New Graph Learning Techniques for Networked Knowledge Inference and Protection

by **Baoling Shan**

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Prof. Ren Ping Liu and Prof. Eryk Dutkiewicz

School of Electrical and Data Engineering
Faculty of Engineering and IT
University of Technology Sydney

January 3, 2024

Certificate of Authorship / Originality

I, Baoling Shan, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: January 3, 2024

Abstract

Graph topologies have been detected in data captured in social networks, biological networks, traffic networks, smart grids, and ecological networks. Graphs provide effective means to represent the statistical dependence or similarity among signals observed at different vertices. A critical challenge is to excavate graphs underlying observed signals, because of non-convex problem structure and associated high computational requirements. On the other hand, latent graph structure and stimulus of graph data contain critical private information, such as brain disorders in functional magnetic resonance imaging data, and can be exploited to identify individuals. It is critical to perturb the latent information while maintaining the utility of the data, which, unfortunately, has never been addressed.

In this thesis, a new alternating optimization (AO) based graph learning technique is investigated to solve the challenge. However, the fidelity of the topologies inferred from the graph signals was penalized due to the use of the AO-based approximation. To surpass the limitations of the AO-based approximation, we propose a new graph learning technique that is able to efficiently infer the graph structure underlying observed graph signals by deriving a new closed-form analytic expression for the graph Fourier transform (GFT) basis, which depends deterministically on the observed signals. The new graph learning technique is applied to accurately infer the graph structure of COVID-19 data, helping to reveal the correlation of pandemic dynamics among different countries and identify influential countries for pandemic response analysis.

To protect the latent privacy of the latent graph structures and stimuli of graph-structured data, a novel approach has been proposed to obfuscate the latent information and maximize its utility. We first analyze the GFT basis that captures the latent graph structures, and the latent stimuli that are the spectral-domain inputs to the latent graphs. Then, we formulate and decouple a new multi-objective problem to alternately obfuscate the GFT basis and stimuli. The difference-of-convex (DC) programming and Stiefel manifold gradient descent are orchestrated to obfuscate the GFT basis. The DC programming and gradient descent are employed to perturb the spectral-domain stimuli. Experiments conducted on an attention-deficit hyperactivity disorder dataset demonstrate that our approach can substantially outperform its differential privacy-based benchmark in the face of the latest graph inference attacks.

Acknowledgements

First of all, I would like to express my sincere gratitude to my principal supervisors Profs. Ren Ping Liu and Wei Ni, for their professional guidance, exceptional expertise, and continuous support throughout my doctoral journey. Their profound knowledge, extensive experience, and insightful feedback were instrumental in shaping the direction and quality of my research.

I am also grateful to my friends and colleagues Dr. Xin Yuan and Dr. Dongwen Yang, who generously shared their time and knowledge, and provided valuable insights and helpful discussion during my research. Their contributions have enriched the quality of my work.

I would like to extend my appreciation to my co-supervisor Prof. Eryk Dutkiewicz and my collaborator Prof. Xin Wang, for their invaluable feedback and scholarly insights. Their expertise and thoughtful engagement enrich the content of my work.

Furthermore, I would also like to express my heartful thanks to my family and friends for their unwavering support, love, and belief in my abilities. Their encouragement and understanding are the constant sources of motivation throughout this journey.

Finally, I would like to acknowledge the financial support and scholarships received from the China Scholarship Council and the University of Technology Sydney. Their support has provided the resources necessary to conduct my research.

Baoling Shan
January 3, 2024
Sydney, Australia

Contents

List of Figures

List of Tables

List of Publications

Papers Published

- **B. Shan**, W. Ni, X. Yuan, D. Yang, X. Wang, and R.P. Liu, "Graph learning from band-limited data by graph Fourier transform analysis," *Signal Processing (SP).* 2023 Jun 1;207:108950.

- **B. Shan**, X. Yuan, W. Ni, X. Wang, R.P. Liu, and E. Dutkiewicz, "Novel Graph Topology Learning for Spatio-Temporal Analysis of COVID-19 Spread," *IEEE J. Biomed. Health Inform. (JBHI)* , 27.6(2023): 2693-2704.

- **B. Shan**, X. Yuan, W. Ni, X. Wang, R.P. Liu, and E. Dutkiewicz, "Preserving the Privacy of Latent Information for Graph-Structured Data," *IEEE Trans. Inf. Forensics Secur.(TIFS)*, 07.8(2023): 5041 - 5055.

- Z. Li, T. Shi, **B. Shan**, and S. Li, "Multi-Intensity Planes Constellation and Code Design Based Color-Shift Keying for Visible Light Communications," *IEEE Internet Things J.* accept.

- Z. Li, T. Shi, **B. Shan**, and S. Li, "Color-Shift Keying Code Design in Visible Light Communications Using Finite-State Machines," *IEEE Communications Letters* , 26.6 (2022): 1333-1337.

- Z. Li, T. Shi, **B. Shan**, and S. Li, "Code design for run-length control in visible light communication," *Frontiers of Information Technology and Electronic Engineering*, 21.9 (2020): 1397-1411.

- Z. Li, H. Yu, **B. Shan**, D. Zou, and S. Li, "New run-length limited codes in on–off keying visible light communication systems," *IEEE Wireless Communications Letters* , 9(2),(2019):148-151.

Papers Under Review

- **B. Shan**, X. Yuan, W. Ni, X. Wang, and R.P. Liu, “Graph learning from band-limited data by graph Fourier transform analysis,” *IEEE Trans. Signal Inf. Process. Netw. (TSIPN)* . (Major Revision)

Chapter 1

Introduction

1.1 Background

An important type of data is graph data, which is the data with latent graph structures (or in other words, a set of concurrent time series with underlying correlations) observed extensively in the areas of physics, biology, transportation, energy, engineering, and social science [1]–[4]. An example of graph data is a brain signal, such as electroencephalogram (EEG) signals [5] acquired by attaching electrodes to the scalp of a patient to measure the electrical activities in its brain [6], as well as blood-oxygen-level-dependent time series obtained by performing functional magnetic resonance imaging (fMRI) on the brain [7]. More examples include social network data released by users on social platforms, such as Facebook, Twitter, and WeChat, which may contain social graphs and large amounts of possibly sensitive and private information on users and their interactions [8], [9].

Graph signal processing (GSP) [10] provides a new paradigm to process and analyze these graph signals with network nature and extends signal processing techniques to graph signal filtering [10], compressing [11], sampling [12], and recovers data in irregular domains. GSP is a general tool to uncover the dependency, physical proximity, or other properties underlying datasets captured in networks [13], [14] and to analyze how the graph signals can impact the algebraic characteristics of the underlying topology. The graph topology captures the instinct relationships of unstructured data encoded on different entries of the graph and produces the graph topology inference from observed signals [15]–[24]. The majority of GSP research conducted so far assumes that the knowledge of the underlying topology is known. In practice, the graph topologies are not always readily available, restricting the use of GSP [25]. Therefore, graph learning from the underlying data and signals in network systems, e.g., biological networks [26], traffic networks [27], [28], smart grid networks [29], communication networks [30], ecological networks [31], and social networks [32], have been attracting considerable raising interest.

Nowadays, privacy is a significant concern for some graph-structured data. For example, the latent graph structures of the brain network data could expose personal health conditions, such as attention-deficit hyperactivity disorder (ADHD) [33] and Alzheimer's disease (AD) [34], under graph interference attacks (based on graph learning techniques [21], [35], [36]). The health conditions could be exploited to reveal the identities of those whose brain network data is under the graph inference attacks [37], [38]. On the other hand, the latent stimuli, which are the input to the latent graphs and derive the output of the observed graph-structured data, are also part of the private information [39]. The bandwidth and waveform shapes of the stimuli can be used as the identifiers of individuals [40]. In this sense, it is critical to preserve the privacy of the graph-structured data by obfuscating the latent graph structures underlying the graph data and the stimulus to the latent graphs, while restraining the changes to the graph data to best maintain its utility.

1.2 Motivations

The graph topology captures the instinct relationships of unstructured data encoded on different entries of the graph and graph learning is an efficient technique used to uncover the latent graph topologies of data [41]. Classical graph topology inference methods, such as graph lasso [16] and covariance selection [15], estimated the covariance matrices of graph signals. More recent graph learning techniques enforced smoothness to graph signals prior to topology inference [18]–[22], [42], [43]. In other words, it has been typically assumed that the frequency-domain representations of graph signals have unlimited bandwidths, e.g., for mathematical tractability.

With the latent graph topologies, the captured data can be transformed into the frequency domain, where data can be effectively processed. This is critical to recover missing data in part of a network [44], or verify the authenticity of data [45]. For the sake of generality, it is reasonable to consider band-limited graph signals, i.e., the frequency-domain representation of the signals can have a finite bandwidth. The band-limitedness is observed in practice, e.g., fMRI data in brain networks [46]. The widely studied smooth graph signals with unlimited frequency-domain bandwidths [18] can be viewed as a special case of band-limited graph signals. Existing graph learning methods cannot accurately and efficiently infer the graph topology (i.e., graph Laplacian [47]) of band-limited graph signals due to difficulties in joint estimation of both the frequency-domain representation and the graph Fourier transform (GFT) basis converting captured data to the frequency domain.

Since the outbreak of COVID-19, numerous studies have been explored to comprehend the spatio-temporal characteristics of the virus spread and control the spread of the pandemic [48]. As of April 2022, the COVID-19 pandemic has led to escalating global health, economic, and social challenges [49], [50] and Europe had recorded 192.09 million confirmed cases and over two million deaths due to the COVID-19 pandemic [1]. Unfortunately, it is difficult to implement evidence-based policies for COVID-19 due to a lack of adequate evidence in policy-making and research [51]. While it is possible to estimate the growth rates of confirmed cases and deaths [52], the relationships between pairs of countries are still unknown as far as the COVID-19 development is concerned. Datasets about ongoing situations in different countries are likely to show spatial-temporal patterns since virus spread tends to follow geographic trends. A spatial-temporal analysis of confirmed COVID-19 cases may also shed light on its evolution.

The record of pandemic evolution in Europe is known to be complex, variable, and non-linear [53]. Consequently, it is essential to uncover hidden information about SARS-CoV-2 as new virus variants emerge. To gain insights into the dynamic spread of the pandemic, one approach is to generate and analyze diffusion graph topologies of the COVID-19 pandemic using graph-theoretic metrics [54]–[57]. These graph-based models help uncover the interconnectedness between different locations and capture the temporal aspect of the spread [58]. Existing studies have examined the

[1] https://coronavirus.jhu.edu/data

spread of epidemics as a complex system by assessing the degree of correlation or synchronization between time-series data [59], [60]. While these approaches provide valuable insights, the transmission dynamics of new variants of SARS-CoV-2 call for additional methods that go beyond simple correlation or synchronization analyses. A deeper understanding of the transmission dynamics of the new variants of SARS-CoV-2 requires new methods beyond assessing correlation or synchronization. There is a need to explore the underlying local structures in the data and reveal the relationships between different countries to understand the spatio-temporal spread of the virus.

Another research issue is the privacy of graph-structured data. Privacy is a significant concern for some graph-structured data, for example, brain network data obtained by fMRI [7]. The graph-structured brain data can be held by the Department of Neurology in a hospital, and shared with and used by other departments or clinics for big data analytics (e.g., detecting or modeling changes in blood flow that occur with brain activity) or educational purposes. On the one hand, the latent graph structures within brain network data could unveil personal health conditions, such as ADHD and AD, when subjected to graph interference attacks employing graph learning techniques [21], [35], [36]. These conditions could potentially expose patient identities [37], [38]. The latent stimuli within graph-structured data, serving as inputs to the latent graphs and contributing to the observed output, also constitute private information [39]. Bandwidth and waveform characteristics of these stimuli could serve as unique identifiers for individuals.

