the basis is selected from a class of unbalanced Haar wavelets. The partition-based multi-scale method has proven to be particularly natural and useful in extending wavelet-like ideas to nontraditional settings, for example, in the context of generalized linear models, irregular spatial domains, etc. – see [48], [58], and [80], for instance. For a recent survey of statistical methods for network inference from time series, in general, see [7, Sec 4.2][45].

1.3 Prior work in graph signal processing

Other than traditional use for modeling spatial-temporal data, the multiscale framework gained a resurgence of interest in recent years in the emerging field of graph signal processing. Wavelet-related methods have been used for classical signal processing problems as they were invented for capturing signal changes both locally and globally. However, unlike the traditional signal processing problem in the one dimensional case, the graph signal is a set of measurements residing on a set of vertices and additional structure is induced. As a result, the extension from classical signal processing to graph signal processing does not come without a cost. One challenge the graph signal processing faces is the lack of mathematical tools for processing data.

Previous studies extend classical signal processing to graph signal processing. In graph signal processing, the signal is defined over the set of vertices [70][73][71]. Choice of tools in the classical signal processing can be applied on the adjacency

matrix $\mathbf{W}$ or on the Laplacian matrix $\mathbf{L}$ (including the normalized laplacian matrix), where the laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Here $\mathbf{D}$ is the degree matrix. Using the adjacency matrix $\mathbf{W}$ reduces the shift from classical (discrete) signal processing and applies to both directed and undirected graph and using the laplacian matrix one can take advantages of the good mathematical properties of the laplacian matrix[66].

A critical problem, and maybe the most important one, is to find an appropriate representation for graph signals that have desirable properties. Wavelet-methods seem to enjoy advantages in orthogonality, sparsity and localization since they were invented. Various previous studies focus on either wavelet transforms *on* networks and/or wavelet transforms *of* networks. The first category includes those transformations based on the graph Laplacian, relevant to analysis of signals over networks, e.g., see [19] and [39]. Earlier work of Crovella and Kolaczyk [21] extends the Mexican hat wavelet to unweighted graphs and uses it to analyze computer network traffic. Also related is the work by Gavish, Nadler and Coifman [34], who develop various results on graphs that possess a hierarchical tree structure, and the work by Irion and Saito [43], where they compute graph basis dictionaries using graph Laplacian eigen transforms and generalized Haar-Walsh transforms.

The second category focuses more exclusively on transforms of the graph itself. Examples include work by Murtagh [63], who develops an invertible wavelet transform based on hierarchical clustering using Ward's criterion, and the work by Lee, Nadler and Wasserman[53], who propose an attribute-based construction of adaptive multi-scale hierarchical trees. Other methods of this type incorporate ideas from matrix compression and factorization. For example, see [49], where they propose a multi-scale way of doing matrix factorization, which effectively

decomposes large matrices using a series of transformation matrices that capture structures at different scales. While a large part of the work mentioned above focus on representations using basis functions selected from frequency components of the graph signal, other recent works explore representations of piecewise smooth graph signals[72][17][83]. The graph signal processing area is a much busier area than the bibliography suggests here. For a more complete overview of the field, please see [73][66].

Our method extends the tail-greedy unbalanced Haar transformation (TGUH) to the network field. The TGUH was originally proposed by Fryzlewicz [31] for the one-dimensional signal plus noise model. The algorithm results in a nonlinear but conditionally orthonormal, multiscale decomposition of the data with respect to an adaptively chosen unbalanced Haar wavelet basis. Related work also includes the work by Fryzlewicz and Timmermans [32], where an adaptive Haar-type of transformation is used for image compression and denoising.

1.4 Prior work in network denoising

In areas ranging from social networks, where nodes are people and edges are connections or communications[25][78][79], to biological network, where nodes are biological entities and edges are functional connectivities[68][76], the edges we observe or infer are often imperfect and contain noises. The simplest case is the flip of edges, where we call it Type I error and Type II error. In the network context, a Type I error is declaring an edge when none exists and a Type II error is omitting an edge when it exists.

There has been a number of studies focusing on making statistical inference on networks with observed errors using both parametric and non-parametric model. On the parametric side, Butts [13] proposed a Bayesian approach to model

certain kinds of error which can be used generically to assess posterior uncertainty in some classical network measure. Priebe et al. [68] propose a model for vertex classification. Newman [65] proposed a likelihood based parametric model using EM algorithm to estimate true network structures. However, these method require additional assumptions on the distribution of the network linkage. On the non-parametric side, Chatterjee[16] took the matrix completion point of view and introduced a SVD-based procedure to estimate large dense matrix, which also includes dense network adjacency matrix.

1.5 Contribution and organization of the dissertation

Our main contributions in this thesis are three fold. We provide solutions to three related network inference problems. The organization of the remainder of this dissertation is as follows.

The first contribution in this thesis is in chapter 2, where we present a partition-based multi-scale dynamic causal network model, and a corresponding method of network topology inference, that captures the dynamics of a system in a manner sensitive to changes at multiple time scales, while encouraging sparsity of network connectivity. There are three key elements in the framework: (i) the partition the non-stationary time axis into blocks at various scales, with independent, stationary VAR models indexed by blocks; (ii) to prevent overfitting, we impose a counting penalty to penalize the number of blocks used; and (iii) we do neighborhood selection within each block using a group-lasso type of estimator.

The second contribution in this thesis is in chapter 3, where we propose an adaptive unbalanced Haar wavelet based transformation of the network data, and show that it is efficient in approximation and denoising of the graph signals over

a network adjacency matrix. The thesis focuses on the exact decompositions of the network, the corresponding approximation theory, and denoising signals over graphs, particularly from the perspective of compression of the networks. A real data application on denoising EEG signals over a DTI network is also provided.

The third contribution in this thesis is in chapter 4, where the we propose a spectral-denoising based resampling method to produce confidence intervals that propagate the inferential errors for a number of Lipschitz continuous network statistics. The effectiveness of the proposed method is demonstrated through a series of simulation studies.

In chapter 5, I summarize the main contributions of the dissertation, and propose an overview of possible future research directions.

Chapter 2

Dynamic Networks with Multi-scale Temporal Structure

2.1 Introduction

The automated, simultaneous monitoring of each unit in a large complex system has become commonplace. Frequently the data observed in such a system is in the form of a high dimensional multivariate time series. Domain areas where such a paradigm is particularly pertinent include computational neuroscience (e.g., temporal imaging across voxels or brain regions) and finance (e.g., investment returns across stocks or levels of lending among central banks). The combination of system and time series in these settings suggests a role for dynamic network modeling, a quickly developing area of study in the field of network analysis.

As the basic object of treatment in this paper we consider a multivariate time series, $(X_t(1), \cdots, X_t(N))$, observed at each of N units at times $t = 1, \ldots, T$, as a set of measurements from across a system. We will use a graph $G = (V, E)$ to describe the conditional dependencies among the time series across the system. Here $V = \{1, \ldots, N\}$ are vertices corresponding to the N units in the system, and E is the collection of vertex pairs joined by edges. Given data, we seek to select an appropriate choice of G that best characterizes the system, using techniques of statistical modeling and inference. This task is known as network topology inference [46, Ch 7.3]. The notion of association used in this paper is a type

of partial correlation, analagous to that underlying so-called Granger causality ([36]). Granger causal types of models have been widely utilized in financial economics – see [38], [40] and [74], for example – and in biological studies – see [61], [12] for instance.

Granger causal models traditionally assume a stationary time series and take a vector-autoregressive (VAR) form. Here we adopt a restricted-VAR(p) model, defined as a VAR model without the self driven components:

$$X_t(u) = \sum_{v \in V \setminus \{u\}} \sum_{\ell=1}^{p} X_{t-\ell}(v)\theta^{(\ell)}(u,v) + \epsilon_t(u),$$

where $\theta^{(\ell)}(u,v)$ collects the influence of the node v on node u at lag ℓ and $\epsilon_t(u)$ is independent Gaussian white noise. It is said that $X(v)$ Granger causes $X(u)$ if and only if $\theta^{(\ell)}(u,v) \neq 0$ for some $\ell = 1, \cdots, p$. We use the term 'restricted' in describing this model because we restrict $\theta^{(\ell)}(u,u)$ to be 0 for all u, ℓ. This requirement is made for notational convenience, and without loss of generality, in that it essentially assumes the self-driven component has been removed and that our network characterizes only relationships between distinct nodes. The notion of 'network' in this framework is made precise through graphs defined as a function of the underlying graphical model. That is, through conditional independence relations, coded in one-to-one correspondence with patterns of non-zero elements among the $\theta^{(\ell)}(u,v)$. Specifically, $G = (V, E)$ is a directed graph with an edge from v to u if and only if $\|\boldsymbol{\theta}(u,v)\|_2 \neq 0$, where $\boldsymbol{\theta}(u,v) = \left(\theta^{(1)}(u,v), \cdots, \theta^{(p)}(u,v)\right)'$.

Multivariate time series data is often non-stationary. Furthermore, it is not uncommon to expect changes in a system across multiple time scales. For example, it is widely recognized that financial time series of quantities like equity, interest, and credit can exhibit volatility across multiple scales (e.g., [28]). Similarly, it is believed that neuronal dynamics within the cerebral cortex in the brain

interact with anatomical connectivity in such a way as to produce functional connectivity relationships between brain regions at multiple time scales ([42]). These observations suggest the need for a notion of multi-scale analysis when doing network-based modeling of multivariate time series in systems like these. However, while temporal multi-scale analysis is a concept well-established in time series analysis, it does not appear to have yet emerged in network modeling.

Motivated by the elements of the above discussion, we focus in this paper on the problem of detecting dynamic connectivity changes across multiple time scales in a network-centric representation of a system, based on multivariate time series observations. Our approach combines the traditional Granger causal type of modeling with partition-based multi-scale modeling. We adopt a change point perspective, so that our model class consists of concatenations of restricted-VAR(p) models, each with its own $\boldsymbol{\theta}$ constant over a given interval of time. The result is then a time-indexed directed graphical model, from which we define a dynamic network $G_t = (V, E_t)$, in analogy to the stationary case. Our goal is then to infer the change points distinguishing the stationary intervals and the corresponding edge sets E_t.

A number of works in recent years have focused on modeling multivariate time series using causal network types of models. A common theme among these is to generalize the work of [60], who show that the Lasso can consistently recover the neighborhood structure of a Gaussian graphical model in high-dimensional settings under appropriate assumptions. Seminal examples of such extensions include [10], where they assume the time series are stationary and carry out variable selection using group-lasso principles; and [5], where they estimate the network Granger causality for panel data using the group-lasso. Similarly, in the work by [4], networks are defined and inferred through use of the long-run partial corre-

lation matrix between multiple time series. For non-stationary multivariate time series processes, [57] use time-varying auto-regressive models with adaptively chosen – but fixed – windows. These latter are applied to functional MRI data.

While we make use of ideas similar to those above, our approach is significantly different from those proposed previously in the sense that we incorporate them within a multi-scale framework. Multi-resolution analysis was formally proposed by [59] and others in the late 1980's and has been known for mathematically elegant, computationally efficient and often domain-specific representations of data that are inhomogeneous in their support. While there is by now a vast literature on the topic of multiscale statistical modeling, with literally scores of representations for standard signal and image analysis applications alone, a key representation is that of recursive dyadic partitioning. A fundamental result from [24] relates the method of recursive dyadic partitioning and the selection of a best-orthonormal basis, where the basis is selected from a class of unbalanced Haar wavelets. The partition-based multi-scale method has proven to be particularly natural and useful in extending wavelet-like ideas to nontraditional settings, for example, in the context of generalized linear models, irregular spatial domains, etc. – see [48], [58], and [80], for instance. For a recent survey of statistical methods for network inference from time series, in general, see [7, Sec 4.2].

Our main contribution in this paper is to present a partition-based multi-scale dynamic causal network model, and a corresponding method of network topology inference, that captures the dynamics of a system in a manner sensitive to changes at multiple time scales, while encouraging sparsity of network connectivity. There are three key elements in the framework: (i) we partition the non-stationary time axis into blocks at various scales, with independent, stationary VAR models indexed by blocks; (ii) to prevent overfitting, we impose a counting penalty to

penalize the number of blocks used; and (iii) we do neighborhood selection within each block using a group-lasso type of estimator.

This paper is organized as follows. In Section 2, we provide the details our partition-based dynamic multi-scale network model and methodology. In Section 3, we present several characterizations of theoretical properties of our estimator. The broad potential impact of our method is demonstrated in Section 4, through the use of both simulated data and a magnetoencephalography (MEG) data set. Technical proofs are provided in the appendix. Code implementing the methodology proposed in this paper is available from `https://github.com/KolaczykResearch/MS-Dyn-Networks-Code`.

2.2 Partition-based multi-scale dynamic network models

In this section we define the class of dynamic network models developed in this paper, we describe our proposed approach to network inference within this class, and we summarize the implementation of this approach in the form of an algorithm.

2.2.1 Piecewise vector autoregressive models

We are interested in non-stationary multivariate time series, as the stationarity assumption required by traditional vector autoregressive modeling is overly restrictive in the types of financial and biological applications motivating our work. Accordingly, we define a class of restricted piece-wise vector autoregressive models. These models are of order p [rP-VAR(p)] and break the non-stationary time series into an unknown number of M stationary blocks, with a stationary restricted VAR(p) model within each block.

More specifically, we equip the parameters in our previously defined restricted

VAR(p) model with a time index:

$$X_t(u) = \sum_{v \in V \setminus \{u\}} \sum_{\ell=1}^{p} X_{t-\ell}(v)\, \theta_t^{(\ell)}(u,v) + \epsilon_t(u) \ . \tag{2.1}$$

Next we restrict the coefficient vectors $\boldsymbol{\theta}_t(u,v) = \left(\theta_t^{(1)}(u,v), \cdots, \theta_t^{(p)}(u,v)\right)'$ to be constant within each of M blocks defined by change points with $\tau_0 = 0$ and $\tau_{M+1} = T$. Finally, we assume independence of the multivariate time series across blocks. We then capture the evolving dependency structure of the data using a time-varying directed graph $G = (V, E_t)$ with an edge from $v \to u$ if and only if $\|\boldsymbol{\theta}_t(u,v)\|_2 \neq 0$.

Some of these choices could be relaxed, at the expense of a nontrivial increase in complexity of both computation and exposition. The assumption of independence between blocks could be relaxed to allow for weak dependence over p time steps just prior to and after each changepoint, following the suggestion in [22, Remark 1]. Additionally, we assume the number of lags p is fixed and known. In contrast, an unknown value of p in principle could be incorporated into our framework, with selection made through an additional penalty term.

To organize the collection of blocks defining our class of rP-VAR(p) models, we use the notion of recursive partitioning. This choice is both consistent with our goal of capturing multi-scale structure (as described above) and facilitates the development of sensible algorithms for computational purposes. We will consider two types of partitioning: recursive dyadic partitioning and (general) recursive partitioning. Without loss of generality, we consider partitioning restricted to the unit interval $(0, 1]$ interchangeably with partitioning of the interval $(0, T]$. A partition $\mathcal{P}$ of $(0, 1]$ is a decomposition of the latter into a collection of disjoint subintervals whose union is the unit interval. In our treatment we restrict attention to partitions of finite cardinality.

Both recursive dyadic partitioning and recursive partitioning produce partitions $\mathcal{P}$ by recursively partitioning the unit interval. They differ only in the rule defining the choice of partitions that may be produced at each iteration, with that for the former being more restrictive than that for the latter. Under recursive dyadic partitioning, starting with the unit interval, we recursively split some previously resulting interval into two sub-intervals of equal length. Under recursive partitioning more generally, the restriction to dyadic subintervals is removed. Under both approaches, partitioning is done only up to the resolution of the data. Therefore, with T observation times, partitioning is done only at the points $\{i/T\}_{i=1}^{T-1}$, and only up to a total of T subintervals. Under recursive dyadic partitioning, we require that the number of observations $T = 2^J$ be a power of two.

Let $\mathcal{P}^*_{D_y}$ denote the complete recursive dyadic partition (with the dependence on T suppressed for notational convenience), and $\mathcal{P}^*$, a complete recursive partition. Additionally, denote by $\mathcal{P} \preceq \mathcal{P}^*_{D_y}$ (respectively, $\mathcal{P} \preceq \mathcal{P}^*$) a subpartition of $\mathcal{P}^*_{D_y}$ (respectively, $\mathcal{P}^*$), i.e., as one of the partitions defined through the process of successive refinement from $(0, 1]$ to $\mathcal{P}^*_{D_y}$ (respectively, $\mathcal{P}^*$). This notation helps emphasize one of the key advantages of the partition-based perspective, i.e., that algorithms to search efficiently over model spaces indexed by these partition classes can be designed to do so in $\mathcal{O}(T)$ and $\mathcal{O}(T^3)$ computational complexity, respectively, using dynamic programming principles. See [48]. The advantage of recursive dyadic partitioning over recursive partitioning therefore typically is in computational cost. We will define a class of rp-VAR(p) models indexed by these partition classes and propose algorithms for model selection that exploit the accompanying dynamic programming principles.

2.2.2 Network Inference

The graphs G corresponding to the restricted piece-wise VAR(p) class of models we have introduced can be thought of as a union of the neighborhoods surrounding each node u. And, in fact, we will infer the topology of the network G neighborhood by neighborhood.

Consider, for example, the cartoon illustration in Figure 2·1 where, without loss of generality, the focus is on the local neighborhood of a node/series u and $T = 160$ for illustration. From time $[0, 60)$, each of the four other nodes B, D, C, and E Granger causes u. From time $[60, 80)$, only node B Granger causes u, and for the rest of the time, B and D Granger cause u. Under our proposed approach, we estimate the times τ_m at which the changes happened. Given the estimated change points, we then infer the neighborhood structure during the time interval $[0, \hat{\tau}_1)$, and then $[\hat{\tau}_1, \hat{\tau}_2)$, and so on. Put simply, our approach is to estimate the change-points and the neighborhood structures within each stationary time-interval defined by those change-points, where the changepoints are defined through either a recursive dyadic partition or a recursive partition. We describe each of these two cases in turn below.

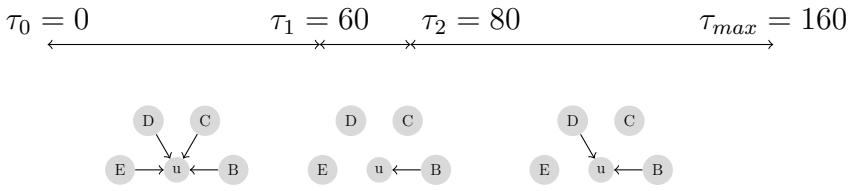


Figure 2·1: Cartoon version of the underlying network structure.

Suppose that our changepoints τ_i are restricted to correspond to the boundaries of some recursive dyadic partition. For a given node u, we estimate the vector $\boldsymbol{\theta} \equiv \left(\theta_{t_i}^{(\ell)}(u, v)\right)$, defined for all nodes $v \in V \setminus \{u\}$ and at all times $t_i = i/T$ where $i = 1, \ldots, T$, by choosing some optimal member from the classes rP-VAR(p) de-

fined by all possible partitions $\mathcal{P} \preceq \mathcal{P}^*_{D_y}$ of the unit interval. Formally, we define the space of all possible values of $\boldsymbol{\theta}$

$$\Gamma^{(N-1)p}_{RDP} \equiv \left\{ \boldsymbol{\theta} \,\middle|\, \theta^{(\ell)}_t(u,v) = \beta^{(\ell)}_0(u,v) + \sum_{I \in \ell_{NT}(\mathcal{P})} \beta^{(\ell)}_I(u,v) h_I(t) \quad \forall\, \ell, v, \text{for some } \mathcal{P} \preceq \mathcal{P}^*_{D_y} \right\}, \tag{2.2}$$

where $\mathcal{P}$ is a partition common to all coefficient functions $\theta^{(\ell)}_t(u,v)$ across nodes v and lags ℓ, for each fixed u. In this expression, $\ell_{NT}(\mathcal{P})$ is the set of all non-terminal (NT) intervals encountered in the construction of $\mathcal{P}$, while $\beta^{(\ell)}_0(u,v)$ and $\beta^{(\ell)}_I(u,v)$ are the (non-zero) coefficients in a reparameterization of $\theta^{(\ell)}_t(u,v)$ with respect to the unique (dyadic) Haar wavelet basis $\{h_I\}_{I \in \ell_{NT}(\mathcal{P}^*_{Dy})}$ associated with the complete recursive dyadic partition $\mathcal{P}_{Dy}$. In particular, a wavelet h_I has as its support the interval I, and is proportional to the values 1 and -1 on the two subintervals defined by a split at the midpoint of I. See [24] or [48], for example, for details on this correspondence between recursive dyadic partitions and classical Haar wavelet bases. It is this correspondence that makes explicit the multiscale nature of our approach.

Based on this model class, we define a complexity-penalized estimator $\hat{\boldsymbol{\theta}}_{RDP}$ of $\boldsymbol{\theta}$ as follows:

$$\hat{\boldsymbol{\theta}}_{RDP} \equiv \underset{\tilde{\boldsymbol{\theta}} \in \Gamma^{(N-1)p}_{RDP}}{\arg\min} \left\{ -\log p\left(\mathbf{X}(u) | \mathbf{X}(-u), \tilde{\boldsymbol{\theta}}\right) + 2 \sum_{v \in V \setminus \{u\}} \text{Pen}_{RDP}(\tilde{\boldsymbol{\theta}}(u,v)) \right\}. \tag{2.3}$$

Here $\mathbf{X}(-u)$ is the lagged design matrix of dimension $T \times (N-1)p$ based on the observed time series information for all nodes except u. That is, we define $\mathbf{X}(-u) = (\mathbf{X}(1), \cdots, X(u-1), X(u+1), \cdots, \mathbf{X}(N))$, with each $\mathbf{X}(\cdot)$ a $T \times p$ matrix defined as $\mathbf{X}(\cdot) = (\mathbf{X}_{-1}(\cdot), \cdots, \mathbf{X}_{-p}(\cdot))$, where $\mathbf{X}_{-\ell}(\cdot)$ contains the lagged

observations $\mathbf{X}_{-\ell}(\cdot) = (X_{T-\ell}(\cdot), \cdots, X_{-\ell+1}(\cdot))'$. The function $\text{Pen}_{RDP}(\tilde{\boldsymbol{\theta}}(u,v))$ is the penalty imposed for incorporating node v into the model.

Now consider the case where the network changepoints τ_i are restricted to correspond to the boundaries of some arbitrary (i.e., non-dyadic) recursive partition. Define $\mathcal{L}$ to be the library of all $(T-1)!$ possible complete recursive partitions $\mathcal{P}^*$, and let

$$\Gamma_{RP}^{(N-1)p} \equiv \left\{ \boldsymbol{\theta} \,\middle|\, \theta_t^{(\ell)}(u,v) = \beta_0^{(\ell)}(u,v) + \sum_{I \in \ell_{NT}(\mathcal{P})} \beta_I^{(\ell)}(u,v) h_I(t) \;\forall \ell, v, \text{ for some } \mathcal{P} \preceq \mathcal{P}^*, \mathcal{P}^* \in \mathcal{L} \right\} . \quad (2.4)$$

Here $\{h_I\}_{I \in \ell_{NT}(\mathcal{P}^*)}$ is the unique (unbalanced) Haar wavelet basis corresponding to a given complete recursive partition $\mathcal{P}$. As in the case of the classical dyadic Haar basis, there will be T piecewise constant basis functions for T time points, each indexed according to its support interval I and proportional in value to 1 or -1 on two subintervals (except for one 'father' wavelet, defined to capture the average of $\theta_t^{(\ell)}(u,v)$ over $(0,T]$). But, unlike before, the subintervals defining these wavelets are not necessarily of equal length. This definition allows, for example, for the representation of non-dyadic changepoints in a potentially more efficient manner (i.e., using fewer recursive splits). See [48] for details.