Preserving the privacy of the graph-structured data, more explicitly, the privacy of the latent graph structures and stimuli underlying the data has never been addressed in the literature. Existing studies have been dedicated to the privacy of graphs (as opposed to the graph-structured data), e.g., through k-anonymity [61], node or edge perturbation [62], and graph differential privacy (DP) [63]–[65]. The results of the studies cannot apply to the graph-structured data, i.e., a set of time series with correlations. In light of the notion of DP [66], a possible solution to obfuscating graph data is to add random noises to the graph data. However, to what extent can this DP-based obfuscation effectively preserve the privacy of the latent graphs and stimuli is unclear. As a matter of fact, the latent graph structure is less susceptible to the added, typically Gaussian noises on the graph data, as revealed in this thesis.

1.3 Research Contributions

This thesis aims to investigate different graph learning methods and applications and preserve the privacy of the latent graph structures and stimulus of graph-structured data. The main contributions are summarized as follows.

- We propose a new graph learning technique to learn the graph Laplacian of the band-limited, brain signals. The key idea is that we show the frequency-domain signals depend deterministically on the graph Laplacian. The joint estimation can be transferred to only concern the graph Laplacian. Alternating optimization (AO) can be applied to learn the eigenvectors and eigenvalues of the graph Laplacian in an iterative manner. Another important aspect is that we reveal the feasible solution of the eigenvectors on a Stiefel manifold which can be efficiently solved using Stiefel manifold dual gradient descent. Given the eigenvectors, the eigenvalues can be obtained using the alternating direction method of multipliers (ADMM). Experiments indicate that the new technique can infer the graph topology of brain signals with substantially better accuracy than the prior art. The technique draws findings validated by neuroanatomical studies.

- We present a new graph learning technique that is able to efficiently infer the graph structure underlying observed graph signals. The key idea is that we reveal the intrinsic relation between the frequency-domain representation of general band-limited graph signals and the GFT basis. Accordingly, we derive a new closed-form analytic expression for the GFT basis, which depends deterministically on the observed signals (as opposed to being solved numerically and approximately in the literature). Given the GFT basis, the estimation of the graph Laplacian, more explicitly, its eigenvalues, is convex and efficiently solved using the ADMM. Simulations based on synthetic data and experiments based on public weather and brain signal datasets show that the new technique outperforms the state of the art in accuracy and efficiency.

- We apply the new graph-learning technique to accurately infer the graph structure of COVID-19 data, helping to reveal the correlation of pandemic dynamics among different countries and identify influential countries for pandemic response analysis. The new technique estimates the graph Laplacian of the COVID-19 data by first deriving analytically its precise eigenvectors, also known as the GFT basis. Given the eigenvectors, the eigenvalues of the graph Laplacian are readily estimated using convex optimization. With the graph Laplacian, we analyze the confirmed cases of different COVID-19 variants among European countries based on centrality measures and identify a different set of the most influential and representative countries from the current techniques. The accuracy of the new method is validated by repurposing part of COVID-19 data to be the test data and gauging the capability of the method to recover missing test data, showing 33.3% better in root mean squared error (RMSE) and 11.11% better in the correlation of determination than existing techniques. The set of identified influential countries by the method is anticipated to be meaningful and contribute to the study of COVID-19 spread.

- We present a novel approach to obfuscating latent graph structure and stimulus and maximizing the utility of graph-structure data, which can be exploited to identify individuals, such as brain disorders in functional magnetic resonance imaging data. The problem of perturbing latent information while maintaining the utility of the data, unfortunately, has never been addressed. Specifically, we first analyze the GFT basis that captures the latent graph structures, and the latent stimuli that are the spectral-domain inputs to the latent graphs. Then, we formulate and decouple a new multi-objective problem to alternately obfuscate the GFT basis and stimuli. The difference-of-convex (DC) programming and Stiefel manifold gradient descent are orchestrated to obfuscate the GFT basis. The DC programming and gradient descent are employed to perturb the spectral-domain stimuli. Experiments conducted on an attention-deficit hyperactivity disorder dataset demonstrate that our approach can substantially outperform its differential privacy-based benchmark in the face of the latest graph inference attacks.

1.4 Thesis Structure

The structure of the thesis is organized as follows.

Chapter 2 reviews the relevant works on the studies of graph learning techniques and graph-structured data privacy-preserving techniques. The graph learning techniques are reviewed first including the traditional graph learning method, graph learning from smooth observations, and graph learning from ban-limited observations. Then we introduce the techniques for preserving the privacy of graph data.

Chapter 3 proposes a new graph learning technique, which learns weighted and undirected graph topologies, more specifically, the graph Laplacian matrices, from fMRI-based, band-limited brain signals.

Chapter 4 presents a new graph learning technique for general band-limited graph signals, which learns the graph topology (i.e., the graph Laplacian matrix) of observed bandlimited graph signals to overcome the high computational complexity of the graph learning method proposed in Chapter 3.

Chapter 5 presents a new graph-learning technique to accurately infer the graph structure of COVID-19 data, helping to reveal the correlation of pandemic dynamics among different countries and identify influential countries for pandemic response analysis.

Chapter 6 presents a new approach to protect the privacy of the latent information underlying graph-structured data (e.g., the graph structure and the stimulus underlying the observed graph-structured data) while minimizing the perturbations on the observed graph-structured data to maintain the utility of the data.

Chapter 7 provides the conclusion of the thesis and summarizes our future work.

Chapter 2

Literature Review

2.1 Graph Signal Processing

Graph signal processing (GSP) [67], [68] provides a powerful framework for processing and analyzing signals that are associated with graph structures. It extends traditional signal processing techniques to signals that are defined on graphs, enabling the analysis of various data types and applications [69]. GSP encompasses a range of methods and algorithms that operate on graph-structured data. These include graph signal filtering [10], [70], which involves modifying or extracting specific components of graph signals based on their spectral properties. Graph signal compression [11] aims to reduce the size of graph signals while preserving important information. Graph signal sampling [12], [13] focuses on selecting a subset of graph nodes to efficiently represent the entire signal. Furthermore, GSP encompasses techniques for recovering data on irregular graph domains [71].

The main objective of GSP is to uncover the underlying dependencies, physical proximity, or other properties within datasets that are indexed by vertices in a graph [13]. By considering the graph structure, GSP enables the exploration and analysis of relationships and patterns in the data, leading to insights and interpretations that may not be apparent in traditional signal processing approaches [39], [72], [73]. However, one challenge in applying GSP is that the graph structure is not always readily available [25]. In earlier studies, the underlying graph topology was assumed to be known, and GSP was used to analyze how graph signals can affect the algebraic characteristics of the underlying graphs [74]. More recently, efforts have been made to reconstruct the network structure from the data itself, allowing for the expression of complex relations in real systems [74]. GSP has gained increasing interest and has been applied in various fields, including social networks [75], neuroscience [26], image processing [76], [77], sensor networks [78]–[80], and communication systems [81], [82]. It provides a powerful framework for understanding and exploiting the rich information encoded in graph-structured data, enabling advancements in data analysis, pattern recognition, and decision-making processes.

2.2 Graph Learning

Traditional methods [15], [16] estimated the covariance matrix to capture the linear/nonlinear and symmetric pairwise or directional dependency among vertex-indexed signals. Such methods only captured pairwise correlations and did not reflect the causality of latent network structures [83].

Later, graph learning techniques were developed to infer the topology from observations, where graph signals were assumed to be smooth; in other words, their frequency-domain representations have unlimited bandwidths. These techniques are not suitable for band-limited signals. Dong *et al.* [41] outlined techniques to solve graph learning problems for globally and locally smooth models, and summarized the potential benefit of GSP-based graph inference methods in many theoretical and practical applications, such as image coding, brain functional connectivity analysis, and meteorology analysis. Chepuri *et al.* [20] studied the learning of a sparse graph to adequately explain observed data under a smoothness prior. They provided an AO algorithm and a one-step convex relaxation-based solution by modeling the learning problem as a sparse edge sampling function. Dong *et al.* [18] inferred the graph Laplacian by minimizing the variations of the smooth signals based on the *a-priori* information about the structure. This method was recently used in [45] to reconstruct missing air pollution data and experimentally verified. Kalofolias *et al.* [19] extended this idea by constructing graph learning as a weighted sparsity problem, and learned a valid structure represented by the adjacent weighted matrix using primal-dual optimization. The method of [19] was recently used in [42] to study the functional connectivity of brain networks from fMRI time series. Saboksayr *et al.* [43] extended the method of [19] to support general multi-class smooth signals. Egilmez *et al.* [84] proposed a block-coordinate descent-based algorithm to learn the graph Laplacian. The original problem was decomposed into subproblems, based on the structural constraints and optimality conditions of the Laplacian estimation. The subproblems were solved in an alternating manner at each iteration. Egilmez's algorithm was applied in [85] for video compression with improvements over the widely used Karhunen-Loeve transform.

Some recent works learned topologies generated by diffusion processes on graphs, typically under the assumption of stationarity; i.e., the graph diffusion operator (e.g., the adjacency or Laplacian matrix) has the same eigenvectors as the covariance matrix of observed signals. Segarra *et al.* [21] estimated a new graph shift operator by minimizing the ℓ_1-norm of the operator based on the complete or partial knowledge of the eigenvectors of the covariance. They used independent samples of signals to evaluate the eigenvectors and then estimated the eigenvalues given the eigenvectors. The obtained graph shift operator supported band-limited signals when partial knowledge was available about the eigenvectors and the observed signal samples were band-limited. Pasdeloup *et al.* [22] studied the case where signals were independent and identically distributed (i.i.d.) and observed after diffusion on a graph. It was verified that the set of graphs has impacts on the eigenvectors of the covariance matrix used to recover the graph topology.

A few existing studies [35], [36] have attempted to learn the latent graphs of band-limited signals. Their accuracy and efficiency were penalized by their AO-based approximate solvers. Specifically, Sardellitti *et al.* [35] discovered a block sparse representation of band-limited graph signals and developed a strategy to associate a graph with observed band-limited signals. The strategy started by learning an orthonormal sparsifying transform based on the AO. The resulting problem of the graph estimation became convex. The strategy recovered the graph Laplacian with convex optimization techniques.

2.3 Graph Learning for Brain Networks Analysis

Network neuroscience plays a crucial role in advancing our understanding of the structure and function of the human brain. It adopts a network perspective by considering the brain as a complex system composed of multiple regions of interest (ROIs), often referred to as brain network nodes [86]. Graph theory [87] has been a crucial tool to analyze complex brain networks, and revealed several non-trivial features of brains, such as modularity and small-worldness, by studying the fMRI time series (i.e., blood-oxygen-level dependent time series) amongst the ROIs in a brain [86].

Traditional brain graph learning methods [15]–[17] estimated the covariance matrix to capture the linear/nonlinear and symmetric pairwise or directional dependency among vertex-indexed observed signals. Pearson's correlation has been one of the most common methods for measuring the pairwise functional relationships between brain regions. However, Pearson's correlation focuses on strong direct marginal correlations of the fMRI time series between two brain regions, and overlooks the latent network effects of other brain regions. Albert *et al.* [88] used partial correlations to measure interactions between any two ROIs. The partial correlation quantifies the dependency between two ROIs by regressing out the other ROIs, leading to difficulties in the suppression of the confounding effect from the other ROIs. Those methods only captured pairwise correlations and cannot reflect the accurate causality in the latent network structures of brains [83], [89], [90]. Sparse inverse covariance estimation (SICE) [91] is another popular technique for measuring the intensity of the most significant direct connection between ROIs. SICE is a principled partial correlation algorithm. It tends to evaluate the sparsest reconstructive coefficient of each ROI and capture only local structures, rather than a representation of the global structure.

Recent graph learning techniques have attempted to address the limitations of SICE by incorporating both local and global features to establish a representation of the entire graph [92]. These techniques deal with specific properties of observed signals, such as smoothness [18], [19] and stationarity [21], but are unsuited for band-limited brain signals. On the other hand, fMRI-based brain data exhibit a distinct characteristic of band-limitedness and smoothness due to their underlying physiological properties [36]. They are different from typically considered (band-unlimited) smooth graph signals. In other words, the signals are sparse in the frequency domain. As a result, existing graph learning algorithms developed under the assumptions of smoothness and stationarity cannot readily apply to fMRI-based brain signals.

The band-limitedness is a widely observed property of fMRI-based brain data [35], [46]. The observed signals can be sparse in the frequency-domain. With the inferred graph topology, it is possible to recover the graph signals throughout the entire brain network by only observing part of the signals. Sardellitti *et al.* [93] discovered a block sparse representation of general graph data by enforcing the band-limitedness of observed signals. Instead of assuming any diffusion process over a graph, a strategy was developed to relate a graph to the received band-limited signals. The strategy started by estimating an orthonormal sparsifying transform based on AO. The resulting problem of graph estimation was convex. The graph Laplacian matrix was then recovered using convex optimization techniques. Humbert *et al.* [94] considered graph learning from multivariate signals with both smoothness and band-limitedness. A three-step AO-based algorithm was developed to leverage manifold gradient descent and linear programming to learn the Laplacian of the graph.

Graph Convolutional Networks (GCNs) [95] have also been employed in more recent approaches for learning brain networks through various methods. Based on Pearson's correlation matrix as the node features, Zhao *et al.* [96] developed a novel dynamic GCN to distinguish ADHD patients from health control for a better understanding the ADHD-associated brain dysfunctions. Li *et al.* [97] developed a BrainGNN framework based on a graph neural network by using the topological and functional information of fMRI for classification tasks. Zhou *et al.* [98] designed an interpretable GCN model to identify and classify Alzheimer's disease and quantify the discriminative features of the brain connectivity patterns.

TABLE 2.1 collates the relevant studies to learn brain networks.

Table 2.1: The state of the art in graph learning techniques for brain signal analysis

Method	**Description**
Pearson's correlation	This algorithm measures the pairwise functional relationships between brain regions with a focus on strong direct marginal correlations of the fMRI time series between two brain regions, and overlooks the latent network effects of other brain regions.
Partial correlation	This algorithm measures the dependency between two nodes by regressing out the remaining ones, leading to difficulties in removing the confounding effect from other nodes.
SICE [91]	As the principled method for partial correlation, SICE tends to evaluate the sparsest reconstructive coefficient of each ROI and capture only local structures, rather than a representation of the global structure.
Dong's algorithm [18]	This algorithm is an alternating minimization algorithm that infers the assumption of the smoothness of the signals.
Kalofolias' algorithm [19]	This algorithm is a scalable primal-dual algorithm that learns the topological structures represented by the adjacent weighted matrix of graphs.

Sardellitti's Total Variation (TV) algorithm [93]	This algorithm is a two-step strategy consisting of (a) learning the orthonormal sparsifying transform from data via AO and (b) then recovering the Laplacian matrix from the sparsifying transform via convex optimization.
Sardellitti's Estimated-Signal-Aid (ESA) algorithm [35]	Different from Sardellitti's TV graph learning algorithm, this two-step strategy recovers the Laplacian matrix from the sparsifying transform and GFT coefficients by using convex optimization in Step 2.
Humbert's algorithm [94]	This is an AO-based algorithm with three alternating steps relying on standard minimization methods, i.e., manifold gradient descent and linear programming. It learns graphs from multivariate signals with smoothness and band-limitedness.

2.4 Graph Learning for COVID-19 Analysis

The COVID-19 pandemic is exacerbating global health, economic, and social challenges. As of April 2022, Europe recorded 192.09 million confirmed cases and more than two million deaths. Since the pandemic outbreak, researchers from various fields have extensively investigated the spread of the disease. The complex network theory based on a pair-wise configuration has been widely utilized for modeling the topological connectivity of the COVID-19 data on a global perspective [50], [99], [100]. Azad *et al.* [99] tracked the spread of COVID-19 by utilizing the network analysis in India based on the travel history of infected patients and revealed that international travel played a key role in the pandemic outbreak in a country. Jo *et al.* [100] developed an infected network using the contact tracing information of confirmed cases, and found that governmental measures had a strong impact on the COVID-19 spread network in Seoul. Through modeling tourism mobility as a complex network, Tsiotas *et al.* [50] created a multidimensional framework to understand the COVID-19 spread across countries. Chu *et al.* [101] constructed an air travel network structure to visualize the connectedness and evolution of the pandemic. Travel subnetworks were formed by aggregating airport data at the national level and adding it to a matrix capturing the flight recurrences between countries. Using a similar conceptualization, they also developed a pandemic space approach [102] that uses the historical correlation of confirmed cases to locate the connections between different countries. By integrating Bayesian parameter inference with a Watts–Strogatz small-world network epidemiological model, Syga *et al.* [103] inferred a time-varying COVID-19 transmission network in Germany. It was shown that government interventions reduced random contacts and transmission probabilities.

Numerous methods have been designed to infer the pandemic's time-dependent transmission network, compared to previous works on network-based models. For instance, the correlation coefficients were exploited to capture the linear/nonlinear and symmetric pairwise matrix between different regions [57], [58], [104], [105]. So *et al.* [57] constructed dynamic pandemic networks over time for 164 countries to forecast and assess the risk of the pandemic using network statistics. The connections in the networks were established based on the relationships of changes in the count of reported cases between the two regions. Pan *et al.* [105] used the Pearson correlation coefficient, time-lagged cross-correlation, and dynamic time wrapping to examine interactions in the evolution of pandemics across the different states of the US. McMahon *et al.* [104] examined the spatial correlations of new active cases across different states in the US and assessed their magnitude over time. Their results showed stronger correlations between urban areas compared to rural areas, revealing that the pandemic spread was largely driven by travel between cities. Using spatio-temporal correlation, Aral *et al.* [58] identified distinct spatial clusters and spatial associations among COVID-19 cases in Turkey, revealing that spatial analysis helped explain the spread of the disease.

Alguliyev *et al.* [106] created a conceptual graph model by taking into account various epidemiological traits of COVID-19, such as social distance, the period of contact with an infected individual, and demographic characteristics based on location, thereby enabling a visual representation of virus propagation. This helps determine undetected cases of infection. Ieracitano *et al.* [107] adopted a deep learning technique based on fuzzy logic to create a classification system for the early identification of COVID-19 cases utilizing portable chest X-ray (CXR) images. Absar *et al.* [108] developed a computer-assisted system for the automatic classification of CXR images of COVID-19 utilizing the Support Vector Machine (SVM) to enable fast diagnosis of COVID-19.

2.5 Privacy Preserving for Graphs

Numerous privacy-preserving techniques for graph have been developed, e.g., k-anonymity [61], node and edge perturbation [109], and graph DP [66]. These methods do not apply to graph data, that is, the three-dimensional data with latent graph structures (or, in other words, a set of concurrent time series), such as the brain signals. The methods would require the latent graphs to be extracted from the data, and separately obfuscated. The usefulness or effectiveness of graph data would degrade.

Yuan *et al.* [61] modeled a k-degree-l-diversity anonymity system and proposed an anonymization method that adds noisy nodes into the original graph to protect structural information. k measures the number of vertices with the same degrees. l indicates the distinct labels. Ding *et al.* [110] developed a privacy-preserving framework to anonymize graphs, which defined an information loss matrix for graph datasets based on a k-decomposition method. k-anonymity was performed on isomorphic subgraphs to reduce computational overhead. The privacy-preserving capability of these k-anonymity methods is heavily influenced by the magnitude of k. However, a higher k value would inevitably lead to a lower data utility.

Approaches based on the node or edge perturbation create rules for processing nodes and edges, including edge modification, nodes clustering, and random walk. Node and edge approaches are achieved by inserting or removing vertices and edges. Hay *et al.* [109] developed a low-complexity random perturbation method by randomly removing and inserting edges on the graph. Yu *et al.* [111] introduced a perturbation scheme derived from local clustering, where edges were modified to mitigate the risk of privacy compromise while minimizing the loss of network structures and data.

DP was proposed to quantify the privacy loss of individuals whose data undergoes algorithmic processing [66]. Graph DP is a special case of DP. It is typically divided into node-DP and edge-DP, built upon the concept of adding and removing a single node or edge. In this sense, graph DP extends the node and edge perturbation approaches by quantifying their privacy loss. Day *et al.* [112] explored the graph degree distribution under node DP by designing a projection technique that incorporates aggregation and cumulative histogram to reduce the degree sensitivity. Huang *et al.* [113] developed a privacy protection algorithm using the adjacency degrees of a graph, which combined clustering and randomization. The privacy of the graph was protected by clustering, reconstructing the graph based on ordered degrees, and generating noisy nodes. Li *et al.* [114] proposed a general method for perturbing subspace to achieve privacy preservation and distributed optimization, where noises were injected into the non-convergent subspace using the dual variables to preserve the privacy information. Compared with k-anonymity, DP can protect the statistics of graph data, e.g., the degree distribution and edge weights.

Only a few studies have been dedicated to preserving the privacy of graph data with latent graph structures, more explicitly, the latent graph structures [115], [116]. He *et al.* [115] developed a privacy analysis framework with a noise-adding process for the average consensus algorithm, to limit the disclosure probability. Li *et al.* [116] addressed the privacy issues using a distributed graph filtering technique, which provides each node with the ability to compute its own desired output while maintaining its privacy. However, these methods [115], [116] only considered the privacy of graph data without taking into account the privacy of the latent graph structure underlying the graph data.

Chapter 3

Graph Extraction and Topology Learning of Band-limited Brain Signals

3.1 Introduction

Network neuroscience contributes to the comprehension of the architecture and functionality of the human brain by viewing a brain to be a complex network comprising many ROIs, also known as brain network nodes [86]. The connectivity and functional interactions among the ROIs serve a key role in brain-related cognitive function [117]. One example is ADHD, a widely observed and severely impairing neurodevelopmental behavioral problem diagnosed with inattention, hyperactivity, and impulsivity. ADHD affects school-age children and adolescents [33]. Latest neuroanatomical and neuropsychological studies indicate that these behavioral disturbances relate to atypical connectivity amongst brain functional area [118].

Graph theory has played a crucial role as a tool for analyzing intricate brain networks, and revealed several non-trivial features of brains, such as modularity and small-worldness, by studying the fMRI time series (i.e., blood-oxygen-level dependent time series) amongst the ROIs in a brain [86]. Pearson's correlation has been one of the most common methods for measuring the pairwise functional relationships between brain regions. However, Pearson's correlation focuses on strong direct marginal correlations in the fMRI data among two brain regions and overlooks the latent effects of other brain regions. Albert *et al.* [88] used partial correlations to measure interactions between any two ROIs. The partial correlation quantifies the dependency between two ROIs by regressing out the other ROIs, leading to difficulties in the suppression of the confounding effect from the other ROIs. SICE [91] has been another popular technique for measuring the intensity of the most significant direct connection between ROIs. SICE is a rigorous algorithm based on partial correlation. It tends to evaluate the sparsest reconstructive coefficient of each ROI and capture only local structures, rather than a representation of the global structure.