Analogous to the dyadic case, our estimator defined under recursive partitioning is given by:

$$\hat{\boldsymbol{\theta}}_{RP} \equiv \underset{\tilde{\boldsymbol{\theta}} \in \Gamma_{RP}^{(N-1)p}}{\arg\min} \left\{ -\log p\left(\mathbf{X}(u) | \mathbf{X}(-u), \tilde{\boldsymbol{\theta}}\right) + 2 \sum_{v \in V \setminus \{u\}} \text{Pen}_{RP}(\tilde{\boldsymbol{\theta}}(u,v)) \right\} . \quad (2.5)$$

This is a maximum complexity-penalized likelihood estimator of $\boldsymbol{\theta}$ defined on a much broader space. It includes all possible partitions that divide the unit interval into $M \leq T$ blocks, where sub-intervals need not necessarily be of equal

size. This increase in richness of representation, however, will be seen to come at a computational cost.

The penalty function used to define these two estimators is described as follows. Define the p-length vector $\boldsymbol{\theta}_I(u,v)$ to be the collection of (fixed) values $\theta_t^{(\ell)}(u,v)$ over all lags $\ell = 1,\ldots,p$ for $t \in I$. For recursive partitioning, we then define the penalty of incorporating a given node v into the model to be

$$\mathrm{Pen}_{RP}(\boldsymbol{\theta}(u,v)) = \frac{3}{2}\#\{\mathcal{P}(\boldsymbol{\theta})\}\log T + \lambda \sum_{I\in\mathcal{P}(\boldsymbol{\theta})} \|\boldsymbol{\theta}_I(u,v)\|_2 \ . \tag{2.6}$$

For recursive dyadic partitioning, we replace the value $3/2$ by $1/2$, indicating that we penalize less severely in the simpler model class.

Note that this penalty is composed of two parts. In the first part, $\#\{\mathcal{P}(\boldsymbol{\theta})\}$ is the cardinality of the partition $\mathcal{P}(\boldsymbol{\theta})$ corresponding to a given value $\boldsymbol{\theta}$ in $\Gamma_{RDP}^{(N-1)p}$ or $\Gamma_{RP}^{(N-1)p}$. Because this partition is assumed common across lags ℓ and for all $v \in V \setminus \{u\}$, it may be thought of as a union, i.e., $\mathcal{P}(\boldsymbol{\theta}) = \bigcup_v \mathcal{P}(\boldsymbol{\theta}(u,v))$, where $\mathcal{P}(\boldsymbol{\theta}(u,v))$ is a partition corresponding specifically to the dynamic behavior of the coefficients $\theta_t^{(\ell)}(u,v)$ collectively over all lags ℓ. Thus the contribution of $\#\{\mathcal{P}(\boldsymbol{\theta})\}$ to the penalty may be thought of as counting the number of times there is a need to insert a changepoint due to a change in the relation of node u with any other node v at any lag ℓ. That is, it controls the number of partitions for the entire neighborhood.

The second part of the penalty in (2.6) is a sum, over intervals I in the relevant partition $\mathcal{P}$, of the ℓ_2 norms of the corresponding coefficient lag vectors. It is essentially a group lasso type penalty, in the spirit of that originally proposed by [82], with tuning parameter λ. The purpose of introducing this term is to encourage sparseness in the connectivity of each neighborhood, and hence of the network as a whole. Our use of the group lasso here derives from the definition

of our network G, where an edge is present regardless of in which lag there is a causal effect of a node v on the node u. The choice of tuning parameter controls the amount of shrinkage of the group of coefficients. Large λ results in sparser coefficient vectors. We describe a method for choosing the tuning parameter in Section 3.

2.2.3 Implementation

In this section, we discuss the implementation of our proposed methods of inference. For both the recursive dyadic partitioning estimator in (2.3) and the recursive partitioning estimator in (2.5), the general structure of the algorithm is similar. We describe the latter here and, for the sake of completeness, provide the former in the appendix.

Calculation of the estimator (2.5) can be accomplished as detailed in Algorithm 1. The required inputs are the time series $\mathbf{X}(u)$ for node u, the lagged time series $\mathbf{X}(-u)$ for all other nodes, and a prespecified number of lags p. Note that $p+1$ is the minimum number of observations necessary to fit a model of p lags. Initially we set the penalized likelihood to be the sum of squares of the data in the intervals I that contain less than the minimum required number of observations. There are $(T-1)!$ possible ways of partitioning (i.e., complete recursive partitions $\mathcal{P}^*$) in the library $\mathcal{L}$. Each partition, however, is composed only of subsets of $\binom{T+1}{2}$ unique intervals, given that each interval is defined between two endpoints. The algorithm begins by fitting group lasso penalized models on intervals I that contain more than $p+1$ observations. Therefore we have $\mathcal{O}(T^2)$ calls for fitting the group lasso type of models. (Because solving the group lasso regression generally requires iterative convex optimization, we do not quantify specifically the corresponding time complexity of this step.) We then consider intervals that contains $2(p+1)$ observations and compare the penalized likelihood pl_I in those intervals to

Data: $\mathbf{X}(u)$, $\mathbf{X}(-u)$, p
Result: $\hat{\boldsymbol{\theta}}_{RP}$
for *j = 1:p* **do**
  for *i = 1: T-j+1* **do**
    Compute and store pl_I on each interval I using:
    $pl_I = \sum_I (\mathbf{X}_I(u))^2$ for $I = \{t : t \in [i, i+j)\}$;
    optimumModel $\leftarrow pl_I$;
  end
end
for *j = p+1:T* **do**
  for *i = 1: T-j+1* **do**
    Fit restricted VAR(p) model for $\mathbf{X}_I(u), I = \{t : t \in [i, i+j)\}$;
    Compute and store pl_I on each interval I;
    if $pl_I \leq pl_{I_l^i} + pl_{I_r^i} + Penalty$ **then**
      optimumModel $\leftarrow pl_I$;
      Update changePoint;
    else
      optimumModel $\leftarrow pl_l$ and pl_r;
      Update changePoint;
    end
  end
end

Algorithm 1: Multiscale dynamic causal network inference using recursive partitioning.

the sum of the penalized likelihoods of the optimal sub intervals containing $p+1$ observations and retain the one with smaller value. The procedure is repeated for intervals containing k observations, with $k = 2(p+1)+1, \cdots, T$. There are $(k-1)$ ways of partitioning an interval of length k into two. Let $\{I_l^i, I_r^i\}_{i=1}^{k=1}$ be all possible pairs of subintervals of I such that $I_l^i \bigcup I_r^i = I$. We compare the penalized likelihood pl_I, defined in (2.5) but restricted to I, versus $\min_i\{pl_{I_l^i} + pl_{I_r^i} + \text{Penalty}\}$, and select the optimal model to be the one which has smallest value. The comparison is of order $\mathcal{O}(T^3)$ and thus the total computational cost is $\mathcal{O}(T^2)$ calls to group lasso type of fitting and $\mathcal{O}(T^3)$ comparisons.

2.3 Theoretical properties

In the previous section, we introduced our partition-based approach to modeling dynamical changes in the dependency relational structure among multiple time series, defined two estimators of the time-varying parameters underlying our models, and described an appropriate algorithm for calculations. In this section, we first show that the proposed approach can estimate a change point consistently. We then present an empirically-based choice of the penalty parameter λ in equation (2.6) and show that through this choice we can control the Type I error rate in recovering the true neighborhood structure of a node u within a given stationary time block. Finally, we quantify the overall risk behavior of our estimators.

2.3.1 Consistency of changepoint estimation

Suppose that there is a single change point at time τ, with $1 < \tau < T$. Then under our approach the time series $\mathbf{X}(u)$ can be written as a concatenation of two parts of length τ and $T - \tau$. We use L to denote the set of all observations in the pre-τ period and use R to denote the set of all observations in the post-τ period. Then we have:

$$X_t(u) = \begin{cases} \sum\limits_{v \in V \setminus \{u\}} \sum\limits_{\ell=1}^{p} X_{t-\ell}(v)\theta_L^{(\ell)}(u,v) + \epsilon_t(u), & t \in [1, \tau] \\ \sum\limits_{v \in V \setminus \{u\}} \sum\limits_{\ell=1}^{p} X_{t-\ell}(v)\theta_R^{(\ell)}(u,v) + \epsilon_t(u), & t \in (\tau, T] \end{cases} .$$

Our change point selection consistency result extends the result of [2], where the estimation consistency of the group lasso regression is established. The assumptions needed are the same as in that previous work, which we briefly restate here.

Assumption 1. *$X_t(u)$ and $\mathbf{X}_t(-u)$ have finite fourth order moments:* $\mathbb{E}(X_t(u))^4 <$

∞, *and* $\mathbb{E}\|\mathbf{X}_t(-u)\|^4 < \infty$.

Assumption 2. *Invertibility of the joint covariance matrix, defined as:* $\Sigma_{\mathbf{X}_t(-u)\mathbf{X}_t(-u)} := \mathbb{E}(\mathbf{X}_t(-u)'\mathbf{X}_t(-u)) - (\mathbb{E}\mathbf{X}_t(-u))'(\mathbb{E}\mathbf{X}_t(-u)) \in \mathbb{R}^{(N-1)p\times(N-1)p}$

Assumption 3. *We denote* $\hat{\boldsymbol{\theta}}_{\boldsymbol{t}}$ *any minimizer of* $\mathbb{E}\left(X_t(u) - \mathbf{X}_t(-u)\boldsymbol{\theta}_t\right)^2$. *We assume that* $\mathbb{E}\left(\left(X_t(u) - \mathbf{X}_t(-u)\hat{\boldsymbol{\theta}}_t\right)^2 |\mathbf{X}_t(-u)\right)$ *is almost surely greater than some* $\sigma^2_{\min} > 0$.

Assumption 4. $\max_{v\in S^c} \frac{1}{p}\left\|\Sigma_{\mathbf{X}(v)\mathbf{X}(S)}\Sigma^{-1}_{\mathbf{X}(S)\mathbf{X}(S)} Diag(1/\|\boldsymbol{\theta}_t(u,v)\|_2)\boldsymbol{\theta}_t(u,S)\right\|_2 < 1$, *where* S *is the set of nodes in the neighborhood of the* u *where* $(\|\boldsymbol{\theta}_t(u,v)\|_2 \neq 0)$ *and* $Diag(1/\|\boldsymbol{\theta}_t(u,v)\|_2)$ *denotes the block-diagonal matrix of size* $|S|p$ *in which each diagonal block equals to* $\frac{1}{\|\boldsymbol{\theta}_t(u,v)\|_2}\mathbf{I}_{|S|p}$ *with* $\mathbf{I}_{|S|p}$ *the identity matrix of size* $|S|p$. $\boldsymbol{\theta}_t(u,S)$ *denotes the concatenation of the coefficient vectors indexed by* S.

Note that when $p = 1$, Assumption 4 is referred to as the strong irrepresentable condition in [84].

Assumption 5. *The size of the network increases no faster than the square root of the length of the time series:* $\exists\,\gamma \geq 0$, *such that* $N = \mathcal{O}(T^\gamma)$ *as* $T \to \infty$ *for* $\gamma < 1/2$.

Consider the local test of

$$H_0 : \mathcal{P} = [1, T] \quad vs \quad H_1 : \mathcal{P} = [1, \tau] \cup (\tau, T],$$

using group lasso penalized least squares. This test corresponds to the basic step of comparing models for two adjacent intervals at the heart of Algorithm 1 (i.e., one model for the union versus a separate model for each interval), where the penalty is simply the second component of Pen_{RP} in (2.6). We have the following theorem:

Theorem 2.3.1. *Assume that Assumptions 1 to 5 are satisfied, where* λ *varies such that* $\lambda \to 0$, $\lambda N \to 0$ *and* $\lambda T^{1/2} \to \infty$, *as* $T \to \infty$. *Then we have that*

$$\mathbb{P}_{H_0}\left(Decide\ \mathcal{P} = [1, T]\right) \longrightarrow 1 \tag{2.7}$$

$$\mathbb{P}_{H_1}\left(|\hat{\tau} - \tau| > \epsilon\right) \longrightarrow 0, \quad \forall \epsilon > 0\ . \tag{2.8}$$

Theorem (2.3.1) contains two parts. The first part states that when the null hypothesis is true – that is, when the time series contains no change point – our method favors the model with no change point. The second part states that under the alternative hypothesis, where there is a change point at τ, our method favors the model with one estimated change point $\hat{\tau}$ and, furthermore, the probability that $\hat{\tau}$ differs from τ by an arbitrary amount ϵ tends to zero. The proof can be found in the appendix. The proof technique can be generalized for the case of multiple change points, although it would require appropriate conditions on the number of change points M and the number of data points T.

2.3.2 Finite sample control of Type I error rate in neighborhood selection

We see that consistent splitting and change point estimation is possible to achieve with the group lasso type of estimation. However, our asymptotic result offers little advice on how to choose a specific penalty parameter for a given problem. We propose a way to adaptively choose the penalty parameters λ, given a stationary time interval. For a specific λ, we guarantee that the probability of committing a certain notion of Type I error in recovering the connected component corresponding to the fixed node u is less than some user specified level α. The connected component $C_u \in G$ of a node $u \in V$ is defined as the set of nodes which are connected to node u by a chain of directed edges. We denote the neighborhood of node u as ne_u. The neighborhood ne_u is clearly part of the connected component C_u. To guarantee the accuracy of the neighborhood selection, we need the following additional assumption:

Assumption 6. *Denote by $\Theta = BV(C)$ the ball of functions of bounded variation for some constant C. We assume that is $\theta^{(\ell)}_{(\cdot)}(u,v) \in \Theta$, for all $\ell = 1, \cdots, p$ and*

all $v \in V \backslash \{u\}$*:*

$$\sup_{J \geq 2} \sup_{t_1 \leq \cdots \leq t_J} \sum_{j=p}^{J} \left| \theta_{t_j}^{(\ell)}(u, \cdot) - \theta_{t_{j-1}}^{(\ell)}(u, \cdot) \right| < C$$

This assumption indicates that $\|\boldsymbol{\theta}_t(u, v)\|_2$ *is bounded.*

In the case where $X(u)$ is stationary on a given interval $[1, T]$, we have the following theorem regarding the estimated connected component $\hat{C}_u$:

Theorem 2.3.2. *Assume Assumptions 1 to 6 hold, and fix* $\alpha \in (0, 1)$*. If* $\mathbf{X}(u)$ *is stationary on* $[1, T]$ *and the penalty parameter* $\lambda(\alpha)$ *is chosen such that*

$$\lambda(\alpha) = 2\hat{\sigma}(u) \sqrt{pQ\left(1 - \frac{\alpha}{N(N-1)}\right)},$$

where $\hat{\sigma}^2(u) = \|\mathbf{X}(u)\|_2^2 / T$ *and* $Q(\cdot)$ *is the quantile function of* $\chi^2(p)$ *distribution, then*

$$\mathbb{P}\left(\exists u \in V : \hat{C}_u \nsubseteq C_u\right) \leq \alpha \ .$$

Theorem (2.3.2) says that by choosing the penalty parameter at $\lambda = \lambda(\alpha)$, the probability of falsely joining two distinct connected components with the estimate of the edge set is bounded above by the level of α, which is a more general result and includes the case when the connected components happened to be the local neighborhood of node u. The proof of the theorem is provided in the appendix.

2.3.3 Risk analysis

We now provide a theorem that gives an upper bound on the risk of the estimators $\hat{\boldsymbol{\theta}}_{RDP}$ and $\hat{\boldsymbol{\theta}}_{RP}$. Through this approach we provide a certain measure of quality for the overall dynamic network inference procedure. Following the perspective of [55], as implemented in [48], we measure the loss of estimating $\boldsymbol{\theta}$ by $\hat{\boldsymbol{\theta}}$ in

terms of the squared Hellinger distance between the two corresponding conditional densities:

$$\begin{aligned} L(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) &\equiv H^2(p_{\hat{\boldsymbol{\theta}}}, p_{\boldsymbol{\theta}}) \\ &= \int \left[\sqrt{p_{\hat{\boldsymbol{\theta}}}(\mathbf{x}|\mathbf{X}(-u))} - \sqrt{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{X}(-u))} \right]^2 d\nu(\mathbf{x}) \end{aligned}$$

with respect to some dominating measure $\nu(\mathbf{x})$. Additionally, define the Kullback-Leibler divergence between two densities of $\mathbf{X}(u)$, conditional on the past of all the neighborhood time series:

$$K(p_{\boldsymbol{\theta}^1}, p_{\boldsymbol{\theta}^2}) \equiv \int \log \frac{p(\mathbf{x}|\mathbf{X}(-u), \boldsymbol{\theta}^1)}{p(\mathbf{x}|\mathbf{X}(-u), \boldsymbol{\theta}^2)} p(\mathbf{x}|\mathbf{X}(-u), \boldsymbol{\theta}^1) d\nu(\mathbf{x}).$$

Theorem 2.3.3. *Denote the loss function of estimating $\boldsymbol{\theta}$ by $\hat{\boldsymbol{\theta}}$ by $L(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})$ and the corresponding risk, by $R(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) = T^{-1}\mathbb{E}_{\mathbf{X}(u)|\mathbf{X}(-u)} \left[L(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) \right]$. Let $\Lambda = \alpha_{max}/T$, where α_{max} is the largest eigenvalue of $\mathbf{X}(-u)'\mathbf{X}(-u)$. Assume each $\theta_t^{(\ell)}(u, v)$ is of bounded variation on $(0, 1]$ for some constant C. Then for any λ of the same order as in Theorem 2.3.1 and for $T > \lceil e^{2p/3} \rceil$, our risk is bounded as*

$$R(\hat{\boldsymbol{\theta}}_{RDP}, \boldsymbol{\theta}) \leq \mathcal{O}\left(\left(\frac{\Lambda \log^4 T}{T} \right)^{1/3} \right)$$

for recursive dyadic partitioning and

$$R(\hat{\boldsymbol{\theta}}_{RP}, \boldsymbol{\theta}) \leq \mathcal{O}\left(\left(\frac{\Lambda \log^2 T}{T} \right)^{1/3} \right)$$

for recursive partitioning.

Theorem 2.3.3 shows that both estimators have risks that end to zero at rates slightly worse than $T^{-1/3}$. The asymptotic risk for recursive partitioning is smaller than the risk for recursive dyadic partitioning, albeit at the cost of increased computational complexity. Proof of this result is in line with the work by [48] and can be found in the appendix.

2.4 Simulation study

In this section, we illustrate the practical performance of our method through a series of simulation studies. In the first part, we simulate multivariate time series data under different settings, as dictated by models A - C below. In the second part, we scale up model B by increasing the size of the vertex set V and include more irrelevant variables. Under each model, we simulate 100 datasets and the white noise is always set to be $\epsilon_t(\cdot) \sim N(0,1)$. In all models, we set $\alpha = 0.05$ and $p = 2$. These choices match that of the computational neuro-science example we present later, in Section 5. We measure performance in three ways: (i) how many change points were detected, (ii) Out of the detected change points, how many specify the right location (iii) whether the correct neighborhood structure was detected. The models we investigate are:

- Model A: VAR(2) process with no change point.

  This scenario is designed to see the performance of the methods when there is no change point and the process is stationary. Specifically,

  $$X_t(1) = 0.5X_{t-1}(2) + 0.25X_{t-2}(2) + 0.5X_{t-1}(3) + 0.25X_{t-2}(3) + \epsilon_t(1)$$

  with sample size $T = 1024$.

- Model B: piecewise stationary VAR(2) process with 2 change points.

  Specifically,

  $$X_t(1) = \begin{cases} 0.5X_{t-1}(2) + 0.25X_{t-2}(2) + \epsilon_t(1) & 0 < t \leq 512 \\ 0.5X_{t-1}(3) + 0.25X_{t-2}(3) + \epsilon_t(1) & 512 < t \leq 768 \\ 0.5X_{t-1}(2) - 0.5X_{t-1}(3) + \epsilon_t(1) & 768 < t \leq 1024 \end{cases}$$

- Model C: change point close to the boundary.

Model	RDP			RP		
	Model A	Model B	Model C	Model A	Model B	Model C
# change point						
0	100	0	100	100	0	89
1	0	28	0	0	0	11
2	0	72	0	0	100	0
# exact detection						
0	100	0	100	100	0	89
1	0	28	0	0	11	11
2	0	72	0	0	89	0
# false edge detection						
0	100	97	100	100	94	100
1	0	3	0	0	6	0
2	0	0	0	0	0	0

Table 2.1: Simulation results under Model A, Model B and Model C, using RDP and RP.

Specifically,

$$X_t(1) = \begin{cases} 0.5X_{t-1}(2) + 0.25X_{t-2}(2) + \epsilon_t(1) & 0 < t \leq 128 \\ 0.5X_{t-1}(3) + 0.25X_{t-2}(3) + \epsilon_t(1) & 128 < t \leq 1024 \end{cases}$$

- Model B with VAR(2) process in a larger vertex set V.

  We use the same coefficients as used in Model B, but with the size of the vertex set ranging from 5 to 15.

The results for models A, B, and C are summarized in Table 2.1. For some error measures, results under the truth are marked in blue. For example, under model A where there is no change point in the true model, positions corresponding to 0 change point and 0 exact detection are marked in blue, i.e., one should not detect anything where there is no change point. Under model B, where there are two change points, results corresponding to the case of two change points and two exact detections are marked in blue. Note that in the case recursive partitioning (i.e., non-dyadic), we treat a detection as being 'exact' if an estimated change point is within ± 5 time points of the true change point (i.e., less than 0.5% the length of the full time series).

A few comments on these results are in order:

	RDP					RP				
Size of V	5	7	11	13	15	5	7	11	13	15
# change point										
0	22	48	67	95	100	0	19	52	93	100
1	20	20	29	5	0	0	4	2	0	0
2	58	32	4	0	0	100	77	46	7	0
# exact detection	0									
0	22	48	67	95	100	0	19	52	93	100
1	20	20	29	5	0	17	19	4	0	0
2	58	32	4	0	0	83	62	44	7	0
# false edge detection										
0	98	100	100	100	100	98	97	100	100	100
1	2	0	0	0	0	2	3	0	0	0
2	0	0	0	0	0	0	0	0	0	0

Table 2.2: Simulation results under Model B for vertex sets of increasing cardinality.

- From the results we see that our proposed estimators did not overestimate the number of change points, as they never detected more change points than the true number of change points.
- Under Model B, in 72 out of 100 and in 89 out of 100 trials we correctly specified the number of positions of the change points using the recursive dyadic partition and the recursive partition estimators, respectively. Note that if we are less conservative and allow more tolerance in defining an 'exact detection' under recursive partitioning, all change points identified in Model B using recursive partitioning are located within $[-13, 13]$ points of the true change points (i.e., within 1.5% of the total length of the full time series).
- Based on the results under model C, we conclude that our methods lose sensitivity to detection of change points as the location of the change points moves closer to the boundary, with recursive partitioning performing better than recursive dyadic partitioning. These results are to be expected.
- We have good control over the false detection of causal structures.

The performance of the proposed estimators upon increasing the size N of the vertex set V, under model B, is summarized in Table 2.2. As N increases, we see the performance decreases, due to the fact that in this setting the variables we are adding are irrelevant and thus induce additional uncertainty. Note that under our

proposed approach there is a tendency to underfit the number of change points rather than over fit. This trait will be relevant to the real data application we describe next.

2.5 Illustration: Inference of a task-based MEG network

Neuroscientists are interested in understanding the interactions among cortical areas that allow subjects to detect the motion of objects. In [15], fMRI was used to study subjects who were asked to perform visual search tasks and it was found that the monitored regions of interest (ROIs) formed four clusters. However, fMRI does not have good temporal resolution for more detailed investigation of the interaction between these clusters. [69] studied the 10 Hz Alpha-band power extracted from MEG signals under a similar multiple-trial visual motion search experiment. They found evidence showing that regions of interest within the identified clusters have similar temporal activation profiles. Specifically, they found significant inhibition of 10Hz alpha power in the visual processing region after 300ms relative to the stimulus, and longer and sustained alpha power in the frontoparietal region. Other evidence of co-activations among regions of interest have been reported by other studies under different experimental set up. For example, see [11], [1] and [8].

To demonstrate the application of our method, we examined the same 10 Hz Alpha-band power data used by [69]. MEG data has excellent temporal resolution, but the spatial resolution is less good than that of fMRI. As a result, it is typical that functional connectivity analyses with MEG data incorporate coarsely defined brain regions and hence networks with only a handful of vertices. We therefore chose three regions of interest each from the two clusters known to have similar activation profiles. The regions of interest are V3a, MT+ and VIP from the visual

processing region, and FEF, SPL and DLPFC from the frontoparietal region. This choice corresponds to a network of six nodes, which is consistent with studies of this type.