3.2 System Model

A whole brain network can be parcellated into a set of N nodes corresponding to different brain regions. Each node represents a specific region containing a fMRI time series. We assume that the i-th time series corresponding to N nodes refers to a vector, denoted by $\mathbf{y}_i \in \mathbb{R}^{N\times 1}$. M is the number of time course of brain signals. Let $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_M] \in \mathbb{R}^{N\times M}$ collect M observed signals of the brain network.

Brain networks describe the physical connectivity patterns between different brain regions. As considered in [35], our objective is to deduce the graph topology of brains from the time series $\mathbf{Y}$. Specifically, we wish to characterize the brain network with a weighted, undirected, and unidentified graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ consisting of N vertices. Every vertex corresponds to a brain network node. The edge between any two vertices indicates the physical proximity or relationship of the corresponding brain nodes. $\mathcal{V} = \{1, \cdots, N\}$ collects N vertices corresponding to specific brain regions. $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the collection of edges of the brain connectivity. The adjacent matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$ indicates to what extent two brain nodes are correlated. $W_{ij} = W_{ji} \neq 0$ for any brain regions $(i, j) \in \mathcal{E}$.

The combinatorial graph Laplacian $\mathbf{L}$ of the brain network $\mathcal{G}$ is defined as [72]:

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \tag{3.1}$$

where $\mathbf{D} \triangleq \operatorname{diag}(\mathbf{W1})$ defines the degree matrix containing the node degrees at its diagonal. $\mathbf{1}$ is an all-one vector. We assume that each brain region is connected to at least one other brain region, ensuring no isolated regions are in the brain topology. In other words, none of the diagonal elements is zero in $\mathbf{D}$.

The graph Laplacian is a semi-definite matrix with positive elements along its main diagonal and non-positive elements anywhere else [84]. By eigenvalue decomposition, $\mathbf{L}$ is rewritten as:

$$\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T, \tag{3.2}$$

where $\boldsymbol{\Lambda}$ denotes the diagonal matrix of non-negative Laplacian eigenvalues, and $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_N]$ represents the orthonormal matrix collecting all the eigenvectors.

To infer the topological knowledge of $\mathcal{G}$ requires the Laplacian matrix $\mathbf{L}$ to be estimated. As done in [35] and [119], we enforce the signals $\mathbf{Y}$ to be band-limited over graph $\mathcal{G}$, e.g., the signals are sparse in the canonical domain [35], [119], [120]. GFT [22], [72] has been utilized to decompose a brain time series into orthonormal components $\mathbf{U}$ in the Laplacian $\mathbf{L}$ [84]. For any $m \in \{1, \cdots, M\}$, the GFT of the time series $\mathbf{y}_m$, denoted by $\mathbf{s}_m$, projects $\mathbf{y}_m$ onto spectral-domain subspace spanned by $\mathbf{U}$, as given by

$$\mathbf{s}_m = \mathbf{U}^T\mathbf{y}_m. \tag{3.3}$$

With the band-limitedness of the observed signal $\mathbf{y}_m$, $\mathbf{s}_m$ is a sparse vector and captures the key characteristics of $\mathbf{y}_m$ in the frequency domain. The band-limited signal is written as

$$\mathbf{y}_m = \mathbf{U}\mathbf{s}_m. \tag{3.4}$$

Let $\mathbf{S} = [\mathbf{s}_1, \cdots, \mathbf{s}_M] \in \mathbb{R}^{N\times M}$ collect all $\mathbf{s}_m \in \mathbb{R}^{N\times 1}$, $m \in \{1, \cdots, M\}$. From (3.4), we have

$$\mathbf{Y} = \mathbf{U}\mathbf{S}. \tag{3.5}$$

With the sparsity of $\mathbf{s}_m, m \in \{1, \cdots, M\}$, we set $\mathbf{S} \in \mathcal{B}_K$ as a K-block sparse matrix with multiple all-zero row-vectors. K accounts for the bandwidth of the frequency-domain representation of the observed band-limited graph signals $\mathbf{Y}$, which can be obtained empirically in prior, or enumerated to find its proper value. $\mathcal{B}_K$ collects all

K-block sparse matrices [121]:

$$\mathcal{B}_K \triangleq \{\mathbf{S} \in\mathbb{R}^{N\times M}, \mathbf{S}(i,:) = \mathbf{0}, \forall i \notin \mathcal{K} \subseteq \mathcal{V},\ K = |\mathcal{K}|\}, \tag{3.6}$$

where $\mathbf{S}(i,:)$ is the i-th row of $\mathbf{S}$, and $\mathcal{K} \subseteq \mathcal{V}$ has the cardinality of K.

Algorithm 1: AO-based joint optimization method for solving problem (3.24)

```
**Initialization**: $\epsilon_1$, $\epsilon_2$, $\mu$, $\beta$, $\rho$, $\epsilon$;
**while** (3.24a) *is yet to converge with the accuracy of* $\epsilon$ **do**
    **Initialization:** Randomly initialize $\mathbf{V}_1', \mathbf{V}_2' \in \mathcal{M}$ as orthogonal matrices;
    **while** $\|\nabla_{\mathbf{V}_1}\mathcal{L}(\mathbf{V}_1)\|_F > \epsilon_1$ *and* $\|\nabla_{\mathbf{V}_2}\mathcal{L}(\mathbf{V}_2)\|_F > \epsilon_1$ **do**
        Update $\tau_1$ and $\tau_2$ and by (3.36) ;
        Compute $\tau_1 = \max(\min(\tau_1, \tau_{\max}), \tau_{\min})$;
        Compute $\tau_2 = \max(\min(\tau_2, \tau_{\max}), \tau_{\min})$;
        Update $\mathbf{V}_1$ and $\mathbf{V}_2$ by (3.34);
        $\mathbf{V}_1' \leftarrow \mathbf{V}_1$; $\mathbf{V}_2' \leftarrow \mathbf{V}_2$;
    **end**
    **Initialization:** Initialize $\mathbf{C}'$ and $\mathbf{Z}'$ as identity matrices;
    **while** $\|\mathbf{C} - \mathbf{C}'\| \,/\, \|\mathbf{C}'\| > \epsilon_2$, *and* $\|\mathbf{Z} - \mathbf{Z}'\| \,/\, \|\mathbf{Z}'\| > \epsilon_2$ **do**
        Update $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$, see (A.4);
        Update $\mathbf{C}$ by (A.6);
        Update $\mathbf{Z}$ by (A.7);
        $\mathbf{\Lambda}_1' \leftarrow \mathbf{\Lambda}_1$; $\mathbf{\Lambda}_2' \leftarrow \mathbf{\Lambda}_2$; $\mathbf{C}' \leftarrow \mathbf{C}$; $\mathbf{Z}' \leftarrow \mathbf{Z}$;
    **end**
**end**
```

3.3 Proposed Alternating Optimization for Graph Learning

We estimate the Laplacian matrix $\mathbf{L}$ and, in turn, the topological knowledge of graph $\mathcal{G}$ substantiating the observation $\mathbf{Y}$. Given the band-limitedness of $\mathbf{Y}$, we cast the problem as

$$\min_{\mathbf{L},\mathbf{U}\in\mathbb{R}^{N\times N},\mathbf{S}\in\mathbb{R}^{N\times M}} \|\mathbf{Y}-\mathbf{U}\mathbf{S}\|_F^2+\beta f(\mathbf{L},\mathbf{Y}) \tag{3.7a}$$

$$\text{s.t.} \quad \mathbf{U}\mathbf{U}^T=\mathbf{I}_N, \tag{3.7b}$$

$$\mathbf{S}\in\mathcal{B}_K, \tag{3.7c}$$

$$\mathbf{L}=\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T,\ \mathbf{L}\in\mathbb{L}, \tag{3.7d}$$

$$\mathbf{u}_1=\frac{1}{\sqrt{N}}\mathbf{1}. \tag{3.7e}$$

The first term of the objective (3.7a) accounts for data fidelity by incorporating a quadratic loss that penalizes any disparity among $\mathbf{US}$ and $\mathbf{Y}$. The second term of (3.7a) is a regularization function [18], [35]. β is an adjustable weighting coefficient of the regularizer.

Constraint (3.7b) ensures $\mathbf{U}$ to be a unitary matrix complying with (3.2). Constraint (3.7c) impose the K-block sparsity of $\mathbf{S}$ in (3.6). Constraint (3.7d) guarantees that $\mathbf{L}$ refers to a Laplacian matrix that satisfies the necessary conditions and properties, and $\mathbb{L}$ collects all valid candidates to $\mathbf{L}$ [84], i.e.,

$$\mathbb{L}=\{\mathbf{L}\succeq\mathbf{0}|\mathbf{L1}=\mathbf{0},L_{ij}=L_{ji}\leq 0,i\neq j\}, \tag{3.8}$$

where $\mathbf{0}$ refers to an all-zero vector. According to $\mathbf{L1}=\mathbf{0}$ in (3.8), we conclude that 0 is one eigenvalue of $\mathbf{L}$ corresponding to $\mathbf{u}_1=\frac{1}{\sqrt{N}}\mathbf{1}$, i.e., the first column of $\mathbf{U}$; see (3.7e).

Given the a-priori statistical knowledge of $\mathbf{L}$, the regularizer $f(\mathbf{L},\mathbf{Y})$ can effectively capture and reflect the intended characteristics of the graph topology. Suppose that $\mathbf{L}$ follows an exponential distribution [84]. The maximum a-posteriori (MAP) estimate of $\mathbf{L}$ is written as [84]

$$\min_{\mathbf{L}} \operatorname{tr}(\mathbf{L}\mathbf{Y}\mathbf{Y}^T)-\log|\mathbf{L}|+\alpha\left\|\operatorname{vec}(\mathbf{L})\right\|_1, \quad \text{s.t.} \quad \mathbf{L}\in\mathbb{L}. \tag{3.9}$$

In this sense, $f(\mathbf{L},\mathbf{Y}) = \mathrm{tr}(\mathbf{L}\mathbf{Y}\mathbf{Y}^T) - \log|\mathbf{L}| + \alpha\left\|\mathrm{vec}(\mathbf{L})\right\|_1$ provides a reasonable regularization in (3.7a). Here, $||\cdot||_1$ stands for the ℓ_1-norm and α is a tunable regularization parameter.

Based on the non-positivity and the structural constraint (i.e., $\mathbf{L}\mathbf{1} = \mathbf{0}$) of $\mathbf{L}$, we obtain $\left\|\mathrm{vec}(\mathbf{L})\right\|_1 = 2\mathrm{tr}(\mathbf{L})$. By using the linearity of trace, $f(\mathbf{L},\mathbf{Y})$ can be rewritten as

$$f(\mathbf{L},\mathbf{Y}) = \mathrm{tr}\left(\mathbf{L}\mathbf{Y}\mathbf{Y}^T + 2\alpha\mathbf{L}\right) - \log|\mathbf{L}| = \mathrm{tr}\left(\mathbf{L}\mathbf{T}\right) - \log|\mathbf{L}|, \tag{3.10}$$

where $\mathbf{T} = \mathbf{Y}\mathbf{Y}^T + 2\alpha\mathbf{I}$ for notational simplicity.

As $f(\mathbf{L},\mathbf{Y}) = \mathrm{tr}\left(\mathbf{L}\mathbf{T}\right) - \log|\mathbf{L}|$ is intractable due to the singular pseudo-determinant of $|\mathbf{L}|$, we exploit the equivalence of $\log|\mathbf{L}|$ and $\log\det(\mathbf{L}+\mathbf{J})$ with $\mathbf{J} = \mathbf{u}_1\mathbf{u}_1^T = \frac{1}{N}\mathbf{1}\mathbf{1}^T$ [84, Prop. 1] and rewrite $f(\mathbf{L},\mathbf{Y})$ as

$$f(\mathbf{L},\mathbf{Y}) = \mathrm{tr}\left(\mathbf{L}\mathbf{T}\right) - \log\det(\mathbf{L}+\mathbf{J}). \tag{3.11}$$

By exploiting the orthonormality of $\mathbf{U}$ in (3.7b), we have $\left\|\mathbf{Y}-\mathbf{U}\mathbf{S}\right\|_F^2 = \left\|\mathbf{U}^T\mathbf{Y}-\mathbf{S}\right\|_F^2$. Then, we can rewrite (3.7) as

$$\begin{aligned}\min_{\mathbf{L},\mathbf{U}\in\mathbb{R}^{N\times N},\mathbf{S}\in\mathbb{R}^{N\times M}} & \left\|\mathbf{U}^T\mathbf{Y}-\mathbf{S}\right\|_F^2 + \beta f(\mathbf{L},\mathbf{Y})\\ \text{s.t.} & \ \ (3.7\text{b}) - (3.7\text{e}).\end{aligned} \tag{3.12}$$

Problem (3.12) is non-convex because of the non-convex nature of the objective function $f(\mathbf{L},\mathbf{Y})$ and the coupling of the variables in (3.7a), the orthonormality in (3.7b), and the sparsity in (3.7c). Since both $\mathbf{U}$ and $\mathbf{S}$ are unknown, we reorganize (3.12) as

$$\begin{aligned}\min_{\mathbf{L},\mathbf{U}\in\mathbb{R}^{N\times N}} & \left(\min_{\mathbf{S}\in\mathcal{B}_K}\sum_{i=1}^{N}\left\|\mathbf{u}_i^T\mathbf{Y}-\mathbf{S}(i,:)\right\|_2^2 + \beta f(\mathbf{L},\mathbf{Y})\right)\\ \text{s.t.} & \quad (3.7\text{b}) - (3.7\text{e}),\end{aligned} \tag{3.13}$$

which, based on the definition of $\mathbf{S}$ (i.e., $\mathbf{S}(i,:) = \mathbf{0}$), can be further rewritten as

$$\begin{aligned}\min_{\mathbf{L},\mathbf{U}\in\mathbb{R}^{N\times N},\mathcal{K}} & \left(\min_{\mathbf{S}\in\mathcal{B}_K}\sum_{i\in\mathcal{K}}\left\|\mathbf{u}_i^T\mathbf{Y}-\mathbf{S}(i,:)\right\|_2^2 + \sum_{i\notin\mathcal{K}}\left\|\mathbf{u}_i^T\mathbf{Y}\right\|_2^2\right) + \beta f(\mathbf{L},\mathbf{Y}), \\ \text{s.t.} & \ \ (3.7\text{b}) - (3.7\text{e}).\end{aligned} \tag{3.14}$$

By closely assessing the objective of (3.14), we can find that the optimal index set, $\mathcal{K}$, collects the indices to the K largest entries of $\{\left\|\mathbf{u}_i^T\mathbf{Y}\right\|\}_i^N$, and satisfies

$$\mathbf{S}(i,:) = \begin{cases} \mathbf{u}_i^T\mathbf{Y}, & \text{if } \ i \in \mathcal{K}; \\ \mathbf{0}, & \text{if } \ i \notin \mathcal{K}. \end{cases} \tag{3.15}$$

As a result, only the $N-K$ smallest entries of $\{||\mathbf{u}_i^T\mathbf{Y}||\}_i^N$ remain in the objective of (3.14). The first term of (3.7a) is rewritten as

$$\|\mathbf{Y}-\mathbf{US}\|_F^2 = \quad \|\mathbf{Y}-\mathbf{U}_{\mathcal{K}}\mathbf{S}_{\mathcal{K}}\|_F^2 = \left\|\left(\mathbf{I}-\mathbf{U}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T\right)\mathbf{Y}\right\|_F^2, \tag{3.16}$$

where $\mathbf{U}_{\mathcal{K}}$ is the matrix collating the column-vectors of $\mathbf{U}$ indexed by $\mathcal{K}$, and $\mathbf{S}_{\mathcal{K}}$ is the matrix collating the row-vectors of $\mathbf{S}$ indexed by $\mathcal{K}$. The second component in the objective of problem (3.12), i.e., $f(\mathbf{L},\mathbf{Y})$, can be reformulated by considering the following two cases in regards to constraint (3.7e):

1. *In the case of* $\mathbf{u}_1 \notin \mathbf{U}_{\mathcal{K}}$*,* the eigenvectors of $\mathbf{L}$, i.e., $\mathbf{U}$, can be arranged as $[\mathbf{u}_1, \mathbf{U}_{\mathcal{K}}, \mathbf{U}_{\mathcal{K}^c\setminus\{1\}}]$. By performing eigenvalue decomposition, $\mathbf{L}$ can be formulated to

$$\begin{aligned}\mathbf{L} &= \left[\mathbf{u}_1, \mathbf{U}_{\mathcal{K}}, \mathbf{U}_{\mathcal{K}^c\setminus\{1\}}\right]\mathbf{\Phi}_1\left[\mathbf{u}_1, \mathbf{U}_{\mathcal{K}}, \mathbf{U}_{\mathcal{K}^c\setminus\{1\}}\right]^T \\ &= \left[\mathbf{U}_{\mathcal{K}}, \mathbf{U}_{\mathcal{K}^c\setminus\{1\}}\right]\mathbf{\Phi}_2\left[\mathbf{U}_{\mathcal{K}}, \mathbf{U}_{\mathcal{K}^c\setminus\{1\}}\right]^T.\end{aligned} \tag{3.17}$$

By plugging (3.17) and $\mathbf{J} = \frac{1}{N}\mathbf{1}\mathbf{1}^T$, $\log\det(\mathbf{L}+\mathbf{J})$ can be rewritten as

$$\begin{aligned}&\log\det(\mathbf{L}+\mathbf{J}) \\ &= \log\det\left(\left[\frac{\mathbf{1}}{\sqrt{N}}, \mathbf{U}_{\mathcal{K}}, \mathbf{U}_{\mathcal{K}^c\setminus\{1\}}\right]\begin{bmatrix}1 & \\ & \mathbf{\Phi}_2\end{bmatrix}\left[\frac{\mathbf{1}}{\sqrt{N}}, \mathbf{U}_{\mathcal{K}}, \mathbf{U}_{\mathcal{K}^c\setminus\{1\}}\right]^T\right) \\ &= \log\det\left(\text{blkdiag}\left(1, \mathbf{\Lambda}_{\mathcal{K}}, \mathbf{\Lambda}_{\mathcal{K}^c\setminus\{1\}}\right)\right) \\ &= \log\det(\mathbf{\Lambda}_{\mathcal{K}}) + \log\det\left(\mathbf{\Lambda}_{\mathcal{K}^c\setminus\{1\}}\right).\end{aligned} \tag{3.18}$$

Then, $f(\mathbf{L},\mathbf{Y})$ can be formulated to

$$\begin{aligned}f(\mathbf{L},\mathbf{Y}) = \ &\text{tr}\left(\mathbf{U}_{\mathcal{K}}\mathbf{\Lambda}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T\mathbf{T}\right) \\ &+ \text{tr}\left(\mathbf{U}_{\mathcal{K}^c\setminus\{1\}}\mathbf{\Lambda}_{\mathcal{K}^c\setminus\{1\}}\mathbf{U}_{\mathcal{K}^c\setminus\{1\}}^T\mathbf{T}\right) \\ &- \log\det(\mathbf{\Lambda}_{\mathcal{K}}) - \log\det\left(\mathbf{\Lambda}_{\mathcal{K}^c\setminus\{1\}}\right),\end{aligned} \tag{3.19}$$

where $\mathbf{\Phi}_1 = \text{blkdiag}\left(0, \mathbf{\Lambda}_{\mathcal{K}}, \mathbf{\Lambda}_{\mathcal{K}^c\setminus\{1\}}\right)$ with $\mathbf{\Lambda}_{\mathcal{K}} \succeq \mathbf{0}$ and $\mathbf{\Lambda}_{\mathcal{K}} \in \mathbb{R}^{K\times K}$ is a block-diagonal matrix and collects all the eigenvalues associated with the eigenvectors $[\mathbf{u}_1, \mathbf{U}_{\mathcal{K}}, \mathbf{U}_{\mathcal{K}^c\setminus\{1\}}]$ of $\mathbf{L}$; and $\mathbf{\Phi}_2 = \text{blkdiag}\left(\mathbf{\Lambda}_{\mathcal{K}}, \mathbf{\Lambda}_{\mathcal{K}^c\setminus\{1\}}\right)$ with $\mathbf{\Lambda}_{\mathcal{K}^c\setminus\{1\}} \succeq \mathbf{0}$ and $\mathbf{\Lambda}_{\mathcal{K}^c\setminus\{1\}} \in \mathbb{R}^{(N-K-1)\times(N-K-1)}$.

2. *In the case of* $\mathbf{u}_1 \in \mathbf{U}_{\mathcal{K}}$, $\mathbf{U}$ can be arranged as $[\mathbf{u}_1, \mathbf{U}_{\mathcal{K}\setminus\{1\}}, \mathbf{U}_{\mathcal{K}^c}]$. Then, (3.16) is rewritten as

$$\begin{aligned}\left\|\left(\mathbf{I} - \mathbf{U}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T\right)\mathbf{Y}\right\|_F^2 &= \left\|\left(\mathbf{I} - \mathbf{U}_{\mathcal{K}\setminus\{1\}}\mathbf{U}_{\mathcal{K}\setminus\{1\}}^T - \mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\right\|_F^2 \\ &\overset{(a)}{=} \text{tr}\Big[\left(\mathbf{I} - \mathbf{U}_{\mathcal{K}\setminus\{1\}}\mathbf{U}_{\mathcal{K}\setminus\{1\}}^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I} - \mathbf{U}_{\mathcal{K}\setminus\{1\}}\mathbf{U}_{\mathcal{K}\setminus\{1\}}^T\right)^T \\ &\quad - \mathbf{u}_1\mathbf{u}_1^T\mathbf{Y}\mathbf{Y}^T\mathbf{u}_1\mathbf{u}_1^T + \mathbf{u}_1\mathbf{u}_1^T\mathbf{U}_{\mathcal{K}\setminus\{1\}}\mathbf{U}_{\mathcal{K}\setminus\{1\}}^T\mathbf{Y}\mathbf{Y}^T\Big] \\ &\overset{(b)}{=} \left\|\left(\mathbf{I} - \mathbf{U}_{\mathcal{K}\setminus\{1\}}\mathbf{U}_{\mathcal{K}\setminus\{1\}}^T\right)\mathbf{Y}\right\|_F^2 - \left\|\mathbf{u}_1\mathbf{u}_1^T\mathbf{Y}\right\|_F^2, \end{aligned} \tag{3.20}$$

where $\mathbf{u}_1\mathbf{u}_1^T\mathbf{U}_{\mathcal{K}\setminus\{1\}}\mathbf{U}_{\mathcal{K}\setminus\{1\}}^T\mathbf{Y}\mathbf{Y}^T = \mathbf{0}$ in (a) since $\mathbf{u}_1^T\mathbf{U}_{\mathcal{K}\setminus\{1\}} = \mathbf{0}$, and $\mathbf{u}_1\mathbf{u}_1^T\mathbf{Y}$ is constant in (b).