Details of the experiment and the data are as follows. In the experiment, a participant was asked to perform a visual search task of a moving object, repeated over 160 trials. Each trial began with a 300 ms blank screen. Then, 9 spheres fade in over a 1000 ms period and these 9 spheres remained static for another 1000 ms. A 1000 ms motion display period then follows, where 8 of the spheres move forward (simulating forward motion of the obeserver) and the target sphere moves independently from the others. The beginning of the motion display period defined the 0 ms marker for each trial. Finally, in the 3000 ms response period, the 9 spheres remained static, four (including the target) were grayed out, and the participant was asked to identify the target sphere.

The MEG signal of the participant was recorded throughout the experiment. The data we used is the 10 Hz Alpha-band power, truncated in a uniform manner across trials, to focus upon the period just prior to the appearance and movement of the spheres. It starts from the second half of the static period and the length of the data is $T = 1502$, corresponding to a time interval of length 2500 ms. The time series we used for our analyses contains the last 500 ms of the static period, the entire motion display period, and the first 1000 ms of the response period, where most of the correct responses occurred. The timeline of our data is illustrated by Fig 2·2. For a more detailed description of the experiment, please refer to [69].

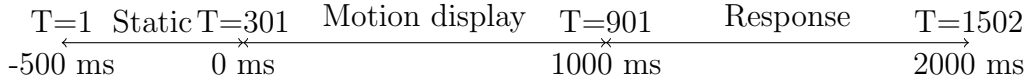
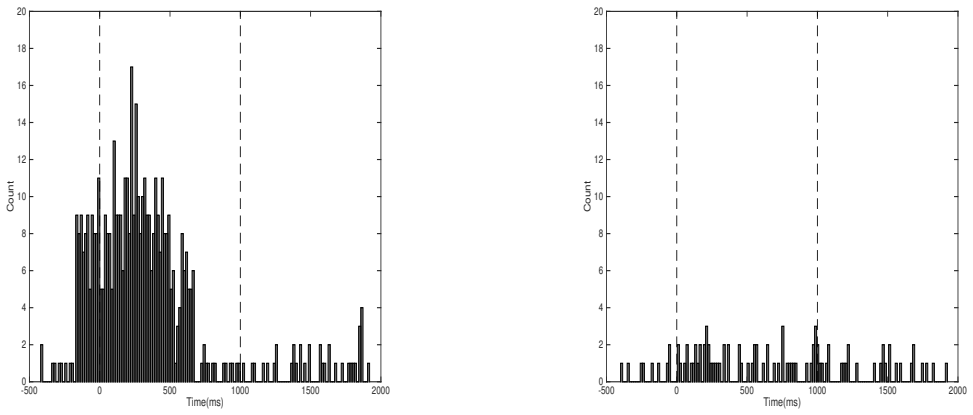


Figure 2·2: Visual search experiment time line.

Each time series has been pre-processed by taking the first order difference to remove the self-driven component. We then use the recursive partition based method with lag $p = 7$ (chosen in preliminary analysis using the Akaike information criterion). We set the level α in Theorem 2.3.2 to 0.05. The recursive dyadic method does not apply here because the length of the data is not a power of 2.



(a) Distribution of change points among the visual processing region.

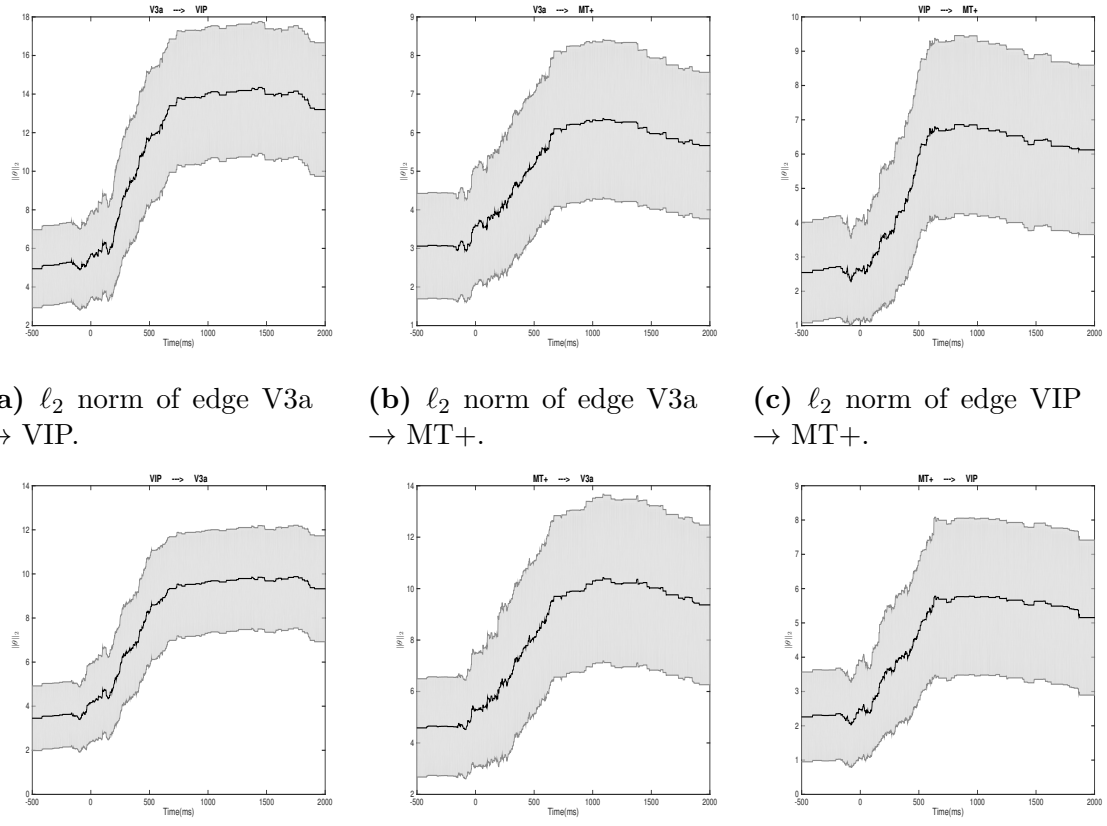
(b) Distribution of change points among the frontoparietal region.

Figure 2·3: The change point distribution among the visual processing region and the frontoparietal region.

Fig 2·3 shows the distribution of the detected change points among each of the two clusters we examined. The two dashed vertical lines indicate the time of the two phase changes. There are 497 change points detected across the 160 trials in the visual search region, of which 427 lie between -150 ms and 750 ms, relative to the stimulus onset. Compared with the visual processing regions, there are much fewer change points detected among the frontoparietal regions, where the Alpha-band power is more sustained.

Strength of the connections between regions of interest, within each of the two clusters, is shown in Figures 2·4 and 2·5, where we have plotted the pointwise

means and one standard deviation error bars of the ℓ_2 norm of the coefficients across the 160 trials. The inhibitive role of the Alpha-band power in the visual processing region (i..e, the creation of a common co-deactivation pattern), in response to the stimulus, is understood to be the reason for the significant increase in the ℓ_2 norms of the coefficients among V3a, MT+ and VIP from -150 ms to 750 ms. And, in fact, most of the changepoints in this time interval among these three regions of interest correspond to an increase in the ℓ_2 norm of the pair-wise regression coefficients. In contrast, the changes of the ℓ_2 norms of the coefficients in the frontoparietal region are much more gradual.



(a) ℓ_2 norm of edge V3a $\rightarrow$ VIP.

(b) ℓ_2 norm of edge V3a $\rightarrow$ MT+.

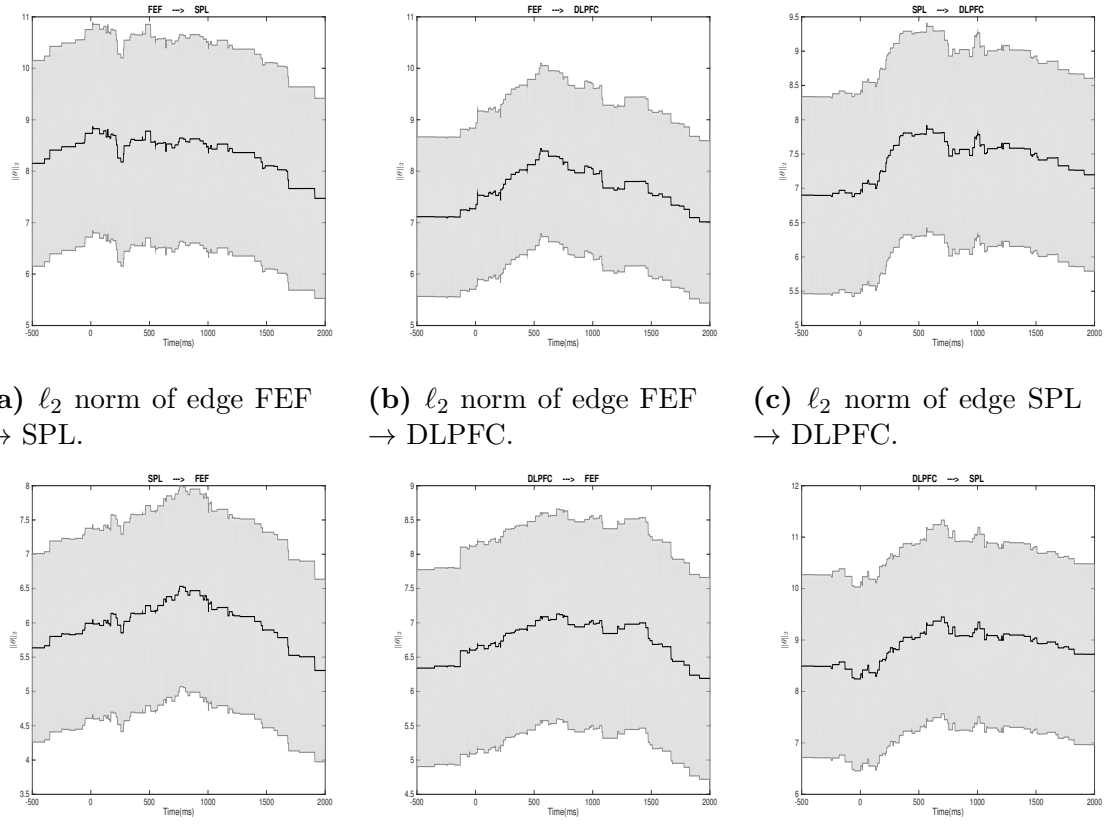
(c) ℓ_2 norm of edge VIP $\rightarrow$ MT+.

(d) ℓ_2 norm of edge VIP $\rightarrow$ V3a.

(e) ℓ_2 norm of edge MT+ $\rightarrow$ V3a.

(f) ℓ_2 norm of edge MT+ $\rightarrow$ VIP.

Figure 2·4: ℓ_2 norms of coefficients between pairs of time series in the visual processing region.



(a) ℓ_2 norm of edge FEF $\rightarrow$ SPL.

(b) ℓ_2 norm of edge FEF $\rightarrow$ DLPFC.

(c) ℓ_2 norm of edge SPL $\rightarrow$ DLPFC.

(d) ℓ_2 norm of edge SPL $\rightarrow$ FEF.

(e) ℓ_2 norm of edge DLPFC $\rightarrow$ FEF.

(f) ℓ_2 norm of edge DLPFC $\rightarrow$ SPL.

Figure 2·5: ℓ_2 norms of coefficients between pairs of time series in the frontoparietal region.

As an aside, we note that comparatively few interactions were found between the visual processing region and the frontoparietal region using our method (results not shown).

2.6 Discussion

Motivated by the types of questions arising in task-based neuroscience – particularly using imaging modalities with fine-scale temporal resolution – we proposed

a novel method for simulataneous network inference and change point detection. Various extensions are possible. For example, a penalty in the spirit of the fused-lasso would be of interest here, to encourage a certain notion of temporal contiguity. In addition, a speed-up of the implementation (particularly for the non-dyadic case) would be desirable – and, indeed, necessary for larger networks than those studied here – adopting, for example, ideas like those underlying the PELT algorithm presented by [44]. Finally, it would be natural to explore the utility of our proposed method in the context of financial economics.

Chapter 3

Multiscale network analysis through tail-greedy bottom-up approximation, with applications in neuroscience

3.1 Introduction

Wavelet-related methods have been developed extensively for classical signal and image processing problems. In recent years, there has been a resurgence of interest in this area, within the emerging field of graph signal processing. Advancement in this area does not come without challenges. Not all tools in classical signal processing can be directly transfered to graph signal processing. Problems with which people are concerned include but are not limited to: How to effectively compress a signal on a network? How to identify and remove noise from a signal on a graph? See [70] for seminal work in graph signal processing. An early characterization of the multi-scale aspect of this field can be found in [73].

In this work, we present an algorithm that offers certain solutions to these problems. We work with an undirected connected graph $G^0 = (V^0, E^0)$, where $V^0 = \{v_1^0, \cdots, v_N^0\}$ is the set of vertices and $E^0 = \{e_{ij}^0 | v_i^0 \in V^0, \quad v_j^0 \in V^0, \quad i \neq j\}$ is the set of edges (assumed to be without self-loops).

Our contribution is a new class of graph wavelets, based on the notion of "tail-greedy" unbalanced Haar transformations. We also establish a consistency result for piecewise-constant function estimation using thresholding procedures in

the corresponding basis obtained from our TGUH.

The tail-greedy unbalanced Haar transformation was originally proposed by Fryzlewicz [31] for the one-dimensional signal plus noise model. The algorithm results in a nonlinear but conditionally orthonormal, multiscale decomposition of the data with respect to an adaptively chosen unbalanced Haar wavelet basis. Related work also includes the work by Fryzlewicz and Timmermans [32], where an adaptive Haar-type of transformation is used for image compression and denoising. Here we extend the TGUH method to networks and use it to show various results and applications.

The organization of this paper is as follows. In section II, we present the development of our TGUH transforms. In section III, we discuss graph signal denoising with the TGUH and propose a consistent method of estimation. In section IV, we illustrate the utility of our algorithm through simulation and application in computational neuroscience. In section V, we discuss potential extensions.

3.2 TGUH of networks

Consider a graph $G^0 = (V^0, E^0)$, the connectivity of which we summarize by its adjacency matrix $\mathbf{W}^0 \in \mathbb{R}^{n\times n}$, where $w^0_{ij} \geq 0$ is the (possibly non-binary) edge weight between vertex i and j, such that $w^0_{ij} = 0$ indicates no edge.

The TGUH is a bottom-up method. At each run, we select columns from the adjacency matrix, corresponding to linked pairs of nodes, and merge them by applying an orthonormal transformation to the columns. We define $\mathbf{W}^\ell$ to be the adjacency matrix, and V^ℓ to be the corresponding set of vertices, after the ℓ-th iteration. We denote by v^ℓ_r the r-th node in V^ℓ, and by N^ℓ_r and $N^\ell_{r'}$, the number of nodes in V^0 represented (through merging) in meta nodes $v^{\ell-1}_r$ and $v^{\ell-1}_{r'}$. Let $\rho \in (0, 1)$ be a constant, used to describe the proportion of pairs of

nodes to merge at each run. More precisely, we merge $\lceil \rho |V^{\ell-1}| \rceil$ pairs of nodes at the ℓ-th iteration, where the expression $|\cdot|$ is the cardinality of the vertex set. The parameter ρ controls the speed and greediness of our method. When $\rho = 1/2$, the transform reduces to the non-adaptive (and therefore non-greedy) classical Haar transform; the degree of greediness increases as ρ decreases. In our applications, we use $\rho = 0.01$.

We now outline the procedure.

1. At the ℓ-th iteration, search for $\lceil \rho |V^{\ell-1}| \rceil$ pairs of columns for which the ℓ_2 norms of the detail coefficient vectors are the smallest. To be more precise, the algorithm proceeds as follows:

   For each pair of columns corresponding to pairs of connected nodes $(v_r^{\ell-1}, v_{r'}^{\ell-1})$, construct a "detail" filter $(a_{(r,r')}^{\ell}, -b_{(r,r')}^{\ell})$, where each filter is uniquely indexed by ℓ and the pair (r, r'). Here $a_{(r,r')}^{\ell} > 0$ and $b_{(r,r')}^{\ell} > 0$ are set in the following way:

   (a) To produce a sparse representation of the initial input matrix, the algorithm needs to produce zero details over regions of constancy of the network, by which we mean nodes sharing identical neighborhood structure and weighting. Let $j = j(r)$ and $j' = j(r')$ denote the corresponding positions in the adjacency matrix and assume $j < j'$. We compute the details using

   $$\mathbf{d}_{(r,r')}^{\ell} = a_{(r,r')}^{\ell} \mathbf{W}_{\cdot j}^{\ell-1} - b_{(r,r')}^{\ell} \mathbf{W}_{\cdot j'}^{\ell-1} \ .$$

   (b) To force orthonormality of the transformation, we impose the following requirement:

   $$a_{(r,r')}^{\ell}{}^2 + b_{(r,r')}^{\ell}{}^2 = 1$$

The two requirements above determine a unique filter. The detail coefficient vector is computed as

$$\mathbf{d}^{\ell}_{(r,r')} = \left[\mathbf{W}^{\ell-1}_{\cdot j}, \mathbf{W}^{\ell-1}_{\cdot j'}\right]_{N\times 2} \begin{bmatrix} a^{\ell}_{(r,r')} \\ -b^{\ell}_{(r,r')} \end{bmatrix}_{2\times 1} .$$

2. Sort the norms of the detail coefficient vectors $\|\mathbf{d}^{\ell}_{(r,r')}\|_{\ell_2}$ in ascending order and extract $\lceil \rho |V^{\ell-1}| \rceil$ detail coefficient vectors corresponding to the smallest $\lceil \rho |V^{\ell-1}| \rceil$ elements in the sorted sequence. If any element of the sorted sequence uses nodes already used by any of the previous elements, it is discarded and the next candidate is considered. In the case where there are fewer than $\lceil \rho |V^{\ell-1}| \rceil$ detail coefficient vectors, extract all of them. No nodes will be merged more than once at iteration ℓ.

3. For each $\|\mathbf{d}^{\ell}_{(r,r')}\|_{\ell_2}$, use filter $(b^{\ell}_{(r,r')}, a^{\ell}_{(r,r')})$, which is orthogonal to the previous filter used for computing the detail coefficient, to produce the corresponding merged columns:

$$\mathbf{W}^{\ell}_{\cdot j} \leftarrow \left[\mathbf{W}^{\ell-1}_{\cdot j}, \mathbf{W}^{\ell-1}_{\cdot j'}\right]_{N\times 2} \begin{bmatrix} b^{\ell}_{(r,r')} \\ a^{\ell}_{(r,r')} \end{bmatrix}_{2\times 1}$$

$$\mathbf{W}^{\ell}_{\cdot j'} \leftarrow \mathbf{d}^{\ell}_{(r,r')}$$

where $\leftarrow$ indicates replacement of the original rows/columns with the new one. Alternatively, these operations can be written as

$$\left[\mathbf{w}^{\ell}_{\cdot j}, \mathbf{w}^{\ell}_{\cdot j'}\right]_{N\times 2} = \left[\mathbf{w}^{\ell-1}_{\cdot j}, \mathbf{w}^{\ell-1}_{\cdot j'}\right]_{N\times 2} \begin{bmatrix} b^{\ell}_{(r,r')} & a^{\ell}_{(r,r')} \\ a^{\ell}_{(r,r')} & -b^{\ell}_{(r,r')} \end{bmatrix}_{2\times 2} .$$

The transformation matrix is a rotation matrix.

4. Perform the corresponding row-wise rotation and symmetrize $\mathbf{W}^{\ell}$.

$$\begin{bmatrix} \mathbf{w}^{\ell}_{j\cdot} \\ \mathbf{w}^{\ell}_{j'\cdot} \end{bmatrix}_{2\times N} = \begin{bmatrix} b^{\ell}_{(r,r')} & a^{\ell}_{(r,r')} \\ a^{\ell}_{(r,r')} & -b^{\ell}_{(r,r')} \end{bmatrix}_{2\times 2} \begin{bmatrix} \mathbf{w}^{\ell-1}_{j\cdot} \\ \mathbf{w}^{\ell-1}_{j'\cdot} \end{bmatrix}_{2\times N}$$

5. Set $\ell \leftarrow \ell + 1$ and go back to step 1, unless the transform is completed.

A compact summary of the above description is provided as Algorithm 2. Code implementing this algorithm is available at

`github.com/KolaczykResearch/NetworkTGUH-Code/`.

Input: Adjacency matrix: $\mathbf{W}^0$

1: **for** level $\ell = 1$ to L **do**
2: **for** each pair of connected nodes (r, r') **do**
3: **Compute** candidate "detail" coefficient vector norms $\|\mathbf{d}^{\ell}_{(r,r')}\|_2 = \|a^{\ell}_{(r,r')}\mathbf{W}^{\ell-1}_{\cdot j} - b^{\ell}_{(r,r')}\mathbf{W}^{\ell-1}_{\cdot j'}\|_2$.
4: **Store** $\|\mathbf{d}^{\ell}_{(r,r')}\|_2$.
5: **end for**
6: **Sort** $\|\mathbf{d}^{\ell}_{(r,r')}\|_2$ in ascending order.
7: **for** $i = 1$ to $\lceil \rho |V^{\ell}| \rceil$ **do**
8: **Select** the column/row corresponding to the smallest $\|\mathbf{d}^{\ell}_{(r,r')}\|_2$.
9: **Update** $W^{\ell-1}$ by

$$\left[\mathbf{W}^{\ell}_{\cdot j}, \mathbf{W}^{\ell}_{\cdot j'}\right] \leftarrow \left[\mathbf{W}^{\ell-1}_{\cdot j}, \mathbf{W}^{\ell-1}_{\cdot j'}\right] \begin{bmatrix} b^{\ell}_{(r,r')} & a^{\ell}_{(r,r')} \\ a^{\ell}_{(r,r')} & -b^{\ell}_{(r,r')} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{W}^{\ell}_{j\cdot} \\ \mathbf{W}^{\ell}_{j'\cdot} \end{bmatrix} \leftarrow \begin{bmatrix} b^{\ell}_{(r,r')} & a^{\ell}_{(r,r')} \\ a^{\ell}_{(r,r')} & -b^{\ell}_{(r,r')} \end{bmatrix} \begin{bmatrix} \mathbf{W}^{\ell-1}_{j\cdot} \\ \mathbf{W}^{\ell-1}_{j'\cdot} \end{bmatrix}.$$

10: **end for**
11: **end for**

Algorithm 2: TGUH transform of network

We make a few comments regarding the TGUH. The resulting transformation is non-linear, but it is orthonormal conditional on the order in which the detail coefficient vectors are selected. Given that the transform is conditionally orthonormal, it preserves the ℓ_2 energy of the adjacency matrix. Because small detail coefficients are selected at the beginning of the algorithm, most energy will be concentrated at coarser scales.

The algorithm can be viewed as a variation on agglomerative hierarchical clustering for community detection (e.g., [46, Ch 4.3.3.1]), using a column-wise ℓ_2 norm as our measure of so-called 'linkage', but with particular attention paid to

the notions of coarsening and detail, in the usual multiscale tradition. From this perspective, our TGUH is similar in spirit to the hierarchical clustering algorithm of Singh, Nowak, and Calderbank [75].

The term "tail-greedy" comes from the fact that in each run, we select from among the lower tail of the distribution of "details". "Tail-greedy" is not as greedy as standard greedy methods, as it may select more than one detail per run, which ensures the method terminates in at most $\mathcal{O}(\log N)$ levels.

The complexity of the TGUH transform is nearly linear in the number of edges in the graph. For a graph of size N, the number of nodes remaining after ℓ iterations is at most $(1-\rho)^\ell N$. Solving for the smallest ℓ such that $(1-\rho)^\ell N < 1$ yields that $\ell > \frac{\log N}{\log(1-\rho)^{-1}}$. At each step, we need to compute details for every edge and sort, which requires up to $\mathcal{O}(|E| \log |E|)$ operations. Accordingly, the complexity of the overall TGUH scales like $\mathcal{O}(\frac{\log N}{\log(1-\rho)^{-1}} \times |E| \log |E|)$. Note that for sparse graphs, where the number of edges is of the same order as the number of vertices, $|E| \sim N$, the complexity scales like $\mathcal{O}(\frac{N \log^2 N}{\log(1-\rho)^{-1}})$.

Note that the TGUH can be expressed in a series of matrix multiplications:

$$\mathbf{W}^L = F^L \cdots F^1 \mathbf{W}^0 {F^1}^\top \cdots {F^L}^\top \quad .$$

Using our tail-greedy algorithm, the collection of the column spaces of the F's corresponds to that of an unbalanced Haar type of basis and the adjacency matrix is effectively decomposed in a bottom-up fashion. From this perspective, our TGUH transform is a special case of the multi-resolution matrix factorization (MMF) method of Kondor, Teneva and Garg [49], which compresses matrices efficiently through the use of a sequence of sparse orthogonal transforms. Specifically, our TGUH constitutes a 2nd order Jacobi MMF in the language of that

paper. We note, however, that the problem of denoising a graph signal is not considered in [49].

3.3 Denoising graph signals using TGUH

We have introduced a Haar-like basis for a network G^0. Now consider a signal f on that network. If the signal varies in a way that is somehow 'consistent with' the network structure that drives the network-adaptive steps of our TGUH algorithm, then we should have good signal compression when transforming f through the resulting TGUH orthobasis. Estimation techniques that can exploit this compression should then prove effective for denoising a signal observed with noise. In this section we explore the use of TGUH bases for denoising graph signals.

We adopt the standard signal plus noise model, $g(v) = f(v) + \epsilon(v)$, for $v \in V^0$. Here $g(v)$ is the observed signal, $f(v)$ is an unknown true signal, and $\epsilon(v)$ is an independent and identically distributed $N(0,1)$ noise. We assume that f is 'piece-wise constant' in the sense that the number of edges $e^0_{ij} \in E^0$ for which $f(i) \neq f(j)$ is no more than some constant $K > 0$. See [17], for example, for related notions of 'piece-wise smooth' functions based on vertex subsets.

Traditional multiresolution analysis constructs a sequence of function spaces $\{U_\ell\}$ of increasingly finer scale, by recursively dividing each U_ℓ into a coarser part $U_{\ell+1}$ and its orthogonal complement $W_{\ell+1}$. The latter are the wavelet subspaces. The original space U_0 can thus be decomposed as

$$U_0 = \bigoplus_{\ell=1}^{L} W_\ell \bigoplus U_L \ .$$

Decompositions of a function $f \in U_0$ follow accordingly.

The TGUH wavelet transform up to level L can be expressed as

$$f(v) = \sum_{\ell=1}^{L} \sum_{r=1}^{k(\ell)} \mu_r^\ell \psi_r^\ell(v) + \sum_{r=1}^{k(L)} \gamma_r \phi_r^L(v),$$

where $\mu_r^\ell = \langle f, \psi_r^\ell \rangle$ are the wavelet coefficients with respective to the Haar basis functions ψ_r^ℓ with support on the nodes set V_r^ℓ. We estimate f by estimating each μ_r^ℓ and then taking the inverse transformation.

Define empirical coefficients

$$\alpha_r^\ell = \sum_v g(v)\psi_r^\ell(v) \ .$$

Suppose that the two sets $V_m^{\ell'-1}$ and $V_{m'}^{\ell'-1}$ contain the nodes that merge into the meta node $v_{r'}^{\ell'}$ at the next level, that is $V_{r'}^{\ell'} = V_m^{\ell'-1} \cup V_{m'}^{\ell'-1}$. We define the estimator

$$\hat{\mu}_r^\ell = \alpha_r^\ell \mathbb{I}\left\{ \exists V_{r'}^{\ell'} \subseteq V_r^\ell \,\middle|\, |\alpha_{r'}^{\ell'}| > \lambda(\ell', r') \right\} \ , \tag{3.1}$$

where

$$\lambda(\ell', r') = \sqrt{2 \log N} \left\{ \frac{\sqrt{|V_m^{\ell'-1}|} + \sqrt{|V_{m'}^{\ell'-1}|}}{\sqrt{|V_m^{\ell'-1}| + |V_{m'}^{\ell'-1}|}} \right\} \ . \tag{3.2}$$

In this case, we estimate μ_r^ℓ by zero if α_r^ℓ and all of its children coefficients fall below the threshold. The advantage of using this type of threshold is that it allows us to construct an unbiased estimator $\hat{f}$ of f, in the sense that within each constant regime, our estimator is the sample mean of the observed signal within that constant section.

For any estimator $\tilde{f}$ of f, we denote the squared empirical L_2 risk as $R = \frac{1}{N}\sum_v (\tilde{f}(v) - f(v))^2$. We then have the following result characterizing the performance of our estimator $\hat{f}$.

Theorem 3.3.1. *Let $\hat{f}$ be an estimator of f obtained through the inverse TGUH transform of the estimated coefficients $\hat{\mu}_r^\ell$ in (3.1), based on the thresholding func-*

tion $\lambda(\ell', r')$ *in (3.2). Suppose* $K = o\left(N/\log^2 N\right)$. *Then we have that the risk* $R(\hat{f}, f)$ *is of order* $\mathcal{O}\left(\frac{K \log^2 N}{N \log(1-\rho)^{-1}}\right)$ *on the set*

$$\mathcal{A} = \left\{ \forall r, \ell, \quad |V_r^\ell|^{-1/2} \left| \sum_{v \in V_r^\ell} \epsilon(v) \right| \leq \sqrt{2 \log N} \right\} \ ,$$

where $\mathbb{P}(\mathcal{A}) \to 1$ *as* $N \to \infty$.

Theorem 3.3.1 says that the estimator $\hat{f}$ is an L_2-consistent estimator of the signal f. The key driver behind the result is the property of "tail-greediness", by which multiple pairs of nodes are merged at each level ℓ. L_2 consistency cannot be guaranteed if the algorithm is greedy, where only one pair of nodes from the tail distribution is merged. Proof of Theorem 3.3.1 can be found in the appendix.

We note that the assumption of piecewise constant f is presumably stronger than needed here, and can likely be relaxed to the case of functions of a certain Hölder smoothness. For example, the type of necessary intermediate approximation theoretic results for Hölder smooth functions follows for our TGUH basis immediately from [20].

3.4 Applications

3.4.1 Simulations

We use simulation to establish a simple proof of concept regarding compression by TGUH. Specifically, we look at the use of our TGUH transform to compress a 'barbell' type of network, i.e., a network with two fully connected components joined by a single link. We generated such a barbell with 10 nodes in each fully connected component and applied the TGUH transform described in Algorithm 2. All detail coefficients are zero except that resulting from the last

step of the algorithm, where the two connected components merge together, which demonstrates that the TGUH transform is able to capture well the structure of the barbell network. The resulting compression curve is shown in Figure 3·1.

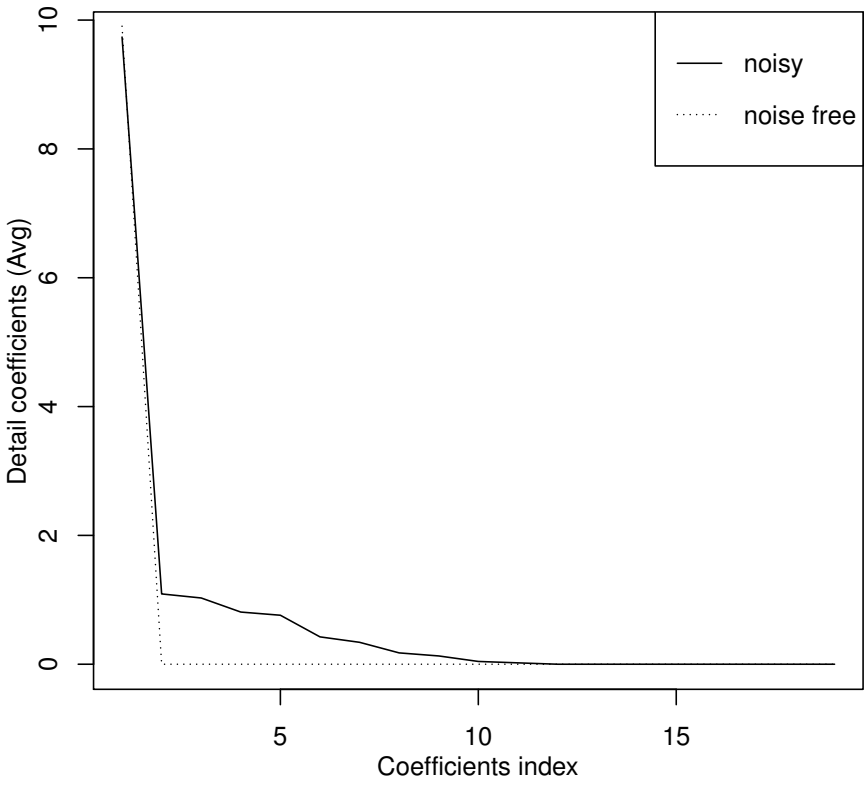


Figure 3·1: Compression curve for barbell network (dotted) and average compression curve for noisy barbell network (solid) using TGUH.

To demonstrate the robustness of this result, we simulated 100 such barbells perturbed with both Type I and Type II errors, i.e., declaring non-edges edges and vice versa. Here we set $P(\text{Type I error}) = 0.01$ and $P(\text{Type II error}) = (1 - \text{Den})/\text{Den} \times P(\text{Type I error})$, where Den is the density of adjacency matrix

of the noise-free barbell. Under this setting, the expected density of the 'noisy' network is the same as the original one. Figure 3·1 shows the average compression curve resulting from applying the TGUH to these noisy networks. We see that this curve is qualitatively quite similar to that for the noise-free version of these networks.

3.4.2 Rate of compression

In fact, we could derive the compression rate, defined as the number of zero entries in $\mathbf{W}^0$ over the number of zero entries in the transformed adjacency matrix $\mathbf{W}^\ell$, for any given step using the TGUH for a noise free 'barbell' type of network. Let $\mathbf{W}^0$ be the adjacency matrix corresponding to a network with two connected components of size N_1 and N_2 joint by a single link. Specifically, we note that at the second to the last step, the transformed adjacency matrix $\mathbf{W}^{L-1}$ is given by:

$$\mathbf{W}^{L-1} = \begin{bmatrix} N_1 & \cdots & 0 & \frac{1}{\sqrt{N_1 N_2}} & \frac{\sqrt{N_2-1}}{\sqrt{N_1 N_2}} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -\frac{\sqrt{N_1-1}}{\sqrt{N_1 N_2}} & -\frac{\sqrt{(N_1-1)(N_2-1)}}{\sqrt{N_1 N_2}} & \ddots & 0 \\ \frac{1}{\sqrt{N_1 N_2}} & \cdots & -\frac{\sqrt{N_1-1}}{\sqrt{N_1 N_2}} & N_2 & 0 & \cdots & 0 \\ \frac{\sqrt{N_2-1}}{\sqrt{N_1 N_2}} & \cdots & -\frac{\sqrt{(N_1-1)(N_2-1)}}{\sqrt{N_1 N_2}} & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

and the compression rate is: $\frac{2N_1N_2-8}{N_1^2+N_2^2-2}+1$, which attends the minimum of all steps. Note that asymptotically we have this compression rate goes to some constant C if N_1 and N_2 go to infinity in the same order.

3.4.3 EEG data on a DTI network

We now provide an application of using the TGUH to denoise EEG signals over a DTI network. Understanding the relationship between brain anatomical connectivity and brain dynamics remains an active research area [18][50]. In general, different frequency bands have been associated with different brain function [14], and different spatial organization over the brains surface, but how brain anatomical connectivity relates to brain rhythms remains incompletely understood.

As an example application of the method developed here, we consider two types of data collected from a human subject. Brain anatomical connectivity was computed from high resolution diffusion tensor imaging data using previously described methods [18]. Briefly, 324 regions of interest (ROI) at the gray-white matter border were selected using the topology of a recursively subdivided icosahedron fitted in the subjects cortical surface inflated to a sphere [27][35]. Quantitative bidirectional white matter connectivity between each ROI pair was computed using Probtrackx2 software [6] where 500 streamlines were sampled per voxel within each ROI. A connectivity index was then computed for each ROI pair as the proportion of successful streamlines connected between the ROI pair over the total number of streamlines sampled.

Dynamic brain electrical activity was collected from the same ROIs using the MNE-C and Freesurfer software packages [27][37] and previously reported methods [18]. Briefly, EEG data during stage 2 sleep was recorded with a 70 channel EEG cap and electrode positions collected using a 3D digitizer. Anatomical cortical surfaces of the brain were reconstructed using T1-weighted MRI data and a forward solution was calculated using a three-layer boundary element model consisting of the inner skull, outer skull and outer skin surfaces. The digitized

EEG electrode coordinates were coregistered to the reconstructed surface using the nasion and auricular points. The inverse operator was computed from the forward model and the resultant current estimates at each ROI calculated. Ten seconds of artifact free data were selected for analysis. From the electrical source estimates at each ROI, we computed the power spectrum using the multitaper method (bandwidth 1 Hz, 9 tapers). We then compute the average power in the theta band (5-8 Hz), alpha band (8-12 Hz), beta band (12-20 Hz), and gamma band (20-40 Hz).

The DTI network contains 324 nodes and 1487 edges. Power in each spectral band was compressed using the TGUH bases on the DTI network. The resulting compression curves appear in Figure 3·2. In all four bands, half of the signals were captured using the leading 50 basis functions.

To illustrate TGUH denoising, we only show the result of denoising the alpha band signal, in Figure 3·3. There the size of the nodes indicates the strength of the signal. Using the theoretical threshold suggested by Theorem 3.3.1, signal on only 10 nodes in the occipital cortex remain; the rest has been eliminated. The intuition is that the cluster of electrodes with increased alpha power likely represents the posterior dominant rhythm in this area.

3.5 Discussion

In this section, we develop a new class of graph wavelets, based on adaptive unbalanced Haar transformations, and establish consistency for estimating appropriate functions over a network. Theoretical analysis, simulation, and real data application in the context of computational neuroscience suggest the promise of this class. Our algorithm is especially useful in the case where the signal behaves

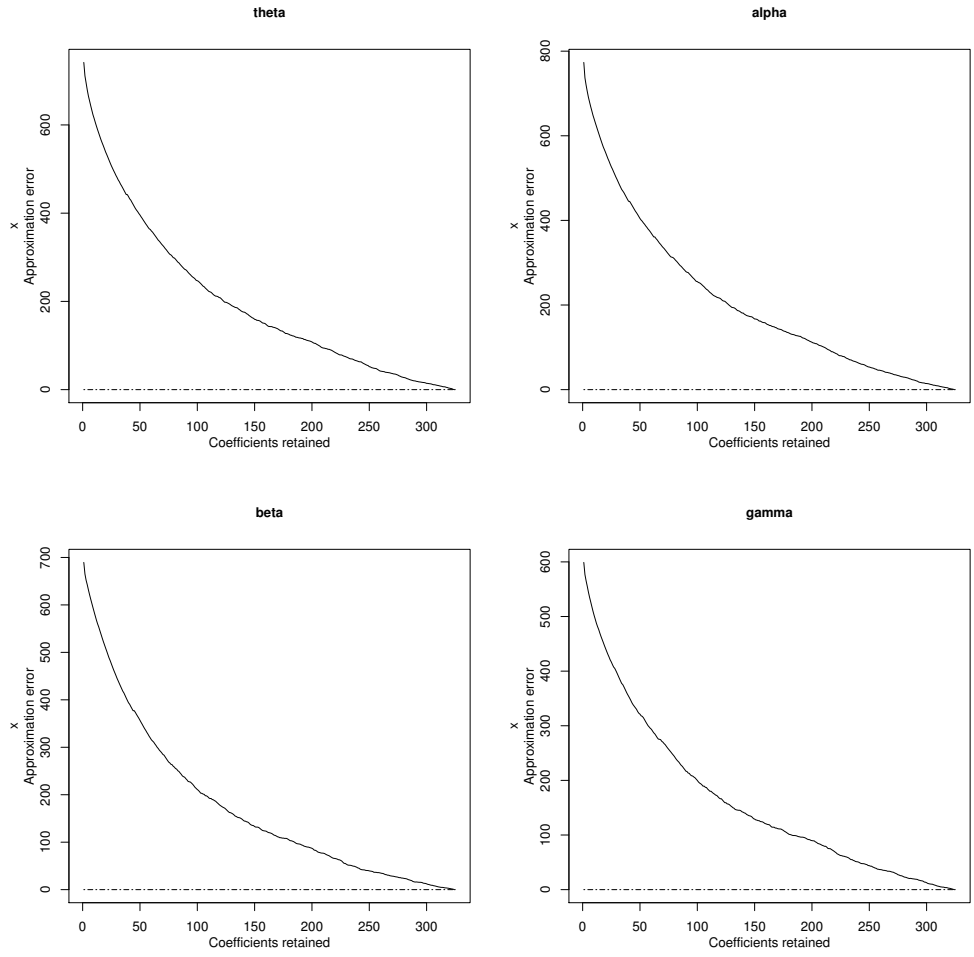
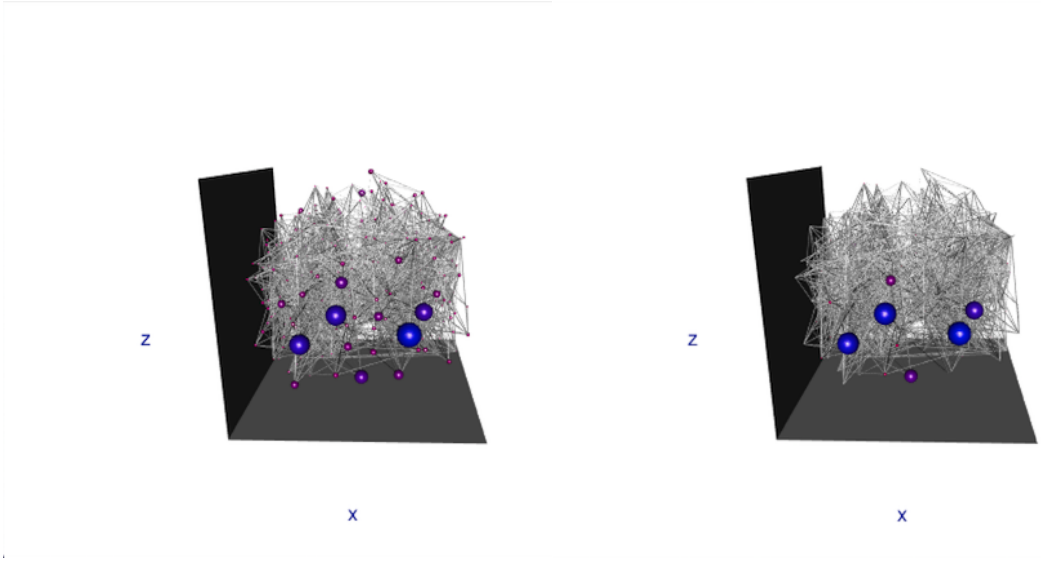


Figure 3·2: Compressed spectral bands of the TGUH bases

differently at different scales and these scales correspond to analogous variations in the underlying network topology.

Even with the current advancement in this now-highly-active space, the interaction of network topology, basis and signal is still an under-explored area. In future work, denoising of the network itself seems a natural extension here, although theoretical analysis even in toy cases is challenging. Connections to graph coarsening and visualization are possible as well.



(a) Noisy signal **(b)** Denoised signal

Figure 3·3: (a)Strength of the alpha band signal of the DTI network; (b) Denoised strength of the alpha band signal of the DTI network.

Chapter 4

Spectral Bootstrapping for Networks Observed with Measurement Errors

4.1 Introduction

Network representation is popular for characterizing complex systems. However it is not uncommon that errors observed in the original measurements will propagate to network statistics and hence induce uncertainties in the summaries of the networks. The two common types of error are the Type I error and the Type II error. In the network context, a Type I error is declaring an edge when none exists and a Type II error is omitting an edge when it exists. Such errors arise from different contexts and are quite common, for example, the studies of social networks suffer from subjectivity of participants and recording errors [25][78][79] and the studies of biological networks, such as the connectivity network constructed from fMRI data, often contain noise corresponding to missing or false connections [68][76].

In this work, we consider an undirected network $\mathbf{G}$ that has been polluted by both Type I and Type II errors in terms of observing edges and propose a spectral-denoising based resampling method to compute confidence intervals for a number of Lipschitz continuous network statistics $g(\cdot)$ of the observed networks adjacency matrix $\mathbf{W}_{obs}$. In particular, we extend the idea of Balachandran, Airoldi and Kolaczyk [3], where the authors proposed a spectral-based method to

denoise the observed network data and quantified the statistical risk of estimating Lipschitz continuous network summary statistics. This work laid out the theoretical foundation for us and we adopt their idea to separate 'signal' from 'noise' of the observed adjacency matrix. Our particular contribution is that we propose a novel idea to characterize the distribution of network summary statistics by bootstrapping the observed network adjacency matrix.

Bootstrapping method has been studied extensively since it was invented by Efron[26]. It was originally proposed for i.i.d data but has extensions to temporal and spatial data, for example, see[52] [56]. Later, a couple of studies have been done on bootstrapping networks. Recent work includes applying the temporal and spatial bootstrap method on random graphs, see [77], and bootstrapping count features of networks, see [9]. Other work, for example, Friedman, Goldszmidt and Wyner [29] and Friedman et al. [30], applied the bootstrap method to compute confidence measures on features of inducing networks from a Bayesian perspective. In our case, we bootstrap the noise and generate bootstrapped networks by adding the noise back to the signal. Based on the resulting bootstrapped adjacency matrices, we calculate bootstrap distributions for various Lipschitz continuous network summary statistics. The reason that only Lipschitz continuous statistics is considered here is that it allows us to control the accuracy in estimating $g(\mathbf{W}_{true})$.

The organization of this paper is as follows. In section II, we provide notations and sketch the procedure of spectral bootstrapping for a network adjacency matrix. In section III, we illustrate the utility of the procedure and evaluate it through a series of simulation studies. In section IV, we discuss potential extensions.

4.2 Spectral bootstrapping of network

4.2.1 Notation

Suppose we observe an undirected, $\{0,1\}$-valued network $\mathbf{G}$ of size $N \times N$ with noise. Let $\mathbf{W}_{obs}$, $\mathbf{W}_{true}$ and $\mathbf{W}_{noise}$ denote the observed adjacency matrix, the underlying true adjacency matrix and the noise matrix respectively. We then have

$$\mathbf{W}_{obs} = \mathbf{W}_{true} + \mathbf{W}_{noise},$$

Note that $\mathbf{W}_{noise}$ is an additive noise matrix with entries such that

- $\mathbf{W}_{noise}(i,j) \sim -Bern(p)$, if $\mathbf{W}_{true}(i,j) = 1$;
- $\mathbf{W}_{noise}(i,j) \sim Bern(q)$, if $\mathbf{W}_{true}(i,j) = 0$.

Here q is the probability of committing a type I error and p is the probability of committing a type II error. We then define $\mathbf{W}_{K_N}$ to be a $N \times N$ matrix of ones with zero diagonals. Balachandran, Airoldi and Kolaczyk [3] showed that $\hat{\mathbf{W}}_{obs} = \frac{\mathbf{W}_{obs} - q\mathbf{W}_{K_N}}{1-(p+q)}$ is an entry-wise unbiased estimator of the true adjacency matrix $\mathbf{W}_{true}$.

4.2.2 Entry-wise Spectral Bootstrap

Ideally, we assume that the underlying true network adjacency matrix is known. We denote the eigensystem of $\mathbf{W}_{true}$ by $\{\psi_i, \lambda_i\}_{i=1}^n$, where $\{\psi_i\}$ and $\{\lambda_i\}$ are the collections of eigenvectors and eigenvalues of $\mathbf{W}_{true}$ respectively. Without loss of generality, we assume that $\lambda_1^2 \geq \lambda_2^2 \geq \cdots \geq \lambda_n^2$.

We then define our estimator $\mathbf{W}_{ideal,s}$ of $\mathbf{W}_{true}$ using the first s modes of $\hat{\mathbf{W}}_{obs}$, to be

$$\hat{\mathbf{W}}_{ideal,s} = \sum_{i=1}^{s} \langle \psi_i, \hat{\mathbf{W}}_{obs}\psi_i \rangle \psi_i \psi_i^T.$$

However, in the real world, it is not realistic to observe $\mathbf{W}_{true}$ and have access to the true eigensystem $\{\psi_i, \lambda_i\}_{i=1}^n$. Alternatively, we use the empirical eigensystem computed from $\hat{\mathbf{W}}_{obs}$.