Then, $\mathbf{L}$ can be rewritten as

$$\mathbf{L} = \left[\mathbf{u}_1, \mathbf{U}_{\mathcal{K}\setminus\{1\}}, \mathbf{U}_{\mathcal{K}^c}\right]\mathbf{\Phi}_1\left[\mathbf{u}_1, \mathbf{U}_{\mathcal{K}\setminus\{1\}}, \mathbf{U}_{\mathcal{K}^c}\right]^T = \left[\mathbf{U}_{\mathcal{K}\setminus\{1\}}, \mathbf{U}_{\mathcal{K}^c}\right]\mathbf{\Phi}_2\left[\mathbf{U}_{\mathcal{K}\setminus\{1\}}, \mathbf{U}_{\mathcal{K}^c}\right]^T. \tag{3.21}$$

Likewise, $f(\mathbf{L}, \mathbf{Y})$ is rewritten as

$$\begin{aligned} f(\mathbf{L}, \mathbf{Y}) &= \text{tr}\left(\mathbf{U}_{\mathcal{K}\setminus\{1\}}\mathbf{\Lambda}_{\mathcal{K}\setminus\{1\}}\mathbf{U}_{\mathcal{K}\setminus\{1\}}^T\mathbf{T}\right) + \text{tr}\left(\mathbf{U}_{\mathcal{K}}\mathbf{\Lambda}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T\mathbf{T}\right) \\ &\quad - \log\det\left(\mathbf{\Lambda}_{\mathcal{K}\setminus\{1\}}\right) - \log\det\left(\mathbf{\Lambda}_{\mathcal{K}}\right), \end{aligned} \tag{3.22}$$

where, with a slight abuse of notation, $\mathbf{\Phi}_1 = \text{blkdiag}\left(0, \mathbf{\Lambda}_{\mathcal{K}\setminus\{1\}}, \mathbf{\Lambda}_{\mathcal{K}^c}\right)$ with $\mathbf{\Lambda}_{\mathcal{K}\setminus\{1\}} \succeq \mathbf{0}$ and $\mathbf{\Lambda}_{\mathcal{K}\setminus\{1\}} \in \mathbb{R}^{(K-1)\times(K-1)}$; and $\mathbf{\Phi}_2 = \text{blkdiag}\left(\mathbf{\Lambda}_{\mathcal{K}\setminus\{1\}}, \mathbf{\Lambda}_{\mathcal{K}^c}\right)$ with $\mathbf{\Lambda}_{\mathcal{K}^c} \succeq \mathbf{0}$ with $\mathbf{\Lambda}_{\mathcal{K}^c} \in \mathbb{R}^{(N-K)\times(N-K)}$.

To unify the presentation and the follow-on discussions of the two cases, we define

1. If $\mathbf{u}_1 \notin \mathbf{U}_{\mathcal{K}}$, then $\mathbf{V}_1 = \mathbf{U}_{\mathcal{K}}$, $\mathbf{V}_2 = \mathbf{U}_{\mathcal{K}^c\setminus\{1\}}$, $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_{\mathcal{K}}, \mathbf{\Lambda}_2 = \mathbf{\Lambda}_{\mathcal{K}^c\setminus\{1\}}, \mathbf{I}_1 \in \mathbb{R}^{K\times K}$, and $\mathbf{I}_2 \in \mathbb{R}^{(N-K-1)\times(N-K-1)}$;

2. If $\mathbf{u}_1 \in \mathbf{U}_{\mathcal{K}}$, then $\mathbf{V}_1 = \mathbf{U}_{\mathcal{K}\setminus\{1\}}$, $\mathbf{V}_2 = \mathbf{U}_{\mathcal{K}^c}$, $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_{\mathcal{K}\setminus\{1\}}, \mathbf{\Lambda}_2 = \mathbf{\Lambda}_{\mathcal{K}^c}, \mathbf{I}_1 \in \mathbb{R}^{(K-1)\times(K-1)}$, and $\mathbf{I}_2 \in \mathbb{R}^{(N-K)\times(N-K)}$.

As a result, $\mathbf{L}$ can be represented as

$$\mathbf{L} = [\mathbf{V}_1, \mathbf{V}_2]\, \mathbf{\Phi}_2\, [\mathbf{V}_1, \mathbf{V}_2]^T . \tag{3.23}$$

Since $[\mathbf{V}_1, \mathbf{V}_2]^T \mathbf{u}_1 = \mathbf{0}$, $\mathbf{L1} = \mathbf{0}$ in (3.8) is preserved in (3.23).

Finally, with the unified presentation (3.23) capturing both cases of $\mathbf{u}_1 \in \mathcal{K}$ and $\mathbf{u}_1 \notin \mathcal{K}$, problem (3.7) is transformed equivalently to

$$\begin{aligned}
\min_{\mathbf{V}_1,\mathbf{V}_2,\mathbf{\Lambda}_1,\mathbf{\Lambda}_2} \quad & \left\| \left(\mathbf{I} - \mathbf{V}_1\mathbf{V}_1^T\right) \mathbf{Y} \right\|_F^2 + \beta \left[\operatorname{tr}\left(\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T\mathbf{T}\right) + \right. \\
& \left. \operatorname{tr}\left(\mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T\mathbf{T}\right) - \log\det\left(\mathbf{\Lambda}_1\right) - \log\det\left(\mathbf{\Lambda}_2\right) \right] && \text{(3.24a)} \\
\text{s.t.} \quad & \mathbf{V}_1^T\mathbf{V}_1 = \mathbf{I}_1, \mathbf{V}_2^T\mathbf{V}_2 = \mathbf{I}_2, \mathbf{V}_1^T\mathbf{V}_2 = \mathbf{0}, && \text{(3.24b)} \\
& \mathbf{u}_1^T\mathbf{V}_1 = \mathbf{0}, \mathbf{u}_1^T\mathbf{V}_2 = \mathbf{0}, && \text{(3.24c)} \\
& \mathbf{\Lambda}_1 \succeq \mathbf{0}, \mathbf{\Lambda}_2 \succeq \mathbf{0}, && \text{(3.24d)} \\
& \mathbf{I} \odot \left[\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T\right] \geq \mathbf{0}, && \text{(3.24e)} \\
& \mathbf{A} \odot \left[\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T\right] \leq \mathbf{0}, && \text{(3.24f)}
\end{aligned}$$

where (3.24b) is from (3.7b), and (3.24c) –(3.24f) correspond to (3.8).

Next, AO is carried out to iteratively solve problem (3.24) by optimizing $[\mathbf{V}_1, \mathbf{V}_2]$ and $[\mathbf{\Lambda}_1, \mathbf{\Lambda}_2]$ in an alternating manner. Finally, given the convergent $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$, we infer the Laplacian matrix $\mathbf{L}$ using (3.23).

3.3.1 Estimation of $\mathbf{V}_1$ and $\mathbf{V}_2$

Given $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$, problem (3.24) is reduced to

$$\begin{aligned}
\min_{\mathbf{V}_1,\mathbf{V}_2} \quad & \left\| \left(\mathbf{I} - \mathbf{V}_1\mathbf{V}_1^T\right) \mathbf{Y} \right\|_F^2 + \beta \left[\operatorname{tr}\left(\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T\mathbf{T}\right) + \operatorname{tr}\left(\mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T\mathbf{T}\right) \right] && \text{(3.25a)} \\
\text{s.t.} \quad & \mathbf{V}_1^T\mathbf{V}_1 = \mathbf{I}_1, \mathbf{V}_2^T\mathbf{V}_2 = \mathbf{I}_2, \mathbf{V}_1^T\mathbf{V}_2 = \mathbf{0}, && \text{(3.25b)} \\
& \mathbf{u}_1^T\mathbf{V}_1 = \mathbf{0}, \mathbf{u}_1^T\mathbf{V}_2 = \mathbf{0}, && \text{(3.25c)}
\end{aligned}$$

which has orthonormal vector variables (or in other words, the feasible solution region is on the Stiefel manifold) and therefore can be solved using Stiefel manifold dual gradient descent [122], as described below.

The Lagrangian function of (3.25) is defined as

$$\begin{aligned}
&\mathcal{L}\left(\mathbf{V}_1, \mathbf{V}_2, \boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2, \boldsymbol{\Psi}_3, \boldsymbol{\Psi}_4, \boldsymbol{\Psi}_5\right) \\
&= \left\|\left(\mathbf{I} - \mathbf{V}_1\mathbf{V}_1^T\right)\mathbf{Y}\right\|_F^2 + \beta \mathrm{tr}\left(\mathbf{V}_1\boldsymbol{\Lambda}_1\mathbf{V}_1^T\mathbf{T}\right) \\
&\quad + \beta \mathrm{tr}\left(\mathbf{V}_2\boldsymbol{\Lambda}_2\mathbf{V}_2^T\mathbf{T}\right) - \frac{1}{2}\mathrm{tr}(\boldsymbol{\Psi}_1^T(\mathbf{V}_1^T\mathbf{V}_1 - \mathbf{I}_1)) \\
&\quad - \frac{1}{2}\mathrm{tr}(\boldsymbol{\Psi}_2^T(\mathbf{V}_2^T\mathbf{V}_2 - \mathbf{I}_2)) - \frac{1}{2}\mathrm{tr}(\boldsymbol{\Psi}_3^T\mathbf{V}_1^T\mathbf{V}_2) \\
&\quad - \frac{1}{2}\mathrm{tr}(\boldsymbol{\Psi}_4^T\mathbf{u}_1^T\mathbf{V}_1) - \frac{1}{2}\mathrm{tr}(\boldsymbol{\Psi}_5^T\mathbf{u}_1^T\mathbf{V}_2),
\end{aligned} \tag{3.26}$$

where $\boldsymbol{\Psi}_1, \cdots, \boldsymbol{\Psi}_5$ are the Lagrange multipliers corresponding to the five conditions in (3.24).

Considering the Karush–Kuhn–Tucker (KKT) conditions, we differentiate (3.26) regarding $\mathbf{V}_1$ and then set the result to zero, as given by

$$\nabla_{\mathbf{V}_1}\mathcal{L} = \nabla\mathcal{F}(\mathbf{V}_1) - \mathbf{V}_1\boldsymbol{\Psi}_1 - \frac{1}{2}\mathbf{V}_2\boldsymbol{\Psi}_3 - \frac{1}{2}\mathbf{u}_1\boldsymbol{\Psi}_4 = \mathbf{0}, \tag{3.27}$$

where $\mathcal{F}(\mathbf{V}_1, \mathbf{V}_2) = \left\|\left(\mathbf{I} - \mathbf{V}_1\mathbf{V}_1^T\right)\mathbf{Y}\right\|_F^2 + \beta\ \mathrm{tr}\left(\mathbf{V}_1\boldsymbol{\Lambda}_1\mathbf{V}_1^T\mathbf{T}\right) + \beta\ \mathrm{tr}\left(\mathbf{V}_2\boldsymbol{\Lambda}_2\mathbf{V}_2^T\mathbf{T}\right)$.