Let $\{\phi_i, \mu_i\}_{i=1}^n$ be the eigensystem of $\hat{\mathbf{W}}_{obs}$, ordered descending in $\{\mu_i^2\}_{i=1}^n$. Define our estimator of $\mathbf{W}_{true}$ to be:

$$\hat{\mathbf{W}}_{obs,s} = \sum_{i=1}^{s} \langle \phi_i, \hat{\mathbf{W}}_{obs}\phi_i \rangle \phi_i \phi_i^T.$$

Balachandran, Airoldi and Kolaczyk [3] also showed that if

- $p + q < 1$,
- and the noise is independent,
- and $\log(N) \leq \delta(1-p) + q(N-1-\delta)$,

then the relative error of estimating $\mathbf{W}_{true}$ by $\hat{\mathbf{W}}_{obs,s}$ using the matrix norm is bounded and the smallest error is achieved if $s = 1$ asymptotically.

Next, define our estimator $\hat{\mathbf{W}}_{noise}$ of $\mathbf{W}_{noise}$ to be

$$\hat{\mathbf{W}}_{noise} = \hat{\mathbf{W}}_{obs} - \hat{\mathbf{W}}_{obs,s}.$$

We then bootstrap the noise matrix $\hat{\mathbf{W}}_{noise}$ by resampling the empirical eigenvectors with replacement. The spectral bootstrapping procedure is given by algorithm 3.

Having obtained the bootstrapped samples, we compute the bootstrap-based empirical distribution and confidence interval of the network statistics $g(\cdot)$, where statistical inference can be made.

4.2.3 Simulation Study

In this section, we use simulation studies to establish a simple proof of concept regarding the utility of the spectral bootstrapping algorithm. We illustrate our

Require: $\hat{\mathbf{W}}_{obs}$,$\hat{\mathbf{W}}_{noise}$

1: **Define** B to be the number of bootstrapped samples;

2: **Define** M to be the number of bootstrapped noise we need to resample: $M = \binom{N}{2} + N - s$;

Define $g(\cdot)$ to be a network statistic that is Lipschitz continuous;

3: **for** $i = 1 : B$ **do**

4: **Sample** $\{(i_k, j_k)\}_{k=1}^{M}$, $i_k \neq j_k$, $i_k, j_k \in \{1, 2, ..., N\}$;

5: **Compute** the bootstrapped noise matrix using $\hat{\mathbf{W}}^{B}_{noise} = \sum_{k=1}^{M} \langle \phi_{i_k}, \hat{\mathbf{W}}_{noise} \phi_{j_k} \rangle \phi_{i_k} \phi_{j_k}^{T}$;

6: **Construct** the bootstrapped network adjacency matrix: $\hat{\mathbf{W}}^{B} = \hat{\mathbf{W}}_{obs,s} + \hat{\mathbf{W}}^{B}_{noise}$;

7: **end for**

Algorithm 3: Spectral bootstrapping of networks

proposed method under four different settings. The first graph is a 2-regular graph, where each vertex has exactly two neighbors. This proposed graph consists of a union of disjoint cycles. The second graph we consider is a power-law graph such that $P(f_d) \sim d^{-3}$, where f_d is the fraction of vertices with degree d. The third graph we simulate is an Erdös-Rényi graph with $\pi = 0.1$, where π is the probability of any pair of vertices being connected. The last graph we have is generated using the stochastic block model with 2 blocks of the same size. We set the probability of joining vertices from the same community to be $\pi_{within} = 0.1$ and the probability of joining vertices from different communities to be $\pi_{between} = 0.005$. In all cases, the graph order N are set to be 50.

For each bootstrapped sample, we compute three (locally) Lipschitz continuous network statistics.

1. **Average Degree**: the average degree of a graph measures the density of the graph.

$$\frac{1}{N} \sum_{v \in \mathbf{G}} d_v$$

2. **Fiedler Value**: The Fiedler value, also known as the algebraic connectivity is the second smallest eigenvalue of the laplacian matrix L of the graph, which is defined as: $L = D - A$, with D the diagonal degree matrix and A the adjacency matrix. The Fielder value measures how "knit" the network is connected.

3. **Average eigenvector centrality of a fixed vertex**: Eigenvector centrality measures the importance of a given vertex. Without loss of generality, we focus on the vertex corresponding to the first row/column of each bootstrapped adjacency matrix. The eigenvector centrality of vertex i is defined as:

$$x_i = \frac{1}{\mu_1} \sum_{j \neq i} \mathbf{W}_{ij} x_j$$

where again μ_1 is the leading eigenvalue of the adjacency matrix $\mathbf{W}$ and x_j's are the eigenvector centralities of vertex j, with the sum over j adjacent to i.

For each type of network, we first simulate the true network $\mathbf{W}_{true}$. Then we pollute it with errors, where $q = 0.01$ to generate type I error and then $p = 0.235$ to generate type II error. The proposed order and values do not change the expected density of the original network. We set the size of each bootstrapped sample to be $B = 200$ and will test it for $K = 500$ times to compute for the coverage probability, defined as the proportion of the time that the confidence interval contains its true value. The number of modes s kept in the simulation study is set to be 2, which empirically works well and simple. The results are summarized in Table 4.1. The actual coverage probability is higher or similar to the nominal coverage probability in all three summary statistics for the Erdös-Rényi graph model and the stochastic block model, which indicates that our

method works fairly well for these two type of graphs as expected. The reason is that the randomness of the simulated errors does not change the structure of the graph too much as the edges in the true graph are simulated with a fixed parameter π for the Erdös-Rényi graph model and two fixed parameter π_{in} and $\pi_{between}$ for the stochastic block model. The proposed way of simulating the type I and type II errors does not change the type and the parameter for the Erdös-Rényi graph model and does not change the type of graph but slightly change the actual parameter for the stochastic block model. However, for the 2-regular graph and the power-law graph, the connectivity of the graph and the importance of nodes are changed by adding both the type I and the type II error, leading to worse actual coverage probabilities.

Type of graph	**Network summary statistics**	**Coverage**
2-Regular graph	Average degree	0.986
	Eigenvector centrality of vertex 1	0.984
	Fiedler value	0.754
Power-law graph	Average degree	0.980
	Eigenvector centrality of vertex 1	0.628
	Fiedler value	0.574
Erdős-Rényi graph	Average degree	0.978
	Eigenvector centrality of vertex 1	0.966
	Fiedler value	0.966
Stochastic block model	Average degree	0.988
	Eigenvector centrality of vertex 1	0.936
	Fiedler value	0.958

Table 4.1: Observed coverage for 95%-CI of network summary statistics using the bootstrapped samples

4.3 Discussion

In this thesis, we develop a new procedure for making statistical inference of summary statistics on errorful network , based on bootstrapping the spectral decomposed network adjacency matrix. Note that in our work, we need to know the exact probability of committing a type I error (q) and a type II error in order to get an unbiased estimator of the true network, which is false practically. In the real world, p and q can be chosen empirically using domain knowledge.

Through the simulation study, we show that our method is especially useful for assessing network statistics for the random graph model. In future work, the theoretical behavior of the proposed method need to be addressed appropriately, although even in toy cases require non-trivial effort and advanced mathematical tools. To the knowledge of the author, the connection between the degree distribution and the noise level plays an important role in the proposed settings.

Chapter 5

Conclusions

5.1 Summary of the thesis

This dissertation addresses three statistical inference problems of networks. The three problems are somewhat related but have clear distinctions.

In chapter 2, we propose a method for simultaneous network inference and change point detection of non-stationary multiple time series data. Specifically, we adopt a causal network type of model and incorporate them within a multi-scale framework. We formulate the problem as finding the best partition-based multi-scale dynamic causal network model that captures the dynamics of a system in a way that is sensitive to changes at multiple scales. More specifically, we partition the non-stationary time axis into blocks of independent, stationary time series, where a VAR type of model can be assumed. We impose a counting penalty to penalize the number of blocks used in the method to prevent overfitting. Then, within each blocks we do neighborhood selection for each elements using a group-lasso type of estimator. Consistency result in estimating the change points and Type I error control are also provided. Our simulation and the application in a MEG data reveals that our proposed method is sensitive in capturing causal structure changes at various time scales. Applications of the method include functional neuroimaging, financial economics, genomics, or any other field where multivariate time series and spatial temporal data is observed.

Various extensions are worth exploring here. For example, a fused-lasso type

of penalty would be good to try as it encourage a certain notion of temporal smoothness and shares similar mathematical properties as we have for the group-lasso based penalty[33]. Moreover, a faster implementation would be desirable, especially for the non-dyadic case where the computation time is in the order of $\mathcal{O}(T^3)$. Linear computation time may perhapes be achieved using ideas like those presented by [44].

In chapter 3, we extend the one dimensional TGUH algorithm and construct a new class of graph wavelets for network data. The network TGUH is based on adaptive unbalanced Haar transformations. This result is an efficient and nice compression of the adjacency matrix where denoising of signals is a natural topic to explore. In fact, we show that if a graph signal varies in a way similar to the network structure we used to find our TGUH bases, we have good signal compression when transforming the signal through the resulting TGUH bases. We also provided an application of using the TGUH to denoise EEG signals over a DTI network use the suggested theoretical threshold. It shows that our algorithm is particular useful when the signal behaves differently at different scales and these scales are somewhat 'consistent' to the underlying network topology.

In future work, denoising of network itself seems a natural aspect to explore. Our simulation result on compressing the barbell type of network suggests that the TGUH compresses the network effectively in the noise free context and may work as well with simple flipped edges as long as the network topology does not change too abruptly. However, theoretical analysis is still challenging even in the toy cases. Another direction worth exploring is the connection to graph coarsening and visualizations, especially in the case where structure of the underlying graph changes at different scales.

In chapter 4, we propose a spectral-based method using the adjacency ma-

trix to denoise the observed network data and make inference on certain type of network summary statistics by bootstrapping the estimated noises. For the denoising part, we follow the method proposed by [3], where signal and noise are separated through the eigen-decomposition of an unbiased estimator of the observed network adjacency matrix. Our bootstrap noise sample is then generated by resampling the entries of the estimated noise matrix. Through the study of a series of carefully designed simulations, we show that the proposed method works well, especially in the case when the degree distribution is preserved when noise is introduced. However, no theoretical results have been established for this type of estimators.

Appendix A

Proof

A.1 Dynamic Networks with Multi-scale Temporal Structure

A.1.1 Algorithm using RDP

Here we provide the algorithm for implementation based on recursive dyadic partitions. Assume the length of the time series equals $T = 2^J$ and $j_{min} = \min_j$ such that $2^j > p+1$. Note that $p+1$ is the minimum required number of observations to fit the restricted VAR(p) model. Assume $J > j_{min}$,

Data: $\mathbf{X}(u)$, $\mathbf{X}(-u)$, p

Result: $\hat{\boldsymbol{\theta}}_{RDP}$

for $i = 0 : 2^{(J-j_{min})} - 1$ **do**

Fit restricted VAR(p) model for $x_I(u)$, for

$I = \{t : t \in [2^{j_{min}} * i + 1, 2^{j_{min}} * (i + 1)]\}$ Compute and store pl_I on each

interval I;

optimumModel $\leftarrow pl_I$;

end

for $j = J - j_{min} - 1 : 0$ **do**

for $i = 0 : 2^j - 1$ **do**

Fit restricted VAR(p) model for $\mathbf{X}_I(u)$, for

$I = \{t : t \in [2^{(J-j)} * i + 1, 2^{(J-j)} * (i + 1)]\}$;

Compute and store pl_I on each interval I;

if $pl_I \leq pl_{I_l^i} + pl_{I_r^i} + Penalty$ **then**

optimumModel $\leftarrow pl_I$;

Update changePoint;

else

optimumModel $\leftarrow pl_l$ and pl_r;

Update changePoint;

end

end

end

Algorithm 4: Multiscale dynamic causal network using RDP

Algorithm 4 splits only at dyadic positions. The candidate partitions $\mathcal{P} \preceq \mathcal{P}^*_{D_y}$ can be represented as subtrees of a binary tree of depth $\log_2 T$. Given a dataset of length $T = 2^J$, we have 2^0 root node, 2^1 nodes at level 1, 2^2 nodes, 2^3 nodes, and so on, at the following levels, until we reach the leaf level, which has $2^{(J-1)}$ nodes. The complexity of the algorithm is then of order $\mathcal{O}(T)$ calls to fit

the group lasso regression and $\mathcal{O}(T)$ calls for comparisons.

A.1.2 Proof of theorem 2.3.1

Proof. Theorem 2.3.1

The proof contains two parts. In the first part, we show that equation (2.7) holds, under H_0. In the second part, we show that equation (2.8) holds, under H_1.

Part 1

We begin by defining the group lasso penalized likelihood on an interval I:

$$PL_I = \frac{1}{|I|} \|\mathbf{X}_I(u) - \mathbf{X}_I(-u)\boldsymbol{\theta}_I(u,v)\|_2^2 + \lambda_I \sum_{v \in V \setminus \{u\}} \|\boldsymbol{\theta}_I(u,v)\|_2 . \tag{A.1}$$

Let $\hat{\boldsymbol{\theta}}_{1:T}$ be the $\boldsymbol{\theta}$ that minimizes the penalized likelihood (A.1) on the interval from 1 to T and $\hat{PL}_{1:T}$ be the quantity upon substituting $\hat{\boldsymbol{\theta}}_{1:T}$ in equation (A.1). Consider any alternative model with a change point detected at point $\hat{\tau} \in (1, T)$. Denote by $\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}$ and $\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}$ the coefficients $\boldsymbol{\theta}$ that minimize equation (A.1) over intervals $[1, \hat{\tau}]$ and $(\hat{\tau}, T]$, respectively. Given our model, equation (2.7) in theorem 2.3.1 is equivalent to

$$\mathbb{P}_{H_0}(\hat{PL}_{1:T} \leq \hat{PL}_{1:\hat{\tau}} + \hat{PL}_{\hat{\tau}:T} + C_3 \log T) \longrightarrow 1.$$

The additional term $C_3 \log T$ comes from the fact that the alternative model has 1 more partition than the null model, with $C_3 = 1/2$ using RDP and $C_3 = 3/2$

using RP. We expand $\hat{PL}_{1:\hat{\tau}} + \hat{PL}_{\hat{\tau}:T} - \hat{PL}_{1:T} + C_3 \log T$ and get:

$$\begin{aligned}
& \frac{1}{\hat{\tau}} \left\| \mathbf{X}_{1:\hat{\tau}}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{1:\hat{\tau}}(v) \hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v) \right\|_2^2 + \lambda_{1:\hat{\tau}} \sum_{v \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v) \right\|_2 \\
+ \quad & \frac{1}{T - \hat{\tau}} \left\| \mathbf{X}_{\hat{\tau}:T}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{\hat{\tau}:T}(v) \hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v) \right\|_2^2 + \lambda_{\hat{\tau}:T} \sum_{v \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v) \right\|_2 \\
- \quad & \frac{1}{T} \left\| \mathbf{X}_{1:T}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{1:T}(v) \hat{\boldsymbol{\theta}}_{1:T}(u,v) \right\|_2^2 - \lambda_{1:T} \sum_{v \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{1:T}(u,v) \right\|_2 \\
+ \quad & C_3 \log T. \qquad \text{(A.2)}
\end{aligned}$$

By rewriting the last two lines of equation (A.2), we have

$$\begin{aligned}
& \frac{1}{\hat{\tau}} \left\| \mathbf{X}_{1:\hat{\tau}}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{1:\hat{\tau}}(v) \hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v) \right\|_2^2 + \lambda_{1:\hat{\tau}} \sum_{v \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v) \right\|_2 \\
+ \quad & \frac{1}{T - \hat{\tau}} \left\| \mathbf{X}_{\hat{\tau}:T}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{\hat{\tau}:T}(v) \hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v) \right\|_2^2 + \lambda_{\hat{\tau}:T} \sum_{v \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v) \right\|_2 \\
- \quad & \frac{1}{T} \left\| \mathbf{X}_{1:\hat{\tau}}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{1:\hat{\tau}}(v) \hat{\boldsymbol{\theta}}_{1:T}(u,v) \right\|_2^2 \\
- \quad & \frac{1}{T} \left\| \mathbf{X}_{\hat{\tau}:T}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{\hat{\tau}:T}(v) \hat{\boldsymbol{\theta}}_{1:T}(u,v) \right\|_2^2 \\
- \quad & \lambda_{1:T} \sum_{v \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{1:T}(u,v) \right\|_2 + C_3 \log T. \qquad \text{(A.3)}
\end{aligned}$$

We then add and subtract a term in both line 3 and line 4 of equation (A.3). In doing so, we have:

$$
\begin{aligned}
& \frac{1}{\hat{\tau}} \left\| \mathbf{X}_{1:\hat{\tau}}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{1:\hat{\tau}}(v) \hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v) \right\|_2^2 + \lambda_{1:\hat{\tau}} \sum_{v \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v) \right\|_2 \\
+ \quad & \frac{1}{T - \hat{\tau}} \left\| \mathbf{X}_{\hat{\tau}:T}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{\hat{\tau}:T}(v) \hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v) \right\|_2^2 + \lambda_{\hat{\tau}:T} \sum_{v \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v) \right\|_2 \\
- \quad & \frac{1}{T} \left\| \mathbf{X}_{1:\hat{\tau}}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{1:\hat{\tau}}(v) \hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v) + \sum_{v \in V \setminus \{u\}} \mathbf{X}_{1:\hat{\tau}}(v) \hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{1:\hat{\tau}}(v) \hat{\boldsymbol{\theta}}_{1:T}(u,v) \right\|_2^2 \\
- \quad & \frac{1}{T} \left\| \mathbf{X}_{\hat{\tau}:T}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{\hat{\tau}:T}(v) \hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v) + \sum_{v \in V \setminus \{u\}} \mathbf{X}_{\hat{\tau}:T}(v) \hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{\hat{\tau}:T}(v) \hat{\boldsymbol{\theta}}_{1:T}(u,v) \right\|_2^2 \\
- \quad & \lambda_{1:T} \sum_{v \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{1:T}(u,v) \right\|_2 + C_3 \log T. \qquad \text{(A.4)}
\end{aligned}
$$

From which we have that equation (A.4)

$$
\begin{aligned}
&\geq \frac{1}{\hat{\tau}}\left\|\mathbf{X}_{1:\hat{\tau}}(u)-\sum_{v\in V\setminus\{u\}}\mathbf{X}_{1:\hat{\tau}}(v)\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v)\right\|_2^2\\
&+ \frac{1}{T-\hat{\tau}}\left\|\mathbf{X}_{\hat{\tau}:T}(u)-\sum_{v\in V\setminus\{u\}}\mathbf{X}_{\hat{\tau}:T}(v)\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v)\right\|_2^2\\
&- \frac{1}{T}\left\|\mathbf{X}_{1:\hat{\tau}}(u)-\sum_{v\in V\setminus\{u\}}\mathbf{X}_{1:\hat{\tau}}(v)\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v)\right\|_2^2\\
&- \frac{1}{T}\left\|\sum_{v\in V\setminus\{u\}}\mathbf{X}_{1:\hat{\tau}}(v)\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v)-\sum_{v\in V\setminus\{u\}}\mathbf{X}_{1:\hat{\tau}}(v)\hat{\boldsymbol{\theta}}_{1:T}(u,v)\right\|_2^2\\
&- \frac{2}{T}\left(\left\|\mathbf{X}_{1:\hat{\tau}}(u)-\sum_{v\in V\setminus\{u\}}\mathbf{X}_{1:\hat{\tau}}(v)\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v)\right\|_2\right.\\
&\quad\left.\times\left\|\sum_{v\in V\setminus\{u\}}\mathbf{X}_{1:\hat{\tau}}(v)\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v)-\sum_{v\in V\setminus\{u\}}\mathbf{X}_{1:\hat{\tau}}(v)\hat{\boldsymbol{\theta}}_{1:T}(u,v)\right\|_2\right)\\
&- \frac{1}{T}\left\|\mathbf{X}_{\hat{\tau}:T}(u)-\sum_{v\in V\setminus\{u\}}\mathbf{X}_{\hat{\tau}:T}(v)\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v)\right\|_2^2\\
&- \frac{1}{T}\left\|\sum_{v\in V\setminus\{u\}}\mathbf{X}_{\hat{\tau}:T}(v)\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v)-\sum_{v\in V\setminus\{u\}}\mathbf{X}_{\hat{\tau}:T}(v)\hat{\boldsymbol{\theta}}_{1:T}(u,v)\right\|_2^2\\
&- \frac{2}{T}\left(\left\|\mathbf{X}_{\hat{\tau}:T}(u)-\sum_{v\in V\setminus\{u\}}\mathbf{X}_{\hat{\tau}:T}(v)\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v)\right\|_2\right.\\
&\quad\left.\times\left\|\sum_{v\in V\setminus\{u\}}\mathbf{X}_{\hat{\tau}:T}(v)\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v)-\sum_{v\in V\setminus\{u\}}\mathbf{X}_{\hat{\tau}:T}(v)\hat{\boldsymbol{\theta}}_{1:T}(u,v)\right\|_2\right)\\
&+ \lambda_{\hat{\tau}:T}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v)\right\|_2+\lambda_{1:\hat{\tau}}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v)\right\|_2-\lambda_{1:T}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{1:T}(u,v)\right\|_2\\
&+ C_3\log T. \qquad\qquad (A.5)
\end{aligned}
$$

Under assumptions (1) to (5), [2] reformulated the group lasso penalized

likelihood (A.1) as:

$$PL_I = \hat{\boldsymbol{\Sigma}}_{\mathbf{X}(u)\mathbf{X}(u)} - 2\hat{\boldsymbol{\Sigma}}'_{\mathbf{X}(-u)\mathbf{X}(u)}\boldsymbol{\theta} + \boldsymbol{\theta}'\hat{\boldsymbol{\Sigma}}_{\mathbf{X}(-u)\mathbf{X}(-u)}\boldsymbol{\theta} + \lambda_I \sum_{v\in V\setminus\{u\}} \|\boldsymbol{\theta}(u,v)\|_2 \tag{A.6}$$

where $\hat{\boldsymbol{\Sigma}}_{\mathbf{X}(u)\mathbf{X}(u)} = \frac{1}{|I|}\mathbf{X}(u)'\Pi_{|I|}\mathbf{X}(u)$, $\hat{\boldsymbol{\Sigma}}_{\mathbf{X}(-u)\mathbf{X}(u)} = \frac{1}{|I|}\mathbf{X}(-u)'\Pi_{|I|}\mathbf{X}(u)$ and $\boldsymbol{\theta}'\hat{\boldsymbol{\Sigma}}_{\mathbf{X}(-u)\mathbf{X}(-u)}\boldsymbol{\theta} = \frac{1}{|I|}\mathbf{X}(-u)'\Pi_{|I|}\mathbf{X}(-u)$ are the empirical covariance matrices with $\Pi_{|I|}$ defined as $\Pi_{|I|} = \mathbf{I}_{|I|} - \frac{1}{|I|}\mathbf{1}_{|I|}\mathbf{1}'_{|I|}$ and showed that the group lasso estimator $\hat{\boldsymbol{\theta}}$ converges in probability to $\boldsymbol{\theta}$. Using expression in (A.6) and collecting similar terms, we could then rewrite (A.5) as:

$$\begin{aligned}
& \frac{T-\hat{\tau}}{T}\left\{\hat{\boldsymbol{\Sigma}}_{\mathbf{X}_{1:\hat{\tau}}(u)\mathbf{X}_{1:\hat{\tau}}(u)} - 2\hat{\boldsymbol{\Sigma}}_{\mathbf{X}_{1:\hat{\tau}}(-u)\mathbf{X}_{1:\hat{\tau}}(u)}\hat{\boldsymbol{\theta}}_{1:\hat{\tau}} + \hat{\boldsymbol{\theta}}'_{1:\hat{\tau}}\hat{\boldsymbol{\Sigma}}_{\mathbf{X}_{1:\hat{\tau}}(-u)\mathbf{X}_{1:\hat{\tau}}(-u)}\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}\right\} \\
+ \quad & \frac{\hat{\tau}}{T}\left\{\hat{\boldsymbol{\Sigma}}_{\mathbf{X}_{\hat{\tau}:T}(u)\mathbf{X}_{\hat{\tau}:T}(u)} - 2\hat{\boldsymbol{\Sigma}}_{\mathbf{X}_{\hat{\tau}:T}(-u)\mathbf{X}_{\hat{\tau}:T}(u)}\hat{\boldsymbol{\theta}}_{\hat{\tau}:T} + \hat{\boldsymbol{\theta}}'_{\hat{\tau}:T}\hat{\boldsymbol{\Sigma}}_{\mathbf{X}_{\hat{\tau}:T}(-u)\mathbf{X}_{1:\hat{\tau}}(-u)}\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}\right\} && \text{(A.7)} \\
- \quad & \left\|\hat{\boldsymbol{\Sigma}}^{1/2}_{\mathbf{X}_{1:\hat{\tau}}(-u)\mathbf{X}_{1:\hat{\tau}}(-u)}\left(\hat{\boldsymbol{\theta}}_{1:\hat{\tau}} - \hat{\boldsymbol{\theta}}_{1:T}\right)\right\|_2^2 - \left\|\hat{\boldsymbol{\Sigma}}^{1/2}_{\mathbf{X}_{\hat{\tau}:T}(-u)\mathbf{X}_{\hat{\tau}:T}(-u)}\left(\hat{\boldsymbol{\theta}}_{\hat{\tau}:T} - \hat{\boldsymbol{\theta}}_{1:T}\right)\right\|_2^2 && \text{(A.8)} \\
- \quad & \frac{2}{T}\left(\left\|\mathbf{X}_{1:\hat{\tau}}(u) - \sum_{v\in V\setminus\{u\}}\mathbf{X}_{1:\hat{\tau}}(v)\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v)\right\|_2 \left\|\sum_{v\in V\setminus\{u\}}\mathbf{X}_{1:\hat{\tau}}(v)\left(\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v) - \hat{\boldsymbol{\theta}}_{1:T}(u,v)\right)\right\|_2\right) && \text{(A.9)} \\
- \quad & \frac{2}{T}\left(\left\|\mathbf{X}_{\hat{\tau}:T}(u) - \sum_{v\in V\setminus\{u\}}\mathbf{X}_{\hat{\tau}:T}(v)\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v)\right\|_2 \left\|\sum_{v\in V\setminus\{u\}}\mathbf{X}_{\hat{\tau}:T}(v)\left(\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v) - \hat{\boldsymbol{\theta}}_{1:T}(u,v)\right)\right\|_2\right) && \text{(A.10)} \\
+ \quad & \lambda_{1:\hat{\tau}}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v)\right\|_2 + \lambda_{\hat{\tau}:T}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v)\right\|_2 - \lambda_{1:T}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{1:T}(u,v)\right\|_2 + C_3\log T. && \text{(A.11)}
\end{aligned}$$

Note that in the previous expression, the first two lines are by definition non-negative. The expression in the last line is composed of a collection of penalty terms. They are the group lasso penalties, and all of them converge to zero asymptotically assuming $\lambda_{(\cdot)} \longrightarrow 0$ and $\lambda_{(\cdot)}N \longrightarrow 0$.