By left multiplying $\mathbf{V}_1^T$ to both sides of (3.27), we have

$$\mathbf{V}_1^T\nabla\mathcal{F}(\mathbf{V}_1) - \mathbf{V}_1^T\mathbf{V}_1\boldsymbol{\Psi}_1 - \frac{1}{2}\mathbf{V}_1^T\mathbf{V}_2\boldsymbol{\Psi}_3 - \frac{1}{2}\mathbf{V}_1^T\mathbf{u}_1\boldsymbol{\Psi}_4 = \mathbf{0}. \tag{3.28}$$

Considering (3.24b), we obtain

$$\boldsymbol{\Psi}_1 = \mathbf{V}_1^T\nabla\mathcal{F}(\mathbf{V}_1). \tag{3.29}$$

By left multiplying $\mathbf{V}_2^T$ to both sides of (3.27) and applying (3.24b), we have

$$\boldsymbol{\Psi}_3 = 2\nabla\mathcal{F}(\mathbf{V}_1)^T\mathbf{V}_2. \tag{3.30}$$

By left multiplying $\mathbf{u}_1^T$ to both sides of (3.27) and applying (3.24b), we have

$$\boldsymbol{\Psi}_4 = 2\mathbf{u}_1^T\nabla\mathcal{F}(\mathbf{V}_1). \tag{3.31}$$

Since the constraint $\mathbf{V}_1^T\mathbf{V}_1 = \mathbf{I}_1$ in (3.25) is symmetric, $\boldsymbol{\Psi}_1$ is symmetric and therefore $\boldsymbol{\Psi}_1 = \nabla\mathcal{F}(\mathbf{V}_1)^T\mathbf{V}_1$. The gradient in (3.27) can be reformulated to

$$\nabla_{\mathbf{V}_1}\mathcal{L} = \nabla\mathcal{F}(\mathbf{V}_1) - \mathbf{V}_1\nabla\mathcal{F}(\mathbf{V}_1)^T\mathbf{V}_1 - \frac{1}{2}\mathbf{V}_2\mathbf{V}_2^T\nabla\mathcal{F}(\mathbf{V}_1) - \frac{1}{2}\mathbf{u}_1\mathbf{u}_1^T\nabla\mathcal{F}(\mathbf{V}_1). \tag{3.32}$$

Likewise,

$$\mathbf{\Psi}_2 = 2\mathbf{V}_2^T \nabla \mathcal{F}(\mathbf{V}_2); \mathbf{\Psi}_5 = 2\mathbf{u}_1^T \nabla \mathcal{F}(\mathbf{V}_2). \tag{3.33}$$

The gradient of the Lagrangian regarding $\mathbf{V}_2$ can be obtained by swapping $\mathbf{V}_1$ and $\mathbf{V}_2$ in (3.32).

With the Stiefel manifold dual gradient descent, problem (3.25) can be solved by iteratively updating the Lagrange multipliers with (3.29)–(3.31) and (3.33) and the variables $\mathbf{V}_1$ and $\mathbf{V}_2$ with the right-hand scaled gradient projection method [122]:

$$\mathbf{V}_1 \leftarrow \pi(\mathbf{V}_1 - \tau_1 \nabla_{\mathbf{V}_1} \mathcal{L}(\mathbf{V}_1) \mathcal{A}_1(\mu, \tau_1)), \tag{3.34}$$

where $\pi(\cdot)$ is the projection operator, i.e., $\pi(\mathbf{V}_1) = \mathbf{U}\mathbf{I}_1\mathbf{V}^T$ if $\mathbf{V}_1 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ by singular value decomposition (SVD) [122]; $\mathcal{A}_1(\mu, \tau_1)$ is a scaling matrix with $\mu \in (0, 1)$, i.e.,

$$\mathcal{A}_1(\mu, \tau_1) = (\mathbf{I}_1 + \mu\tau_1 \mathbf{V}_1^T \nabla_{\mathbf{V}_1} \mathcal{L}(\mathbf{V}_1))^{-1}, \tag{3.35}$$

and τ_1 is the step size and given by

$$\tau_1 = \begin{cases} \frac{\|\mathbf{V}_1 - \mathbf{V}_1'\|_F^2}{\left\langle \mathbf{V}_1 - \mathbf{V}_1', \nabla_{\mathbf{V}_1} \mathcal{L}(\mathbf{V}_1) - \nabla_{\mathbf{V}_1} \mathcal{L}(\mathbf{V}_1') \right\rangle}, & \text{in odd-numbered iterations,} \\ \frac{\left\langle \mathbf{V}_1 - \mathbf{V}_1', \nabla_{\mathbf{V}_1} \mathcal{L}(\mathbf{V}_1) - \nabla_{\mathbf{V}_1} \mathcal{L}(\mathbf{V}_1') \right\rangle}{\|\nabla_{\mathbf{V}_1} \mathcal{L}(\mathbf{V}_1) - \nabla_{\mathbf{V}_1} \mathcal{L}(\mathbf{V}_1')\|_F^2}, & \text{in even-numbered iterations.} \end{cases} \tag{3.36}$$

Here, $\mathbf{V}_1'$ is the counterpart of $\mathbf{V}_1$ at the previous iteration. $\tau_1 \in [\tau_{\min}, \tau_{\max}]$ with $\tau_{\min} = 10^{-5}$ and $\tau_{\max} = 10^5$ being the minimum and maximum values of step-sizes, respectively.

The right-hand scaled gradient projection of $\mathbf{V}_2$ can be obtained by replacing $\mathbf{V}_1$ and $\mathcal{A}_1(\mu, \tau_1)$ with $\mathbf{V}_2$ and $\mathcal{A}_2(\mu, \tau_2)$, respectively, in (3.34). Here, the scaling matrix $\mathcal{A}_2(\mu, \tau_2)$ can be obtained by replacing $\mathbf{I}_1$ and $\mathbf{V}_1$ with $\mathbf{I}_2$ and $\mathbf{V}_2$, respectively, in (3.35). The step-size $\tau_2 \in [\tau_{\min}, \tau_{\max}]$ can be obtained by replacing $\mathbf{V}_1$ and $\mathbf{V}_1'$ with $\mathbf{V}_2$ and $\mathbf{V}_2'$, respectively, in (3.36). $\mathbf{V}_2'$ is the counterpart of $\mathbf{V}_2$ in the previous iteration.

3.3.2 Estimation of $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$

Given $[\mathbf{V}_1, \mathbf{V}_2]$, problem (3.24) is rewritten as

$$\begin{aligned}
\min_{\mathbf{\Lambda}_1,\mathbf{\Lambda}_2} & \operatorname{tr}\left(\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T\mathbf{T}\right) + \operatorname{tr}\left(\mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T\mathbf{T}\right) - \log\det\left(\mathbf{\Lambda}_1\right) - \log\det\left(\mathbf{\Lambda}_2\right) \\
\text{s.t.} \quad & \mathbf{\Lambda}_1 \succeq \mathbf{0}, \mathbf{\Lambda}_2 \succeq \mathbf{0}, \\
& \mathbf{I} \odot [\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T] \geq \mathbf{0}, \\
& \mathbf{A} \odot [\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T] \leq \mathbf{0}.
\end{aligned} \tag{3.37}$$

Since both $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ may have zero diagonal elements, problem (3.37) is not continually differentiable and cannot be solved using CVX toolbox. By referring to [123], we introduce $\mathbf{C} = \mathbf{L}$ and rewrite the problem (3.37) as

$$\begin{aligned}
\min_{\mathbf{\Lambda}_1,\mathbf{\Lambda}_2} \quad & \left[\operatorname{tr}\left(\mathbf{\Lambda}_1\widetilde{\mathbf{T}}_1\right) + \operatorname{tr}\left(\mathbf{\Lambda}_2\widetilde{\mathbf{T}}_2\right) - \log\det\left(\mathbf{\Lambda}_1\right) - \log\det\left(\mathbf{\Lambda}_2\right)\right] \\
\text{s.t.} \quad & \mathbf{\Lambda}_1 \succeq \mathbf{0}, \mathbf{\Lambda}_2 \succeq \mathbf{0}, \\
& \mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T - \mathbf{C} = \mathbf{0}, \\
& \mathbf{I} \odot \mathbf{C} \geq \mathbf{0}, \\
& \mathbf{A} \odot \mathbf{C} \leq \mathbf{0},
\end{aligned} \tag{3.38}$$

which can be addressed by utilizing ADMM; see Appendix A.

Algorithm 1 summarizes the proposed AO-based algorithm that solves problem (3.24). The algorithm runs until the objective of (3.24) converges with the accuracy of ϵ. Each of the AO iterations starts by running the Stiefel manifold dual gradient descent till convergence, followed by the ADMM with $\mathbf{\Lambda}_1'$, $\mathbf{\Lambda}_2'$, $\mathbf{C}'$, and $\mathbf{Z}'$ being the results of $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$, $\mathbf{C}$, and $\mathbf{Z}$ in the previous iteration. Their respective convergence criteria are

$$\|\nabla_{\mathbf{V}_1}\mathcal{L}(\mathbf{V}_1)\|_F \leq \epsilon_1 \text{ and } \|\nabla_{\mathbf{V}_2}\mathcal{L}(\mathbf{V}_2)\|_F \leq \epsilon_1; \tag{3.39}$$

$$\|\mathbf{C} - \mathbf{C}'\| \,/\, \|\mathbf{C}'\| \leq \epsilon_2 \text{ and } \|\mathbf{Z} - \mathbf{Z}'\| \,/\, \|\mathbf{Z}'\| \leq \epsilon_2, \tag{3.40}$$

where ϵ_1 and ϵ_2 are the predefined thresholds. For example, in our experiments, we take $\epsilon = \epsilon_1 = \epsilon_2 = 10^{-4}$; see Section 3.4.

3.3.3 Complexity Analysis

In each iteration of the Stiefel manifold dual gradient descent, the computational cost of evaluating $\mathbf{V}_1$ is governed by the evaluation of $\nabla\mathcal{L}(\mathbf{V}_1)$ and $\mathcal{A}_1(\mu,\tau)$, which is $\mathcal{O}(N^2K+K^3)$. Likewise, the computational cost of evaluating $\mathbf{V}_2$ is governed by the evaluation of $\nabla\mathcal{L}(\mathbf{V}_2)$ and $\mathcal{A}_2(\mu,\tau)$, which is $\mathcal{O}(NK^2+(N-K-1)^3)$. The total computational complexity of the Stiefel manifold dual gradient descent is $\mathcal{O}(\log(1/\epsilon_1)(N^2K+NK^2+K^3+(N-K-1)^3))$. In most cases, $K<N$ and the total complexity is $\mathcal{O}\left(\log(1/\epsilon_1)N^3\right)$.

In each iteration of the ADMM, the computational cost of evaluating primal variable $\mathbf{\Lambda}_1$ is governed by the matrix multiplication and eigenvalue decomposition with $\mathcal{O}(K^3)$. Likewise, the complexity of evaluating the primal variable $\mathbf{\Lambda}_2$ is $\mathcal{O}\big((N-K-1)^3\big)$. The cost of evaluating the primal variable $\mathbf{C}$ is dominated by matrix multiplication with a complexity of $\mathcal{O}\big(N^2K+N^2(N-K-1)\big)$. The derivation of the dual variable $\mathbf{Z}$ incurs the complexity of $\mathcal{O}\big(N^2K+N^2(N-K-1)\big)$. Since $K<N$, the complexity of the ADMM is $\mathcal{O}(N^3)$ per iteration. The total computational cost of the ADMM is $\mathcal{O}\big(\log(1/\epsilon_2)N^3\big)$. As a result, the collective computational cost of the joint optimization is $\mathcal{O}\Bigg(N^3\log(1/\epsilon)\big[\log(1/\epsilon_1)+\log(1/\epsilon_2)\big]\Bigg)$, where ϵ denotes the desired level of convergence.

3.4 Numerical Results

This section conducts the experiments to gauge Algorithm 1. The experiments are carried out on a laptop with an i7-8650U CPU and 16G RAM. Algorithm 1 is initialized by setting $\mathbf{C}^{(0)}$ and $\mathbf{Z}^{(0)}$ as two symmetric unit matrices and the Lagrange multiplier ρ to 1. We stop the algorithm after 10^4 iterations or when the difference of the objective (3.24) is smaller than 10^{-5} between two consecutive iterations.