Since $\hat{\boldsymbol{\theta}}_{1:\hat{\tau}} \xrightarrow{P} \boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}_{\hat{\tau}:T} \xrightarrow{P} \boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}_{1:T} \xrightarrow{P} \boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}_{1:\hat{\tau}} - \hat{\boldsymbol{\theta}}_{1:T} \xrightarrow{P} 0$ and X's have

finite moments up to order 4, each term in (A.8), (A.9) and (A.10) converges to 0 in probability.

Putting everything together, we then complete the proof of the first part of the theorem:

$$\mathbb{P}_{H_0}(\hat{PL}_{1:T} \leq \hat{PL}_{1:\hat{\tau}_i} + \hat{PL}_{\hat{\tau}:T} + C_3 \log T) \longrightarrow 1.$$

Part 2

Suppose H_1 is true. We denote the estimated change point by $\hat{\tau}$. We show that $\hat{PL}_{1:\hat{\tau}} + \hat{PL}_{\hat{\tau}:T}$ is minimized at $\hat{\tau} = \tau$. Assume we have a competing estimator $\tilde{\tau}$ with change point detected at time $\tilde{\tau} = s$ with $s \neq \tau$. We show that

$$\hat{PL}_{1:\hat{\tau}} + \hat{PL}_{\hat{\tau}:T} \leq \hat{PL}_{1:s} + \hat{PL}_{s:T} \tag{A.12}$$

holds with high probability under H_1. Without loss of generality, we assume that $\tau - s = \delta$, for some $\delta > 0$ as shown in figure A·1. For the case that $s > \tau$, a similar argument holds.

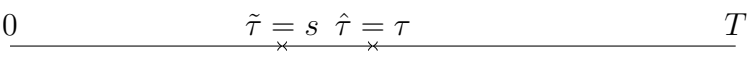


Figure A·1: Relative position of two detected change points

Denote by $\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}$ and $\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}$ the estimated coefficients that minimize the penalized likelihoods, given that $I = \{t : t \in [1, \hat{\tau})\}$ and $I = \{t : t \in [\hat{\tau}, T]\}$. We also define $\hat{\boldsymbol{\theta}}_{1:s}$ and $\hat{\boldsymbol{\theta}}_{s:T}$ to be the estimated coefficients that minimize the penalized likelihoods in A.1, given that $I = \{t : t \in [1, s)\}$ and $I = \{t : t \in [s, T]\}$. The key idea is that $\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}$ and $\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}$ are consistent estimators of $\boldsymbol{\theta}_{1:\tau}$ and $\boldsymbol{\theta}_{\tau:T}$ but $\hat{\boldsymbol{\theta}}_{s:T}$ is not a consistent estimator of $\boldsymbol{\theta}_{1:\tau}$ nor $\boldsymbol{\theta}_{\tau:T}$ due to the mis-specification error. Therefore, one of the estimators from $\hat{\boldsymbol{\theta}}_{1:s}$ and $\hat{\boldsymbol{\theta}}_{s:T}$ such that $s < \tau$ is not a consistent estimator on the corresponding intervals. Formally, we have that

$$
\begin{aligned}
&\hat{PL}_{1:s} + \hat{PL}_{s:T} \\
&= \frac{1}{s}\left\|\mathbf{X}_{1:s}(u) - \sum_{v\in V\setminus\{u\}} \mathbf{X}_{1:s}(v)\hat{\boldsymbol{\theta}}_{1:s}(u,v)\right\|_2^2 + \lambda_{1:s}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{1:s}(u,v)\right\|_2 \\
&+ \frac{1}{T-s}\left\|\mathbf{X}_{s:T}(u) - \sum_{v\in V\setminus\{u\}} \mathbf{X}_{s:T}(v)\hat{\boldsymbol{\theta}}_{s:T}(u,v)\right\|_2^2 + \lambda_{s:T}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{s:T}(u,v)\right\|_2 \\
&= \frac{1}{s}\left\|\mathbf{X}_{1:s}(u) - \sum_{v\in V\setminus\{u\}} \mathbf{X}_{1:s}(v)\hat{\boldsymbol{\theta}}_{1:s}(u,v)\right\|_2^2 + \lambda_{1:s}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{1:s}(u,v)\right\|_2 \\
&+ \frac{1}{T-s}\left\|\mathbf{X}_{s:\tau}(u) - \sum_{v\in V\setminus\{u\}} \mathbf{X}_{s:\tau}(v)\hat{\boldsymbol{\theta}}_{s:T}(u,v)\right\|_2^2 + \frac{\delta\lambda_{s:T}}{T-s}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{s:T}(u,v)\right\|_2 \qquad \text{(A.13)} \\
&+ \frac{1}{T-s}\left\|\mathbf{X}_{\tau:T}(u) - \sum_{v\in V\setminus\{u\}} \mathbf{X}_{\tau:T}(v)\hat{\boldsymbol{\theta}}_{s:T}(u,v)\right\|_2^2 + \frac{(T-s-\delta)\lambda_{s:T}}{T-s}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{1:s}(u,v)\right\|_2 \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(A.14)}
\end{aligned}
$$

and

$$
\begin{aligned}
&\hat{PL}_{1:\hat{\tau}} + \hat{PL}_{\hat{\tau}:T} \\
&= \frac{1}{\tau}\left\|\mathbf{X}_{1:\hat{\tau}}(u) - \sum_{v\in V\setminus\{u\}} \mathbf{X}_{1:\hat{\tau}}(v)\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v)\right\|_2^2 + \lambda_{1:\hat{\tau}}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v)\right\|_2 \\
&+ \frac{1}{T-\tau}\left\|\mathbf{X}_{\hat{\tau}:T}(u) - \sum_{v\in V\setminus\{u\}} \mathbf{X}_{\hat{\tau}:T}(v)\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v)\right\|_2^2 + \lambda_{\hat{\tau}:T}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v)\right\|_2
\end{aligned}
$$

We write expression (A.13) as $\hat{PL}_{1:s} + \hat{PL}_{s:\hat{\tau}}$, and expression (A.14), as $\tilde{PL}_{s:T}$. We show (A.12) holds by first showing that $\hat{PL}_{1:s} + \hat{PL}_{s:\hat{\tau}} \geq \hat{PL}_{1:\hat{\tau}}$, and then

showing $\tilde{PL}_{s:T} \geq \hat{PL}_{\hat{\tau}:T}$. We first compute $\hat{PL}_{1:s} + \hat{PL}_{s:\hat{\tau}} - \hat{PL}_{1:\hat{\tau}}$:

$$
\begin{aligned}
&= \frac{1}{s} \left\| \mathbf{X}_{1:s}(u) - \sum_{v \in V \setminus \{u\}} v_{1:s}(v) \hat{\boldsymbol{\theta}}_{1:s}(u,v) \right\|_2^2 + \lambda_{1:s} \sum_{\mathbf{X} \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{1:s}(u,v) \right\|_2 \\
&+ \frac{1}{T-s} \left\| \mathbf{X}_{s:\hat{\tau}}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{s:\hat{\tau}}(v) \hat{\boldsymbol{\theta}}_{s:T}(u,v) \right\|_2^2 + \frac{\delta \lambda_{s:T}}{T-s} \sum_{v \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{s:T}(u,v) \right\|_2 \\
&- \frac{1}{\tau} \left\| \mathbf{X}_{1:\hat{\tau}}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{1:\hat{\tau}}(v) \hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v) \right\|_2^2 - \lambda_{1:\hat{\tau}} \sum_{v \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v) \right\|_2 .
\end{aligned}
$$

Assuming there is another group-lasso estimator defined on the the interval between s and $\hat{\tau}$, which is given by

$$
\begin{aligned}
&\hat{\boldsymbol{\theta}}_{s:\hat{\tau}} = \\
&\arg\min_{\boldsymbol{\theta}} \frac{1}{\hat{\tau}-s} \left\| \mathbf{X}_{s:\hat{\tau}}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{s:\hat{\tau}}(v) \boldsymbol{\theta}_{s:\hat{\tau}}(u,v) \right\|_2^2 + \lambda_{s:\hat{\tau}} \sum_{v \in V \setminus \{u\}} \left\| \boldsymbol{\theta}_{s:\hat{\tau}}(u,v) \right\|_2 .
\end{aligned}
$$

The estimator $\hat{\boldsymbol{\theta}}_{s:\hat{\tau}}$ is again a consistent estimator of $\boldsymbol{\theta}_{1:\hat{\tau}}$ and we have that:

$$
\begin{aligned}
&\frac{1}{\hat{\tau}-s} \left\| \mathbf{X}_{s:\hat{\tau}}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{s:\hat{\tau}}(v) \hat{\boldsymbol{\theta}}_{s:\hat{\tau}}(u,v) \right\|_2^2 + \lambda_{s:\hat{\tau}} \sum_{v \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{s:\hat{\tau}}(u,v) \right\|_2 \qquad \text{(A.15)} \\
&\leq \frac{1}{T-s} \left\| \mathbf{X}_{s:\hat{\tau}}(u) - \sum_{v \in V \setminus \{u\}} \mathbf{X}_{s:\hat{\tau}}(v) \hat{\boldsymbol{\theta}}_{s:T}(u,v) \right\|_2^2 + \frac{\delta \lambda_{s:T}}{T-s} \sum_{v \in V \setminus \{u\}} \left\| \hat{\boldsymbol{\theta}}_{s:T}(u,v) \right\|_2 \qquad \text{(A.16)}
\end{aligned}
$$

These are directly implied by Theorem (2) in [2] given that $\hat{\boldsymbol{\theta}}_{s:T}$ is not consistent in

the ℓ_2 sense of estimating $\boldsymbol{\theta}_{1:\hat{\tau}}$ whenever $s \neq \hat{\tau}$. Given (A.16), we have that

$$
\begin{aligned}
&\hat{PL}_{1:s} + \hat{PL}_{s:\hat{\tau}} - \hat{PL}_{1:\hat{\tau}} \\
&\geq \frac{1}{s}\left\|\mathbf{X}_{1:s}(u) - \sum_{v\in V\setminus\{u\}} \mathbf{X}_{1:s}(v)\hat{\boldsymbol{\theta}}_{1:s}(u,v)\right\|_2^2 + \lambda_{1:s}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{1:s}(u,v)\right\|_2 \\
&+ \frac{1}{\hat{\tau}-s}\left\|\mathbf{X}_{s:\hat{\tau}}(u) - \sum_{v\in V\setminus\{u\}} \mathbf{X}_{s:\hat{\tau}}(v)\hat{\boldsymbol{\theta}}_{s:\hat{\tau}}(u,v)\right\|_2^2 + \lambda_{s:\hat{\tau}}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{s:\hat{\tau}}(u,v)\right\|_2 \\
&- \frac{1}{\hat{\tau}}\left\|\mathbf{X}_{1:\hat{\tau}}(u) - \sum_{v\in V\setminus\{u\}} \mathbf{X}_{1:\hat{\tau}}(v)\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v)\right\|_2^2 - \lambda_{1:\hat{\tau}}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{1:\hat{\tau}}(u,v)\right\|_2
\end{aligned}
$$

The same argument in Part 1 holds here and we have

$$\mathbb{P}_{H_1}\left(\hat{PL}_{1:s} + \hat{PL}_{s:\hat{\tau}} \geq \hat{PL}_{1:\hat{\tau}}\right) \longrightarrow 1 \ .$$

Note that $\hat{\boldsymbol{\theta}}_{s:T}$ is not a consistent estimator of $\boldsymbol{\theta}_{\hat{\tau}:T}$ given the change point. Therefore, similar to A.16, we have

$$
\begin{aligned}
&\frac{1}{T-\hat{\tau}}\left\|\mathbf{X}_{\hat{\tau}:T}(u) - \sum_{v\in V\setminus\{u\}} \mathbf{X}_{\hat{\tau}:T}(v)\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v)\right\|_2^2 + \lambda_{\hat{\tau}:T}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{\hat{\tau}:T}(u,v)\right\|_2 \\
\leq\ &\frac{1}{T-s}\left\|\mathbf{X}_{\hat{\tau}:T}(u) - \sum_{v\in V\setminus\{u\}} \mathbf{X}_{\hat{\tau}:T}(v)\hat{\boldsymbol{\theta}}_{s:T}(u,v)\right\|_2^2 \\
+\ &\frac{(T-s-\delta)\lambda_{s:T}}{T-s}\sum_{v\in V\setminus\{u\}}\left\|\hat{\boldsymbol{\theta}}_{1:s}(u,v)\right\|_2
\end{aligned}
$$

and so

$$\mathbb{P}_{H_1}\left(\tilde{PL}_{s:T} \geq \hat{PL}_{\hat{\tau}:T}\right) \longrightarrow 1 \ .$$

Putting the two parts together, we have

$$\mathbb{P}_{H_1}\left(\hat{PL}_{1:s} + \hat{PL}_{s:T} \geq \hat{PL}_{1:\hat{\tau}} + \hat{PL}_{\hat{\tau}:T}\right) \longrightarrow 1$$

for any $s < \hat{\tau}$.

A.1.3 Proof of theorem 2.3.2

Under the assumption of stationarity, we could omit the time index in this section, that is $\boldsymbol{\theta} = \boldsymbol{\theta}_t$, $\forall t$. To show theorem 2.3.2, we begin with the following lemma.

Lemma A.1.1. *Given $\boldsymbol{\theta} \in \mathbb{R}^{(N-1)p}$, let $G(\boldsymbol{\theta}(u,v))$ be a p-dimensional vector with elements*

$$G(\boldsymbol{\theta}(u,v)) \ = -2T^{-1}\left(\mathbf{X}(v)'(\mathbf{X}(u) - \textstyle\sum_{v\in V\setminus\{u\}} \mathbf{X}(v)\boldsymbol{\theta}(u,v))\right). \qquad \text{(A.17)}$$

A vector $\hat{\boldsymbol{\theta}}$ with $\|\hat{\boldsymbol{\theta}}(u,v)\|_2 = 0$, $\forall\, v \in V\setminus\{u\}$ is a solution to the group lasso type of estimator iff for all $v \in V\setminus\{u\}$, $G(\hat{\boldsymbol{\theta}}(u,v)) + \lambda\mathbf{D}(\hat{\boldsymbol{\theta}}(u,v)) = \mathbf{0}$, where $\|\mathbf{D}(\hat{\boldsymbol{\theta}}(u,v))\|_2 = 1$ in the case of $\|\hat{\boldsymbol{\theta}}(u,v)\|_2 > 0$ and $\|\mathbf{D}(\hat{\boldsymbol{\theta}}(u,v)\|_2 < 1$ in the case of $\|\hat{\boldsymbol{\theta}}(u,v)\|_2 = 0$.

Proof Lemma A.1.1

Under KKT conditions, using subdifferential methods, the subdifferential of

$$\frac{1}{T}\left\|\mathbf{X}(u) - \sum_{v\in V\setminus\{u\}} \mathbf{X}(v)\boldsymbol{\theta}(u,v)\right\|^2 + \lambda \sum_{v\in V\setminus\{u\}} \|\boldsymbol{\theta}(u,v)\|_2$$

is given by $G(\boldsymbol{\theta}(u,v)) + \lambda\mathbf{D}(\hat{\boldsymbol{\theta}}(u,v))$, where $\|\mathbf{D}(\hat{\boldsymbol{\theta}}(u,v))\|_2 = 1$ if $\|\boldsymbol{\theta}(u,v)\|_2 > 0$ and $\|\mathbf{D}(\hat{\boldsymbol{\theta}}(u,v))\|_2 < 1$ if $\|\boldsymbol{\theta}(u,v)\|_2 = 0$. The lemma follows.

Proof Theorem 2.3.2

Assuming that $\hat{C}_u \nsubseteq C_u$, there must exist at least one estimated edge that joins two nodes in two different connectivity components. Given the assumptions, we use similar arguments as in the proof of Theorem 3 in [60]. Hence we have

$$\mathbb{P}(\exists\, u \in V : \hat{C}_u \nsubseteq C_u) \leq N \max_{u \in V} \mathbb{P}(\exists\, v \in V \backslash C_u : v \in \hat{\text{ne}}_u)\,,$$

where $\hat{\text{ne}}_u$ is the estimated neighborhood of node u and $v \in \hat{\text{ne}}_u$ means $\|\hat{\boldsymbol{\theta}}(u,v)\|_2 > 0$.

Let $\mathscr{E}$ be the event that

$$\max_{u \in V \backslash C_u} \left\| G\left(\hat{\boldsymbol{\theta}}(u,v)\right) \right\|_2^2 < \lambda^2.$$

Conditional on the event $\mathscr{E}$, $\hat{\boldsymbol{\theta}}$ is also a solution to the group lasso problem. As $\|\hat{\boldsymbol{\theta}}(u,v)\|_2 = 0$ for all $v \in V \backslash C_u$, it follows from lemma (A.1.1) that $\|\hat{\boldsymbol{\theta}}(u,v)\|_2 = 0$ for all $v \in V \backslash C_u$. Hence

$$\begin{aligned} \mathbb{P}(\exists\, v \in V \backslash C_u : \|\hat{\boldsymbol{\theta}}(u,v)\|_2 > 0) &\leq 1 - \mathbb{P}(\mathscr{E}) \\ &= P\left(\max_{v \in V \backslash C_u} \left\| G\left(\hat{\boldsymbol{\theta}}(u,v)\right) \right\|_2^2 \geq \lambda^2 \right). \end{aligned}$$

It is then sufficient to show that

$$N^2 \max_{u \in V,\ v \in V \backslash C_u} \mathbb{P}\left(\left\| G(\hat{\boldsymbol{\theta}}(u,v)) \right\|_2^2 \geq \lambda^2 \right) \leq \alpha.$$

Note that now the v and C_u are in different connected components, which means that $\mathbf{X}(v)$ is conditionally independent of $\mathbf{X}(C_u)$. Hence, conditional on all $\mathbf{X}(C_u)$, we have

$$\begin{aligned} \left\| G(\hat{\boldsymbol{\theta}}(u,v)) \right\|_2^2 &= \left\| -2T^{-1} \left(\mathbf{X}(v)'(\mathbf{X}(u) - \sum_{i \in C_u} \mathbf{X}(i)\hat{\boldsymbol{\theta}}(u,i)) \right) \right\|_2^2 \\ &= 4T^{-2} \left\| (\hat{\mathbf{R}}_1, \cdots, \hat{\mathbf{R}}_p)' \right\|_2^2 \end{aligned}$$

where $\hat{\mathbf{R}}_\ell = X_{-\ell}(v)' \left(\mathbf{X}(u) - \sum_{i \in C_u} \mathbf{X}(i) \hat{\boldsymbol{\theta}}(u,i) \right)$ is the remainder term and is independent of $\mathbf{X}(v)$, at all lags ℓ, for $\ell = 1, \cdots, p$. It follows that the joint distribution

$$(\hat{\mathbf{R}}_1, \cdots, \hat{\mathbf{R}}_p | \mathbf{X}(C_u)) \sim N(\mathbf{0}, \boldsymbol{\Omega})$$

for some covariance matrix $\boldsymbol{\Omega}$. Note that this is a conditional distribution given $\mathbf{X}(C_u)$. Hence, in the expression of $\boldsymbol{\Omega}$, every term appearing with a suffix u is constant and every term appearing with a suffix v is a normalized random variable. This simplifies the covariance term. Note that

$$\boldsymbol{\Omega}_{p \times p} = \mathbf{Cov} \left(\hat{\mathbf{R}}_1, \cdots, \hat{\mathbf{R}}_p \right)$$

and

$$\mathbf{tr}\left(\boldsymbol{\Omega}\right) = \sum_{\ell=1}^{p} \mathbf{Var}(\hat{\boldsymbol{R}}_\ell) = \sum_{\ell=1}^{p} \mathbf{Var} \left(\sum_{t=1}^{T} \left(X_t(u) - \sum_{i \in C_u} X_{t-\ell}(i) \hat{\theta}^{(\ell)}(u,i) \right) X_{t-\ell}(v) \right) =$$
$$\sum_{\ell=1}^{p} \sum_{s=1}^{T} \sum_{t=1}^{T} \mathbf{Cov} \left[\left(\left(X_t(u) - \sum_{i \in C_u} X_{t-\ell}(i) \hat{\theta}^{(\ell)}(u,i) \right) X_{t-\ell}(v) \right), \left(\left(X_s(u) - \sum_{i \in C_u} X_{s-\ell}(i) \hat{\theta}^{(\ell)}(u,i) \right) X_{s-\ell}(v) \right) \right] \tag{A.18}$$

Conditional on $\mathbf{X}(C_u)$, equation (A.18) can be further simplified as:

$$\mathbf{tr}\left(\boldsymbol{\Omega}\right)$$
$$= \sum_{\ell=1}^{p} \sum_{t=1}^{T} \sum_{s=1}^{T} \left(X_t(u) - \sum_{i \in C_u} X_{t-\ell}(i) \hat{\theta}^{(\ell)}(u,i) \right) \left(X_s(u) - \sum_{i \in C_u} X_{s-\ell}(i) \hat{\theta}^{(\ell)}(u,i) \right) \mathbf{Cov} \left[X_{t-\ell}(v), X_{s-\ell}(v) \right]$$
$$\leq \sum_{\ell=1}^{p} \sum_{t=1}^{T} \sum_{s=1}^{T} \left(X_t(u) - \sum_{i \in C_u} X_{t-\ell}(i) \hat{\theta}^{(\ell)}(u,i) \right) \left(X_s(u) - \sum_{i \in C_u} X_{s-\ell}(i) \hat{\theta}^{(\ell)}(u,i) \right) \sqrt{\mathbf{Var}(X_{t-\ell}(v)) \mathbf{Var}(X_{s-\ell}(v))}$$

We have the above bounded by

$$
\begin{aligned}
&\leq p \sum_{s=1}^{T} \sum_{t=1}^{T} \left(X_t(u) - \sum_{i \in C_u} X_{t-\ell}(i) \hat{\theta}^{(\ell)}(u,i) \right) \left(X_s(u) - \sum_{i \in C_u} X_{s-\ell}(i) \hat{\theta}^{(\ell)}(u,i) \right) \\
&= p \left[\sum_{t=1}^{T} \left(X_t(u) - \sum_{i \in C_u} X_{t-\ell}(i) \hat{\theta}^{(\ell)}(u,i) \right) \right]^2 \\
&\leq Tp \left[X_t(u) - \sum_{i \in C_u} X_{t-\ell}(i) \hat{\theta}^{(\ell)}(u,i) \right]^2 \\
&\leq Tp \|\mathbf{X}(u)\|_2^2
\end{aligned}
$$

The last inequality comes from the Cauchy-Schwarz inequality. Denote by ν_{max} the largest eigenvalue of the covariance matrix $\mathbf{\Omega}$. Since $\mathbf{\Omega}$ is PSD, we have $(\nu_{max}\mathbf{I} - \mathbf{\Omega})$ is also PSD. Following [62]'s argument, we can show $(\hat{\mathbf{R}}_1, \cdots, \hat{\mathbf{R}}_p) \leq_{cx} \mathbf{Y}$ for some random vector $\mathbf{Y} \sim N(\mathbf{0}, \nu_{max}\mathbf{I}_p)$, where $\leq_{cx}$ is the convex order that means $\mathbf{X} \leq \mathbf{Y}$, if and only if $\boldsymbol{\mu}_x = \boldsymbol{\mu}_y$ and $\sigma_x^2 \leq \sigma_y^2$. It follows that

$$
\begin{aligned}
\max_{u \in V, v \in V \setminus C_u} \mathbb{P}\left(\left\| G(\hat{\boldsymbol{\theta}}(u,v)) \right\|_2^2 \geq \lambda^2 \right) &\leq \max_{u \in V, v \in V \setminus C_u} \mathbb{P}(4T^{-2}(\mathbf{Y}'\mathbf{Y}) \geq \lambda^2) \\
&= \max_{u \in V, b \in V \setminus C_u} \mathbb{P}\left(\frac{1}{\nu_{max}} \mathbf{Y}'\mathbf{Y} \geq \frac{\lambda^2 T^2}{4\nu_{max}} \right) \quad .
\end{aligned}
$$

Note that the matrix $\frac{1}{\nu_{max}}\mathbf{Y}'\mathbf{Y}$ is idempotent and thus it follows a $\chi^2(p)$ distribution, and $\nu_{max} \leq \mathbf{tr}(\mathbf{\Omega}) \leq Tp\|\mathbf{X}(u)\|_2^2$. Put everything together, we have

$$
\begin{aligned}
& \max_{u \in V, b \in V \setminus C_u} \mathbb{P}\left(\|G(\hat{\boldsymbol{\theta}}(u,v))\|_2^2 \geq \lambda^2 \right) \\
\leq \quad & \max_{u \in V, v \in V \setminus C_u} \mathbb{P}\left(\chi^2(p) \geq \frac{\lambda^2 T^2}{4\nu_{max}} \right) \\
\leq \quad & \max_{u \in V, v \in V \setminus C_u} \mathbb{P}\left(\chi^2(p) \geq \frac{\lambda^2 T^2}{4Tp\|\mathbf{X}(u)\|_2^2} \right) \leq \frac{\alpha}{N(N-1)}
\end{aligned}
$$

and thus we have the desired $\lambda(\alpha, a)$

$$\lambda(\alpha) = 2\hat{\sigma}_u \sqrt{pQ\left(1 - \frac{\alpha}{N(N-1)}\right)}. \tag{A.19}$$

A.1.4 Proof of theorem 2.3.3

The proof of the theorem is in line with the work in [48]. The core idea is to bound the expected Hellinger loss in terms of the Kullback-Leibler distance. This approach, building on the original work of [55], leverages the union of unions bound, after discretizing the underlying parameter space. We assume a similar discretization here, while omitting the straightforward but tedious numerical analysis arguments that accompany. See, for example, [48] for details. Our fundamental bound is given by the following theorem.

Theorem A.1.2. *Let $\Gamma_T^{(N-1)p}$ be a space of finite collection of estimators $\tilde{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$, and pen$(\cdot)$ a function on Γ_T^p satisfying the condition*

$$\sum_{\tilde{\boldsymbol{\theta}}(u,v) \in \Gamma_T^p} e^{-pen(\tilde{\boldsymbol{\theta}}(u,v))} \leq 1, \tag{A.20}$$

Let $\hat{\boldsymbol{\theta}}$ be a penalized maximum likelihood estimator of the form

$$\hat{\boldsymbol{\theta}} \equiv \underset{\tilde{\boldsymbol{\theta}} \in \Gamma_T^{(N-1)p}}{\arg\min} \left\{ -\log p(\mathbf{X}(u)|\mathbf{X}(-u), \tilde{\boldsymbol{\theta}}) + 2 \sum_{v \in V \setminus \{u\}} Pen(\tilde{\boldsymbol{\theta}}(u,v)) \right\}.$$

Then

$$\mathbb{E}[H^2(p_{\hat{\boldsymbol{\theta}}}, p_{\boldsymbol{\theta}})] \leq \min_{\tilde{\boldsymbol{\theta}} \in \Gamma_T^{(N-1)p}} \left\{ K(p_{\boldsymbol{\theta}}, p_{\tilde{\boldsymbol{\theta}}}) + 2 \sum_{v \in V \setminus \{u\}} Pen(\tilde{\boldsymbol{\theta}}(u,v)) \right\}. \tag{A.21}$$

Note that the result of theorem A.1.2 requires that inequality (A.20) holds.

Lemma A.1.3 shows that our proposed penalty satisfies inequality (A.20). We now prove theorem A.1.2.

Proof Theorem A.1.2

Note that we have

$$\begin{aligned} H^2(p_{\hat{\boldsymbol{\theta}}}, p_{\boldsymbol{\theta}}) &= \int \left[\sqrt{p(\mathbf{x}|\mathbf{X}(-u), \hat{\boldsymbol{\theta}})} - \sqrt{p(\mathbf{x}|\mathbf{X}(-u), \boldsymbol{\theta})} \right]^2 d\nu(\mathbf{x}) \\ &= 2\left(1 - \int \sqrt{p(\mathbf{x}|\mathbf{X}(-u), \hat{\boldsymbol{\theta}}) p(\mathbf{x}|\mathbf{X}(-u), \boldsymbol{\theta})} d\nu(\mathbf{x})\right) \\ &\leq -2 \log \int \sqrt{p(\mathbf{x}|\mathbf{X}(-u), \hat{\boldsymbol{\theta}}) p(\mathbf{x}|\mathbf{X}(-u), \boldsymbol{\theta})} d\nu(\mathbf{x}), \end{aligned}$$

Taking the conditional expectation respect to $\mathbf{X}(u)|\mathbf{X}(-u)$, we then have

$$\begin{aligned} & \mathbb{E}[H^2(p_{\hat{\boldsymbol{\theta}}}, p_{\boldsymbol{\theta}})] \\ \leq \quad & 2\mathbb{E} \log \left(\frac{1}{\int \sqrt{p(\mathbf{x}|\mathbf{X}(-u), \hat{\boldsymbol{\theta}}) p(\mathbf{x}|\mathbf{X}(-u), \boldsymbol{\theta})} d\nu(\mathbf{x})} \right) \\ \leq \quad & 2\mathbb{E} \log \left(\frac{p^{1/2}(\mathbf{X}(u)|\mathbf{X}(-u), \hat{\boldsymbol{\theta}}) e^{-\sum_v pen(\hat{\boldsymbol{\theta}}(u,v))}}{p^{1/2}(\mathbf{X}(u)|\mathbf{X}(-u), \check{\boldsymbol{\theta}}) e^{-\sum_v pen(\check{\boldsymbol{\theta}}(u,v))}} \frac{1}{\int \sqrt{p(\mathbf{x}|\mathbf{X}(-u), \hat{\boldsymbol{\theta}}) p(\mathbf{x}|\mathbf{X}(-u), \boldsymbol{\theta})} d\nu(\mathbf{x})} \right), \end{aligned}$$

where the collection of $\check{\boldsymbol{\theta}}(u, v)$'s are the arguments that minimize the right-hand side of the expression (A.21). The last expression can be written in two pieces, that is

$$\mathbb{E}\left[\log \frac{p(\mathbf{X}(u)|\mathbf{X}(-u), \boldsymbol{\theta})}{p(\mathbf{X}(u)|\mathbf{X}(-u), \check{\boldsymbol{\theta}})} \right] + 2 \sum_v pen(\check{\boldsymbol{\theta}}(u, v)) \tag{A.22}$$

$$+ 2\mathbb{E} \log \left(\frac{p^{1/2}(\mathbf{X}(u)|\mathbf{X}(-u), \hat{\boldsymbol{\theta}})}{p^{1/2}(\mathbf{X}(u)|\mathbf{X}(-u), \boldsymbol{\theta})} \frac{\prod_v \prod_\ell e^{-pen(\hat{\boldsymbol{\theta}}^{(\ell)}(u,v))}}{\int \sqrt{p(\mathbf{x}|\mathbf{X}(-u), \hat{\boldsymbol{\theta}}) p(\mathbf{x}|\mathbf{X}(-u), \boldsymbol{\theta})} d\nu(\mathbf{x})} \right) \tag{A.23}$$

Note that the expression (A.22) is the right hand side of (A.21). What we need to show then is that expression (A.23) is bounded above by zero. By applying

Jensen's inequality, we have (A.23) bounded by:

$$2\log\mathbb{E}\left[\prod_{v} e^{-pen(\hat{\boldsymbol{\theta}}(u,v))}\frac{\sqrt{p(\mathbf{X}(u)|\mathbf{X}(-u),\hat{\boldsymbol{\theta}})/p(\mathbf{X}(u)|\mathbf{X}(-u),\boldsymbol{\theta})}}{\int\sqrt{p(\mathbf{x}|\mathbf{X}(-u),\hat{\boldsymbol{\theta}})p(\mathbf{x}|\mathbf{X}(-u),\boldsymbol{\theta})}d\nu(\mathbf{x})}\right] \tag{A.24}$$

The integrand in the expectation in (A.24) can be bounded by

$$\sum_{\tilde{\boldsymbol{\theta}}\in\Gamma_T^{(N-1)p}}\prod_{v} e^{-pen(\tilde{\boldsymbol{\theta}}(u,v))}\frac{\sqrt{p(\mathbf{X}(u)|\mathbf{X}(-u),\tilde{\boldsymbol{\theta}})/p(\mathbf{X}(u)|\mathbf{X}(-u),\boldsymbol{\theta})}}{\int\sqrt{p(\mathbf{x}|\mathbf{X}(-u),\tilde{\boldsymbol{\theta}})p(\mathbf{x}|\mathbf{X}(-u),\boldsymbol{\theta})}d\nu(\mathbf{x})}.$$

Given the fact that $\tilde{\boldsymbol{\theta}}$ does not depend on the $\mathbf{X}(-u)$, (A.24) can be bounded by

$$\begin{aligned}&2\log\sum_{\tilde{\boldsymbol{\theta}}\in\Gamma_T^{(N-1)p}}\prod_{v} e^{-pen(\tilde{\boldsymbol{\theta}}(u,v))}\frac{\mathbb{E}\left[\sqrt{p(\mathbf{X}(u)|\mathbf{X}(-u),\tilde{\boldsymbol{\theta}})/p(\mathbf{X}(u)|\mathbf{X}(-u),\boldsymbol{\theta})}\right]}{\int\sqrt{p(\mathbf{x}|\mathbf{X}(-u),\tilde{\boldsymbol{\theta}})p(\mathbf{x}|\mathbf{X}(-u),\boldsymbol{\theta})}d\nu(\mathbf{x})}\\ =&2\log\sum_{\tilde{\boldsymbol{\theta}}\in\Gamma_T^{(N-1)p}}\prod_{v} e^{-pen(\tilde{\boldsymbol{\theta}}(u,v))}\end{aligned} \tag{A.25}$$

Since $e^{-pen(\tilde{\boldsymbol{\theta}}(u,v))}>0$ for any $\tilde{\boldsymbol{\theta}}(u,v)$, and using the inequality $\sum_i a_i b_i \le \sum_i a_i \sum_i b_i$ for any $a_i>0, b_i>0$, we can bound (A.25) by:

$$2\log\prod_{v}\sum_{\tilde{\boldsymbol{\theta}}(u,v)\in\Gamma_T^{p}} e^{-pen(\tilde{\boldsymbol{\theta}}(u,v))}$$

From the condition in (A.20), we see that the above expression is bounded by zero. We now show that our proposed estimator satisfies condition (A.20) by the following lemma.

Lemma A.1.3. *Let Γ_T be the collection of all $\tilde{\boldsymbol{\theta}}^{(\ell)}(u,v)$ with components $\tilde{\boldsymbol{\theta}}_t^{(\ell)}(u,v)\in D_T[-C,C]$ and possessed of a Haar like expansion through a common partition, using either RDP (see expression (2.2)) or RP (see expression (2.4)), where $D_T[-C,C]$ denotes a discretization of the interval $[-C,C]$ into $T^{1/2}$ equispaced*

values. For any type of penalty such that

$$Pen(\tilde{\boldsymbol{\theta}}(u,v)) = C_3 \log T \#\{\mathcal{P}(\tilde{\boldsymbol{\theta}})\} + \lambda \sum_{\mathcal{I}\in\mathcal{P}(\tilde{\boldsymbol{\theta}})} \|\tilde{\boldsymbol{\theta}}_{\mathcal{I}}(u,v)\|_2,$$

where $C_3 = 1/2$ *for recursive dyadic partitioning and* $C_3 = 3/2$ *for recursive partitioning, we have*

$$\sum_{\tilde{\boldsymbol{\theta}}(u,v)\in\Gamma_T^p} e^{-pen(\tilde{\boldsymbol{\theta}}(u,v))} \leq 1,$$

for $T > \lceil e^{2p/3} \rceil$.

Proof Lemma A.1.3

We prove Lemma A.1.3 for the case of recursive partitioning. We write $\Gamma_T = \bigcup_{d_\ell=1}^{T} \Gamma_T^{(d_\ell)}$ where $\Gamma_T^{(d_\ell)}$ is the subset of values $\tilde{\boldsymbol{\theta}}_t^{(\ell)}(u,v)$ that is composed of d_ℓ constant valued sequences. For example, $\Gamma_T^{(d_\ell)}$ consists of all length T sequences such that there are exactly d_ℓ alternating sequences of zero and nonzero elements. So, for example, $(0,0,4,0,0)$ and $(2,0,1,1,1)$ might be two such sequences in $\Gamma_5^{(3)}$. Then we have

$$
\begin{aligned}
\sum_{\tilde{\boldsymbol{\theta}}(u,v)\in\Gamma_T^p} e^{-pen(\tilde{\boldsymbol{\theta}}(u,v))} &= \sum_{\tilde{\boldsymbol{\theta}}(u,v)\in\Gamma_T^p} e^{-(3/2)\log T\{\#\mathcal{P}(\tilde{\boldsymbol{\theta}})\}-\lambda\sum_{\mathcal{I}\in\mathcal{P}(\tilde{\boldsymbol{\theta}})}\|\tilde{\boldsymbol{\theta}}_{\mathcal{I}}(u,v)\|_2} \\
&\leq \sum_{\tilde{\boldsymbol{\theta}}(u,v)\in\Gamma_T^p} e^{-(3/2)\log T\{\#\mathcal{P}(\tilde{\boldsymbol{\theta}})\}} \\
&\leq \prod_{\ell=1}^{p} \sum_{\tilde{\boldsymbol{\theta}}^{(\ell)}(u,v)\in\Gamma_T} e^{-(3/2p)\log T\{\#\mathcal{P}(\tilde{\boldsymbol{\theta}})\}} \\
&= \prod_{\ell=1}^{p} \sum_{d_\ell=1}^{T} \binom{T-1}{d_\ell-1} e^{-d_\ell(3/2p)\log T} \\
&= \prod_{\ell=1}^{p} \sum_{d^{\ell'}=0}^{T-1} \binom{T-1}{d^{\ell'}} e^{-(d^{\ell'}+1)(3/2p)\log T} \\
&= \prod_{\ell=1}^{p} \sum_{d\prime=0}^{T-1} \frac{(T-1)!}{d^{\ell'}!(T-d^{\ell'}-1)!} T^{-(d^{\ell'}+1)(3/2p)} \\
&\leq \prod_{\ell=1}^{p} T^{-(3/2p)} \sum_{d^{\ell'}=0}^{T-1} \frac{(T-1)^{d^{\ell'}}}{d^{\ell'}!} \frac{1}{T^{(3/2p)d^{\ell'}}} \\
&\leq T^{-(3/2)} e^{p}
\end{aligned}
$$

which is bounded by 1 for any $T > \lceil e^{2p/3} \rceil$. The argument follows analogously for the case of recursive dyadic partitioning.

Using the loss function and the corresponding risk function we defined before, recovering the neighborhood of node u is essentially a univariate Gaussian time series problem, and thus the KL divergence of the conditional likelihood function takes the form:

$$
K(p_{\boldsymbol{\theta}}, p_{\tilde{\boldsymbol{\theta}}}) = \mathbb{E}\left\{\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x})}{p_{\tilde{\boldsymbol{\theta}}}(\mathbf{x})}\right\} = \mathbb{E}\left\{\sum_{t=1}^{T} \log \frac{p_{\boldsymbol{\theta}}(X_t(u))}{p_{\tilde{\boldsymbol{\theta}}}(X_t(u))}\right\} = \sum_{t=1}^{T} (\tilde{\mu}_t - \mu_t)^2/(2\sigma^2)
$$

where each μ_t is the mean of $X_t(u)$, and $\tilde{\mu}_t$ is an approximation/estimate thereof, for a given estimator $\tilde{\boldsymbol{\theta}}$. Since these means in turn are based on linear combina-

tions of all neighborhood observations, over p lags, we have:

$$\tilde{\mu}_t - \mu_t = \sum_{v \in V \setminus \{u\}} \sum_{\ell=1}^{p} X_{t-\ell}(v)[\tilde{\theta}_t^{(\ell)}(u,v) - \theta_t^{(\ell)}(u,v)]$$

So the KL divergence for each neighborhood problem involves values at other nodes.

Assume without loss of generality that $\sigma \equiv 1$. From (A.21) and the fact that the K-L divergence in the Gaussian case is simply proportional to a squared ℓ_2-norm, the risk of estimating $\boldsymbol{\theta}$ by $\hat{\boldsymbol{\theta}}$ should be in the form:

$$\begin{aligned}
& \mathbb{R}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) \\
\leq \quad & \min_{\tilde{\boldsymbol{\theta}} \in \Gamma_T^{(N-1)p}} \left\{ \frac{1}{T} K(p_{\boldsymbol{\theta}}, p_{\tilde{\boldsymbol{\theta}}}) + \frac{2}{T} \sum_{v=1}^{N-1} Pen(\tilde{\boldsymbol{\theta}}(u,v)) \right\} \\
\leq \quad & \min_{\tilde{\boldsymbol{\theta}} \in \Gamma_T^{(N-1)p}} \left\{ \frac{1}{2T} \|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2 + \frac{\lambda}{T} \sum_{\mathcal{I} \in \mathcal{P}(\tilde{\boldsymbol{\theta}})} \sum_{v=1}^{N-1} \|\tilde{\boldsymbol{\theta}}_{\mathcal{I}}(u,v)\|_2 + \frac{2}{T} \sum_{v=1}^{N-1} (3/2) \log T \#\{\mathcal{P}(\tilde{\boldsymbol{\theta}})\} \right\}
\end{aligned}$$

From Cauchy-Schwarz, we have that

$$\begin{aligned}
\mathbb{R}(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}) \leq & \min_{\tilde{\boldsymbol{\theta}} \in \Gamma_T^{(N-1)p}} \left\{ \frac{1}{2T} \|\mathbf{X}(-u)'\mathbf{X}(-u)\|_2 \sum_{t=1}^{T} \sum_{v=1}^{N-1} \sum_{\ell=1}^{p} \left(\tilde{\theta}_t^{(\ell)}(u,v) - \theta_t^{(\ell)}(u,v) \right)^2 \right. \\
& \left. + \frac{\lambda}{T} \sum_{\mathcal{I} \in \mathcal{P}(\tilde{\boldsymbol{\theta}})} \sum_{v=1}^{N-1} \|\tilde{\boldsymbol{\theta}}_{\mathcal{I}}(u,v)\|_2 + 3(N-1) \frac{\log T}{T} \#\{\mathcal{P}(\tilde{\boldsymbol{\theta}})\} \right\} \\
\leq & \min_{\tilde{\boldsymbol{\theta}} \in \Gamma_T^{(N-1)p}} \left\{ \frac{1}{2} \Lambda \sum_{v=1}^{N-1} \sum_{\ell=1}^{p} \left\| \tilde{\boldsymbol{\theta}}_t^{(\ell)}(u,v) - \boldsymbol{\theta}_t^{(\ell)}(u,v) \right\|_2^2 \right. \\
& \left. + \frac{\lambda}{T} \sum_{\mathcal{I} \in \mathcal{P}(\tilde{\boldsymbol{\theta}})} \sum_{v=1}^{N-1} \|\tilde{\boldsymbol{\theta}}_{\mathcal{I}}(u,v)\|_2 + 3(N-1) \frac{\log T}{T} \#\{\mathcal{P}(\tilde{\boldsymbol{\theta}})\} \right\} . \qquad \text{(A.26)}
\end{aligned}$$

The minimization of the expression (A.26) tries to find the optimal balancing of bias and variance. To bound it, the following L_2 result from [23] plays the core role.

Lemma A.1.4. *Let $\theta_{(\cdot)}^{(\ell)}(u,v) \in BV(C)$. Define ${\theta_{bd}}_{(\cdot)}^{(\ell)}(u,v)$ to be the best d-term approximant to $\theta_{(\cdot)}^{(\ell)}(u,v)$ in the dyadic Haar basis for $L_2([0,1])$. Then $\|{\theta_{bd}}^{(\ell)}(u,v) - \theta^{(\ell)}(u,v)\|_{L_2} = \mathcal{O}(d^{-1})$.*

Define ${\boldsymbol{\theta}_{bd}}^{(\ell)}(u,v)$ to be the average sampling of ${\theta_{bd}}^{(\ell)}(u,v)$ on the interval I_i, that is ${\boldsymbol{\theta}_{bd}}^{(\ell)}(u,v) = T\int_{I_i} {\theta_{bd}}^{(\ell)}(u,v)(t)dt$. Then let ${\tilde{\boldsymbol{\theta}}_{bd}}^{(\ell)}(u,v)$ be the result of discretizing the elements of ${\boldsymbol{\theta}_{bd}}^{(\ell)}(u,v)$ to the set $D_T[-C,C]$, where C is the radius of the bounded variation ball defined in Assumption 6. We have the following by triangle inequality:

$$\left\|\tilde{\boldsymbol{\theta}}^{(\ell)}(u,v) - \boldsymbol{\theta}^{(\ell)}(u,v)\right\|_{\ell_2}^2 \leq \left\|{\boldsymbol{\theta}_{bd}}^{(\ell)}(u,v) - \boldsymbol{\theta}^{(\ell)}(u,v)\right\|_{\ell_2}^2 + \left\|\tilde{\boldsymbol{\theta}}^{(\ell)}(u,v) - {\boldsymbol{\theta}_{bd}}^{(\ell)}(u,v)\right\|_{\ell_2}^2 + 2\left\|{\boldsymbol{\theta}_{bd}}^{(\ell)}(u,v) - \boldsymbol{\theta}^{(\ell)}(u,v)\right\|_{\ell_2}\left\|\tilde{\boldsymbol{\theta}}^{(\ell)}(u,v) - \boldsymbol{\theta}^{(\ell)}(u,v)\right\|_{\ell_2}. \tag{A.27}$$

For sequence ${\boldsymbol{\theta}_{bd}}^{(\ell)}(u,v)$ and ${\tilde{\boldsymbol{\theta}}_{bd}}^{(\ell)}(u,v)$ obtained from average sampling, a simple argument relating Haar function on the discrete set $D_T[-C,C]$ to the functions on the interval $[0,1]$ is to show that

$$\frac{1}{T}\left\|{\tilde{\boldsymbol{\theta}}_{bd}}^{(\ell)}(u,v) - \boldsymbol{\theta}^{(\ell)}(u,v)\right\|_{\ell_2}^2 \leq \left\|\theta_{bd}^{(\ell)}(u,v) - \theta^{(\ell)}(u,v)\right\|_{L2}^2 .$$

See equation (27) of [48]. On the right hand side of (A.27), the first resulting squared term will be of order $\mathcal{O}(Td^{-2})$. The second term is a discretization error and by lemma (A.1.4) is of order $\mathcal{O}(1)$. The third cross-term is therefore of order $\mathcal{O}(T^{1/2}d^{-1})$.

Given these results, we have the following bound of equation (A.26) by bounding the bias term over each $\Gamma_T^{(d)}$, where $d = \bigcup_i d_i$, for each d_i and $i =$

$1, \cdots, (N-1)p$. We then we optimize for d:

$$\min_{\tilde{\boldsymbol{\theta}} \in \Gamma_T^{(N-1)p(d)}} \left\{ \frac{1}{2}\Lambda \sum_{v=1}^{N-1} \sum_{\ell=1}^{p} \left\| \tilde{\boldsymbol{\theta}}^{(\ell)}(u,v) - \boldsymbol{\theta}^{(\ell)}(u,v) \right\|_2^2 + \frac{\lambda}{T} \sum_{\mathcal{I} \in \mathcal{P}(\tilde{\boldsymbol{\theta}})} \sum_{v=1}^{N-1} \|\tilde{\boldsymbol{\theta}}_{\mathcal{I}}(u,v)\|_2 + 3(N-1)\frac{\log T}{T} \#\{\mathcal{P}(\tilde{\boldsymbol{\theta}})\} \right\} \tag{A.28}$$

The first term is dominated by the first part of expression (A.27) and is of order $\mathcal{O}(\Lambda T d^{-2})$. In the second term, we have $\frac{\lambda}{T} \sum_{\mathcal{I} \in \mathcal{P}(\tilde{\boldsymbol{\theta}})} \sum_{v=1}^{N-1} \|\tilde{\boldsymbol{\theta}}_{\mathcal{I}}(u,v)\|_2$, which are the group lasso terms. Given the fact that $\theta_{(\cdot)}^{(\ell)}(u,v)$ is of $BV(C)$, we have that $1/(T^{1/2})\|\tilde{\boldsymbol{\theta}}_{\mathcal{I}}(u,v)\|_2$ is of order $\mathcal{O}(C + d^{-1})$. Note that λ is of order $T^{-1/2}$ and the number of interval $\#\{\mathcal{P}(\tilde{\boldsymbol{\theta}})\}$ is proportional to d. So the second term is of order $\mathcal{O}(T^{-1} * d * (C + d^{-1}))$. The third term is of order $\mathcal{O}(dT^{-1} \log T)$. Combining the above results, we have that:

$$\begin{aligned} &\min_{\tilde{\boldsymbol{\theta}} \in \Gamma_T^{(N-1)p(d)}} \left\{ \frac{1}{2}\Lambda \sum_{v=1}^{N-1} \sum_{\ell=1}^{p} \left\| \tilde{\boldsymbol{\theta}}^{(\ell)}(u,v) - \boldsymbol{\theta}^{(\ell)}(u,v) \right\|_2^2 + \frac{\lambda}{T} \sum_{\mathcal{I} \in \mathcal{P}(\tilde{\boldsymbol{\theta}})} \sum_{v=1}^{N-1} \|\tilde{\boldsymbol{\theta}}_{\mathcal{I}}(u,v)\|_2 + 3(N-1)\frac{\log T}{T} \#\{\mathcal{P}(\tilde{\boldsymbol{\theta}})\} \right\} \\ &\leq \mathcal{O}(\Lambda T d^{-2}) + \mathcal{O}(T^{-1} * d * (C + d^{-1})) + \mathcal{O}(dT^{-1} \log T), \end{aligned}$$

which is minimized for $d \sim (\Lambda T^2 / \log T)^{1/3}$. Substitution then yields the result that the risk is bounded by a quantity of order $\mathcal{O}((\Lambda \log^2 T / T)^{1/3})$. For estimation via recursive dyadic partitioning, where $\#\{\mathcal{P}(\tilde{\theta})\}$ is proportional to $d \log T$, the expression is minimized at $d \sim (\Lambda T^2 / \log^2 T)^{1/3}$, which gives the bound of the risk of order $\mathcal{O}(\Lambda \log^4 T / T)^{1/3}$.

A.2 Multiscale network analysis through tail-gredy bottom-up approximation, with applications in neuroscience

A.2.1 Proof of Theorem 3.3.1

Proof Theorem 3.3.1

The fact that $\mathbb{P}(\mathcal{A}) \to 1$ as $N \to \infty$ follows from Lemma 1 of [81]. We begin by defining two sets S_0^ℓ and S_1^ℓ with $S_1^\ell = \left\{1 \le r \le k(\ell) : \text{the support of } \psi_r^\ell \text{ crosses multiple regions of constancy at level } \ell\right\}$ and $S_0^\ell = \{1, \cdots, k(\ell)\} \backslash S_1^\ell$.

$$
\begin{aligned}
R(\hat{f}, f) &= \frac{1}{N} \sum_v (\hat{f}(v) - f(v))^2 \\
&= \frac{1}{N} \sum_{\ell=1}^{L} \sum_{r=1}^{k(\ell)} \left(\alpha_r^\ell \mathbb{I}\left\{ \exists V_{r'}^{\ell'} \subseteq V_r^\ell \,\middle|\, |\alpha_{r'}^{\ell'}| > \lambda(\ell', r') \right\} - \mu_r^\ell \right)^2 \\
&+ \frac{1}{N} (\alpha_0^0 - \mu_0^0)^2 \\
&= \frac{1}{N} \sum_{\ell=1}^{L} \left(\sum_{r \in S_0^\ell} + \sum_{r \in S_1^\ell} \right) \left(\alpha_r^\ell \mathbb{I}\left\{ \exists V_{r'}^{\ell'} \subseteq V_r^\ell \,\middle|\, |\alpha_{r'}^{\ell'}| > \lambda(\ell', r') \right\} - \mu_r^\ell \right)^2 \\
&+ \frac{1}{N} (\alpha_0^0 - \mu_0^0)^2 \\
&\le \frac{1}{N} \sum_{\ell=1}^{L} \left(\sum_{r \in S_0^\ell} + \sum_{r \in S_1^\ell} \right) \left(\alpha_r^\ell \mathbb{I}\left\{ \exists V_{r'}^{\ell'} \subseteq V_r^\ell \,\middle|\, |\alpha_{r'}^{\ell'}| > \lambda(\ell', r') \right\} - \mu_r^\ell \right)^2 \\
&+ \frac{2}{N} \log N
\end{aligned}
$$

By Lemma A.2.1, we have that on the set S_0^ℓ, $|\alpha_r^\ell| \le \sqrt{2 \log N} \left\{ \frac{\sqrt{|V_m^{\ell'-1}|} + \sqrt{|V_{m'}^{\ell'-1}|}}{\sqrt{|V_m^{\ell'-1}| + |V_{m'}^{\ell'-1}|}} \right\}$. We then have

$$R(\hat{f},f) \leq \frac{1}{N}\sum_{\ell=1}^{L}\sum_{r\in S_1^\ell}\left(\alpha_r^\ell \mathbb{I}\left\{\exists V_{r'}^{\ell'} \subseteq V_r^\ell \,\middle|\quad |\alpha_{r'}^{\ell'}| > \lambda(\ell',r')\right\} - \mu_r^\ell\right)^2 + \frac{2}{N}\log N \ .$$

Denote by $\mathcal{E}$ the event $\left\{\exists V_{r'}^{\ell'} \subseteq V_r^\ell \,\middle|\quad |\alpha_{r'}^{\ell'}| > \sqrt{2\log N}\left\{\frac{\sqrt{|V_m^{\ell'-1}|}+\sqrt{|V_{m'}^{\ell'-1}|}}{\sqrt{|V_r^\ell|}}\right\}\right\}$.

We compute

$$\begin{aligned}
&(\alpha_r^\ell \mathbb{I}(\mathcal{E}) - \mu_r^\ell)^2 = (\alpha_r^\ell \mathbb{I}(\mathcal{E}) - \alpha_r^\ell + \alpha_r^\ell - \mu_r^\ell)^2 \\
&\leq (\alpha_r^\ell)^2 \mathbb{I}(\neg\mathcal{E}) + (\alpha_r^\ell - \mu_r^\ell)^2 + 2|\alpha_r^\ell \mathbb{I}(\neg\mathcal{E})||\alpha_r^\ell - \mu_r^\ell| \\
&\leq \lambda^2 + 2\lambda\sqrt{2\log N} + 2\log N \\
&\leq (6+4\sqrt{2})\log N \ .
\end{aligned}$$

Note that the level L associated with the TGUH transformation is bounded, i.e., $L \leq \log N/\log(1-\rho)^{-1}$. Combining this with the fact that $|S_\ell^1| \leq K$, and the assumption that $K = o\left(N/\log^2 N\right)$, we have that $R(\hat{f},f)$ is of order $\mathcal{O}\left(\frac{K\log^2 N}{N\log(1-\rho)^{-1}}\right)$.

Lemma A.2.1. *Let $S_0^\ell = \{1 \leq m \leq k(\ell) : \mu_r^\ell = 0\}$. On $\mathcal{A}$, for $\ell = 1,\cdots,L$, $k \in S_0^\ell$, we have*

$$|\alpha_r^\ell| \leq \sqrt{2\log N}\left\{\frac{\sqrt{|V_m^{\ell-1}|}+\sqrt{|V_{m'}^{\ell-1}|}}{\sqrt{|V_m^\ell|}}\right\} \ .$$

Proof Lemma A.2.1

Denote the two sub-regions which merge into V_r^ℓ in the next level as $V_m^{\ell-1}$ and $V_{m'}^{\ell-1}$. On $\mathcal{A}$, for $\ell = 1,\cdots,L$, $k \in S_0^\ell$, we have

$$
\begin{aligned}
\left|\alpha_r^\ell\right| &= \left| \left\{ \frac{|V_{m'}^{\ell-1}|}{|V_r^\ell|} \right\}^{1/2} \frac{\sum_{v \in V_m^{\ell-1}} \epsilon(v)}{\sqrt{|V_m^{\ell-1}|}} - \left\{ \frac{|V_m^{\ell-1}|}{|V_r^\ell|} \right\}^{1/2} \frac{\sum_{v \in V_{m'}^{\ell-1}} \epsilon(v)}{\sqrt{|V_{m'}^{\ell-1}|}} \right| \\
&\leq \sqrt{2 \log N} \left(\left\{ \frac{|V_{m'}^{\ell-1}|}{|V_r^\ell|} \right\}^{1/2} + \left\{ \frac{|V_m^{\ell-1}|}{|V_r^\ell|} \right\}^{1/2} \right) \\
&= \sqrt{2 \log N} \left\{ \frac{|V_{m'}^{\ell-1}|^{1/2} + |V_m^{\ell-1}|^{1/2}}{|V_r^\ell|^{1/2}} \right\}.
\end{aligned}
$$

References

[1] Kaoru Amano, Tsunehiro Takeda, Tomoki Haji, Masahiko Terao, Kazushi Maruya, Kenji Matsumoto, Ikuya Murakami, and Shin'ya Nishida. Human neural responses involved in spatial pooling of locally ambiguous motion signals. *Journal of neurophysiology*, 107(12):3493–3508, 2012.

[2] Francis R Bach. Consistency of the group lasso and multiple kernel learning. *The Journal of Machine Learning Research*, 9:1179–1225, 2008.

[3] Prakash Balachandran, Edoardo Airoldi, and Eric Kolaczyk. Inference of network summary statistics through network denoising. *arXiv preprint arXiv:1310.0423*, 2013.

[4] Matteo Barigozzi and Christian T Brownlees. Nets: network estimation for time series. *Available at SSRN 2249909*, 2014.

[5] Sumanta Basu, Ali Shojaie, and George Michailidis. Network granger causality with inherent grouping structure. *Journal of Machine Learning Research*, 16:417–453, 2015.

[6] Timothy EJ Behrens, H Johansen Berg, Saad Jbabdi, Matthew FS Rushworth, and Mark W Woolrich. Probabilistic diffusion tractography with multiple fibre orientations: What can we gain? *Neuroimage*, 34(1):144–155, 2007.

[7] Brenda Betancourt, Abel Rodríguez, and Naomi Boyd. Bayesian fused lasso regression for dynamic binary networks. *Journal of Computational and Graphical Statistics*, (just-accepted), 2017.

[8] Katherine C Bettencourt and Yaoda Xu. Decoding the content of visual

short-term memory under distraction in occipital and parietal areas. *Nature neuroscience*, 19(1):150–157, 2016.

[9] Sharmodeep Bhattacharyya, Peter J Bickel, et al. Subsampling bootstrap of count features of networks. *The Annals of Statistics*, 43(6):2384–2411, 2015.

[10] Andrew Bolstad, Barry D Van Veen, and Robert Nowak. Causal network inference via group sparse regularization. *Signal Processing, IEEE Transactions on*, 59(6):2628–2641, 2011.

[11] OJ Braddick, JMD O?Brien, J Wattam-Bell, J Atkinson, and Robert Turner. Form and motion coherence activate independent, but not dorsal/ventral segregated, networks in the human brain. *Current Biology*, 10(12):731–734, 2000.

[12] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.

[13] Carter T Butts. Network inference, error, and informant (in) accuracy: a bayesian approach. *social networks*, 25(2):103–140, 2003.

[14] Gyorgy Buzsaki. *Rhythms of the Brain*. Oxford University Press, 2006.

[15] FJ Calabro and LM Vaina. Interaction of cortical networks mediating object motion detection by moving observers. *Experimental brain research*, 221(2):177–189, 2012.

[16] Sourav Chatterjee et al. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2015.

[17] Siheng Chen, Rohan Varma, Aarti Singh, and Jelena Kovačević. Representations of piecewise smooth signals on graphs. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*,

pages 6370–6374. IEEE, 2016.

[18] Catherine Jean Chu, Naoaki Tanaka, J Diaz, Brian L Edlow, Ona Wu, M Hämäläinen, S Stufflebeam, Sydney S Cash, and Mark A Kramer. Eeg functional connectivity is partially predicted by underlying white matter connectivity. *Neuroimage*, 108:23–33, 2015.

[19] Ronald R Coifman and Mauro Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53–94, 2006.

[20] RR Coifman and WE Leeb. Earth movers distance and equivalent metrics for spaces with hierarchical partition trees. 2013.

[21] Mark Crovella and Eric Kolaczyk. Graph wavelets for spatial traffic analysis. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1848–1857. IEEE, 2003.

[22] Richard A Davis, Thomas Lee, and Gabriel A Rodriguez-Yam. Break detection for a class of nonlinear time series models. *Journal of Time Series Analysis*, 29(5):834–867, 2008.

[23] David L Donoho. Unconditional bases are optimal bases for data compression and for statistical estimation. *Applied and computational harmonic analysis*, 1(1):100–115, 1993.

[24] David L Donoho. Cart and best-ortho-basis: a connection. *Ann. Statist.*, 25(5):1870–1911, 1997.

[25] Dean Eckles, Brian Karrer, and Johan Ugander. Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference*, 5(1), 2017.

[26] Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer, 1992.

[27] Bruce Fischl. Freesurfer. *Neuroimage*, 62(2):774–781, 2012.

[28] Jean-Pierre Fouque, George Papanicolaou, Ronnie Sircar, and Knut Sølna. *Multiscale stochastic volatility for equity, interest rate, and credit derivatives*. Cambridge University Press, 2011.

[29] Nir Friedman, Moises Goldszmidt, and Abraham Wyner. Data analysis with bayesian networks: A bootstrap approach. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 196–205. Morgan Kaufmann Publishers Inc., 1999.

[30] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

[31] Piotr Fryzlewicz. Tail-greedy bottom-up data decompositions and fast mulitple change-point detection. *Annals of Statistics*, 2017.

[32] Piotr Fryzlewicz and Catherine Timmermans. Shah: Shape-adaptive haar wavelets for image processing. *Journal of Computational and Graphical Statistics*, 25(3):879–898, 2016.

[33] Apratim Ganguly and Wolfgang Polonik. Local neighborhood fusion in locally constant gaussian graphical models. *arXiv preprint arXiv:1410.8766*, 2014.

[34] Matan Gavish, Boaz Nadler, and Ronald R Coifman. Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning. 2010.

[35] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, et al. Meg and eeg data analysis with mne-python. *Frontiers in neuroscience*, 7:267, 2013.

[36] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, pages 424–438, 1969.

[37] MATTI S Hamalainen and Jukka Sarvas. Realistic conductivity geometry model of the human head for interpretation of neuromagnetic data. *IEEE transactions on biomedical engineering*, 36(2):165–171, 1989.

[38] James D Hamilton. Oil and the macroeconomy since world war ii. *The Journal of Political Economy*, pages 228–248, 1983.

[39] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[40] Craig Hiemstra and Jonathan D Jones. Testing for linear and nonlinear granger causality in the stock price-volume relation. *The Journal of Finance*, 49(5):1639–1664, 1994.

[41] Shawndra Hill, Deepak K Agarwal, Robert Bell, and Chris Volinsky. Building an effective representation for dynamic networks. *Journal of Computational and Graphical Statistics*, 15(3):584–608, 2006.

[42] Christopher J Honey, Rolf Kötter, Michael Breakspear, and Olaf Sporns. Network structure of cerebral cortex shapes functional connectivity on multiple time scales. *Proc. Natn. Acad. Sci. USA*, 104(24):10240–10245, 2007.

[43] Jeff Irion and Naoki Saito. Efficient approximation and denoising of graph signals using the multiscale basis dictionaries. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):607–616, 2017.

[44] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

[45] Bomin Kim, Kevin Lee, Lingzhou Xue, and Xiaoyue Niu. A review of dynamic network models with latent variables. *arXiv preprint arXiv:1711.10421*, 2017.

[46] Eric D. Kolaczyk. *Statistical Analysis of Network Data: Methods and Models*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[47] Eric D Kolaczyk and Gábor Csárdi. *Statistical analysis of network data with R*, volume 65. Springer.

[48] Eric D Kolaczyk and Robert D Nowak. Multiscale generalised linear models for nonparametric function estimation. *Biometrika*, 92(1):119–133, 2005.

[49] Risi Kondor, Nedelina Teneva, and Vikas Garg. Multiresolution matrix factorization. In *International Conference on Machine Learning*, pages 1620–1628, 2014.

[50] Nancy J Kopell, Howard J Gritton, Miles A Whittington, and Mark A Kramer. Beyond the connectome: the dynome. *Neuron*, 83(6):1319–1328, 2014.

[51] Gueorgi Kossinets and Duncan J Watts. Empirical analysis of an evolving social network. *science*, 311(5757):88–90, 2006.

[52] Hans R Kunsch. The jackknife and the bootstrap for general stationary observations. *The annals of Statistics*, pages 1217–1241, 1989.

[53] Ann B Lee, Boaz Nadler, and Larry Wasserman. Treelets: an adaptive multi-scale basis for sparse unordered data. *The Annals of Applied Statistics*, 2(2):435–471, 2008.

[54] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187. ACM, 2005.

[55] Jonathan Q Li and Andrew R Barron. Mixture density estimation. In *Advances in neural information processing systems*, pages 279–285, 2000.

[56] JM Loh and ML Stein. Bootstrapping a spatial point process. *Statistica Sinica*, pages 69–101, 2004.

[57] CJ Long, EN Brown, C Triantafyllou, I Aharon, LL Wald, and V Solo. Nonstationary noise estimation in functional mri. *NeuroImage*, 28(4):890–903, 2005.

[58] Mary M Louie and Eric D Kolaczyk. A multiscale method for disease mapping in spatial epidemiology. *Statistics in medicine*, 25(8):1287–1306, 2006.

[59] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7):674–693, 1989.

[60] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *Ann. Statist.*, pages 1436–1462, 2006.

[61] Nitai D Mukhopadhyay and Snigdhansu Chatterjee. Causality and pathway search in microarray time series experiment. *Bioinformatics*, 23(4):442–449, 2007.

[62] Alfred Müller. Stochastic ordering of multivariate normal distributions. *Annals of the Institute of Statistical Mathematics*, 53(3):567–575, 2001.

[63] Fionn Murtagh. The haar wavelet transform of a dendrogram. *Journal of Classification*, 24(1):3–32, 2007.

[64] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[65] Michael Newman. Network structure from rich but noisy data. *Nature Physics*, In press.

[66] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing. *arXiv preprint arXiv:1712.00468*, 2017.

[67] Carey E Priebe, John M Conroy, David J Marchette, and Youngser Park. Scan statistics on enron graphs. *Computational & Mathematical Organization Theory*, 11(3):229–247, 2005.

[68] Carey E Priebe, Daniel L Sussman, Minh Tang, and Joshua T Vogelstein. Statistical inference on errorfully observed graphs. *Journal of Computational and Graphical Statistics*, 24(4):930–953, 2015.

[69] Kunjan D Rana and Lucia M Vaina. Functional roles of 10 hz alpha-band power modulating engagement and disengagement of cortical networks in a complex visual motion task. *PloS one*, 9(10):e107715, 2014.

[70] Aliaksei Sandryhaila and José MF Moura. Discrete signal processing on graphs. *IEEE transactions on signal processing*, 61(7):1644–1656, 2013.

[71] Aliaksei Sandryhaila and Jose MF Moura. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Processing Magazine*, 31(5):80–90, 2014.

[72] David I Shuman, Mohammad Javad Faraji, and Pierre Vandergheynst. A multiscale pyramid transform for graph signals. *IEEE Transactions on Signal Processing*, 64(8):2119–2134.

[73] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

[74] Christopher A Sims. Money, income, and causality. *The American Eco-*

nomic Review, 62(4):540–552, 1972.

[75] Aarti Singh, Robert Nowak, and Robert Calderbank. Detecting weak but hierarchically-structured patterns in networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 749–756, 2010.

[76] Stephen M Smith, Karla L Miller, Gholamreza Salimi-Khorshidi, Matthew Webster, Christian F Beckmann, Thomas E Nichols, Joseph D Ramsey, and Mark W Woolrich. Network modelling methods for fmri. *Neuroimage*, 54(2):875–891, 2011.

[77] Mary E Thompson, Lilia L Ramirez Ramirez, Vyacheslav Lyubchich, and Yulia R Gel. Using the bootstrap for statistical inference on random graphs. *Canadian Journal of Statistics*, 44(1):3–24, 2016.

[78] Dan J Wang, Xiaolin Shi, Daniel A McFarland, and Jure Leskovec. Measurement error in network data: A re-classification. *Social Networks*, 34(4):396–409, 2012.

[79] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[80] Rebecca M Willett and Robert D Nowak. Multiscale poisson intensity and density estimation. *IEEE Transactions on Information Theory*, 53(9):3171–3187, 2007.

[81] Yi-Ching Yao. Estimating the number of change-points via schwarz'criterion. *Statistics & Probability Letters*, 6(3):181–189, 1988.

[82] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *J. R. Statist. Soc. B*, 68(1):49–67, 2006.

[83] Dong Zhang and Jie Liang. Graph-based transform for 2d piecewise smooth signals with random discontinuity locations. *IEEE Transactions on Image*

Processing, 26(4):1679–1693, 2017.

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

Curriculum Vitae