Apart from Algorithm 1, we assess the following advanced solutions in the literature: Dong's algorithm[18], Kalofolias' algorithm [19], Sardellitti's TV algorithm [35], Sardellitti's ESA algorithm [35], and Humbert's algorithm [94].

With reference to [35], we consider the performance metrics, including F-measure, Recall, Precision scores, and the percentage of recovery errors. The ground-truth graph set is denoted by $\mathcal{E}_g$, while the set of recovered graphs is represented as $\mathcal{E}_r$. Precision measures the proportion of edges correctly recovered in the graphs compared to the total number of edges in the recovered graphs, i.e., Precision $= \mathcal{E}_g \cap \mathcal{E}_r / \mathcal{E}_r$. Recall measures the proportion of edges correctly recovered in the ground-truth graphs compared to the total number of edges in the ground-truth graphs, i.e., Recall $= \mathcal{E}_g \cap \mathcal{E}_r / \mathcal{E}_g$.

F-measure is calculated as the harmonic mean of Recall and Precision, providing a measure of the overall accuracy in recovering the edges:

$$\text{F-measure} = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3.41}$$

The correlation coefficient $\rho_{\mathbf{W}}(\mathbf{W}_0, \mathbf{W})$ (or $\rho_{\mathbf{W}}$ for brevity) between a connected ground-truth graph and its corresponding recovered graph is defined as [35]

$$\rho_{\mathbf{W}}\left(\mathbf{W}_0, \mathbf{W}\right) = \frac{\sum_{ij} W_{0ij} W_{ij}}{\sqrt{\sum_{ij} W_{0ij}^2}\sqrt{\sum_{ij} W_{ij}^2}}, \tag{3.42}$$

where $\mathbf{W}_0$ and $\mathbf{W}$ are generated from (3.1) and give the weighted adjacent matrices of the ground-truth and recovered graphs, respectively. The coefficient correlation measures the correlation between the ground-truth graph and its corresponding recovered graph. The recovery accuracy can be enhanced with a higher coefficient value of $\rho_{\mathbf{W}}$.

The estimation error, or "Error" for short, is defined as [35]

$$\text{Error} = \frac{\|\mathbf{A} - \mathbf{A}_0\|_F}{N(N-1)}, \tag{3.43}$$

where $\mathbf{A}$ and $\mathbf{A}_0$ are the binary adjacency matrices of the recovered graphs and the ground-truth graphs, respectively.

3.4.1 Results on Synthetic Data

We consider three synthetic random graphs generated separately from Random Geometric model [124], Erdős-Rènyi model [125], and Barábasi-Albert model [126]. We generate the Random Geometric model with six connections per node. We set the connection probability to 0.3 to generate the Erdős-Rènyi graph. For the Barábasi-Albert model, we generate the graph with two initial nodes and then add more nodes with two edges for each of the nodes added. We generate over 100 independent random graphs for each of the three graph models. The average results obtained from these graphs are then plotted.

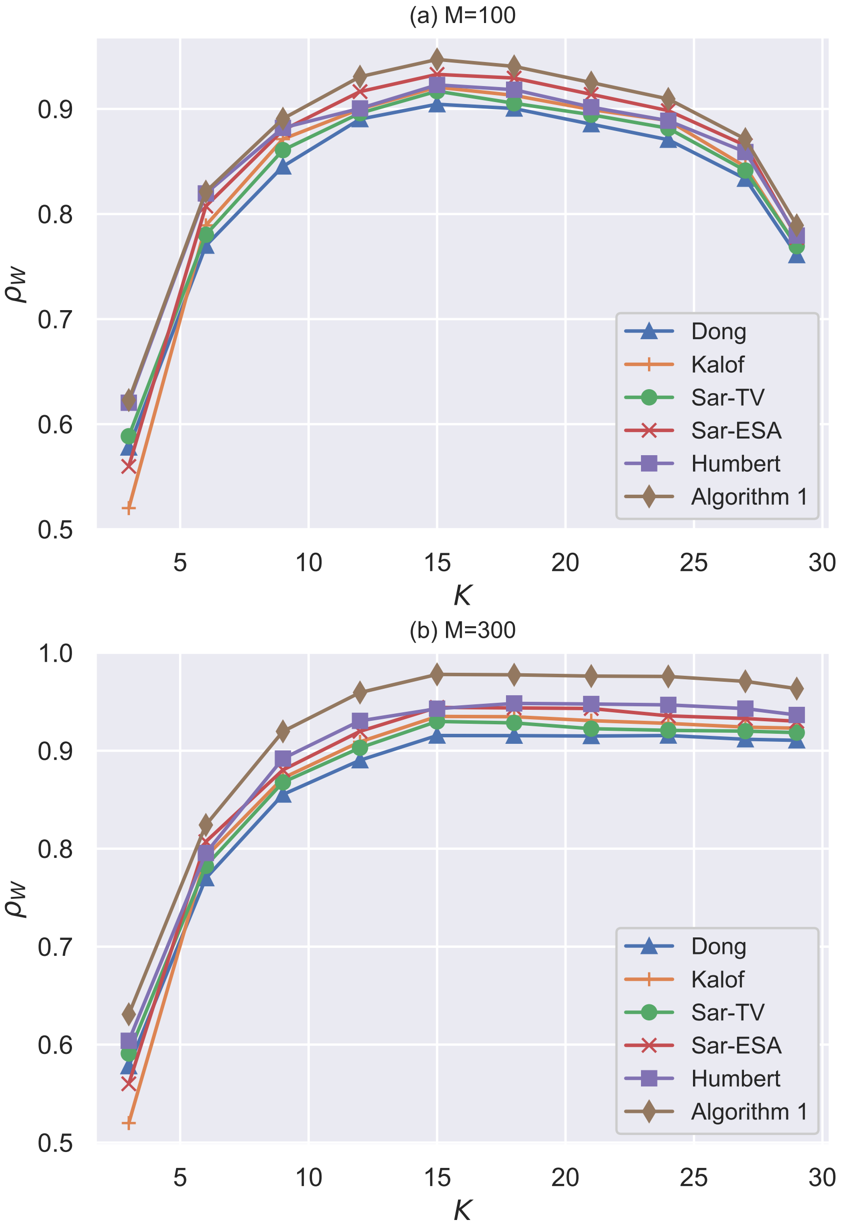


Figure 3.1: The correlation coefficient with an increasing signal bandwidth K under the Random Geometric model, where $N = 30$.

After generating a graph based on the above three models, we can obtain the ground-truth Laplacian of the graph, denoted by $\mathbf{L}_0$. We take the SVD of $\mathbf{L}_0$ and obtain the ground-truth GFT, denoted by $\mathbf{U}_0$. Then, we generate the observed band-limited graph signals $\mathbf{Y} = \mathbf{U}_0\mathbf{S}_0 + \mathbf{\Gamma}$ with the frequency-domain signals $\mathbf{S}_0 = [\mathbf{s}_{0,1}, \cdots, \mathbf{s}_{0,M}] \in \mathbb{R}^{N\times M}$ and the additive noises $\mathbf{\Gamma} = [\boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_M] \in \mathbb{R}^{N\times M}$ by following [119]. Specifically, $\mathbf{s}_{0,m} \sim \mathcal{N}(0, \mathbf{\Lambda}_K^{\dagger})$, where $\mathrm{diag}(\mathbf{\Lambda}_K) = (\lambda_1, \cdots, \lambda_K, 0, \cdots, 0)$. The precision matrix of $\mathbf{s}_{0,m}$ is defined to collect the eigenvalues with the largest $(N-K)$ values set to zero, as in [119]. $\boldsymbol{\gamma}_m \sim \mathcal{N}(0, \sigma^2\mathbf{I}_N)$ is the zero-mean multivariate Gaussian noise and $\sigma^2 = 0.1$ by default. The key difference between the band-limited signals and smooth signals is the degenerate values in some dimensions of the multivariate Gaussian signal $\mathbf{S}$. The smooth signals can be considered as a specific scenario of the band-limited signals with $K = N$. In this sense, our proposed algorithm designed for band-limited signals can be applied to smooth signals.

Tables 3.1 and 3.2 provide the comparison studies of Algorithm 1 and the benchmarks under the considered synthetic graph models, where each of the random graphs consists of $N = 30$ nodes and observes $M = 300$ time series. We take $K = 3$ in TABLE 3.1 and $K = 15$ in TABLE 3.2; in other words, the algorithms assume the number of significant components underlying the observed signals $\mathbf{Y}$ is 3 or 15. The regularization parameter α is optimized by trying and testing for each of the considered algorithms.

The tables show that Algorithm 1 is, in general, the best among all the considered techniques, and achieves high scores in both F-measure and correlation coefficient $\rho_{\mathbf{W}}$ under all random graph models. In particular, TABLE 3.1 shows that Algorithm 1 yields at least 20% improvement in F-measure under the Erdős-Rènyi model, as compared to Sardellitti's TV and ESA algorithms. By comparing Tables 3.1 and 3.2, we see that Algorithm 1 increasingly outperforms the benchmark techniques, as K grows from 3 to 15. In this sense, the algorithm is responsive to the signal bandwidth K, and increasingly improves with K.

Fig. 3.1 evaluates the average correlation coefficients of the considered algorithms against the signal bandwidth K under the Random Geometric model, where $N = 30$, $M = 100$ in Fig. 3.1(a) and $M = 300$ in Fig. 3.1(b). Our algorithm is shown to achieve the largest correlation coefficient among all algorithms over the wide spectrum of K. We see in Fig. 3.1(a) that the correlation coefficient first increases with

Table 3.1: Comparison of the considered algorithms, incl. Algorithm 1, Dong [18], Kalofolias [19], Sar-TV [35], Sar-ESA [35], and Humbert [94], where $K = 3$, $N = 30$, and $M = 300$.

	[18]	**[19]**	**[35]**	**[35]**	**[94]**	**Algo. 1**
RG						
F-measure	**0.5478 (±0.022)**	0.5537 (±0.047)	0.5536 (±0.031)	0.5255 (±0.046)	0.5078 (±0.029)	0.5601 (±0.052)
Recall	0.4500 (±0.023)	0.4408 (±0.026)	**0.4627 (±0.019)**	0.3913 (±0.035)	0.3844 (±0.056)	0.4123 (±0.017)
Precision	0.7000 (±0.013)	0.7444 (±0.024)	0.6889 (±0.031)	0.8000 (±0.018)	0.7422 (±0.025)	**0.8731 (±0.033)**
$\rho_{\mathbf{W}}$	0.6324 (±0.022)	0.5179 (±0.040)	0.5996 (±0.025)	0.5600 (±0.023)	0.6002 (±0.032)	**0.6452 (±0.035)**
Error	0.0189 (±0.021)	**0.0169 (±0.016)**	0.0220 (±0.011)	0.0219 (±0.013)	0.0232 (±0.014)	0.0156 (±0.012)
ER						
F-measure	0.4632 (±0.021)	0.4902 (±0.036)	0.3784 (±0.022)	0.3033 (±0.019)	0.3589 (±0.024)	0.4576 (±0.043)
Recall	**0.5000 (±0.022)**	0.4902 (±0.013)	0.2612 (±0.016)	0.2200 (±0.026)	0.2355 (±0.031)	0.3583 (±0.026)
Precision	0.4314 (±0.018)	0.4902 (±0.016)	0.6863 (±0.023)	0.6275 (±0.017)	0.7543 (±0.022)	0.6330 (±0.026)
$\rho_{\mathbf{W}}$	0.5611 (±0.017)	0.5776 (±0.019)	0.4623 (±0.023)	0.4496 (±0.034)	0.5445 (±0.023)	0.6062 (±0.028)
Error	0.0170 (±0.010)	0.0199 (±0.009)	0.0174 (±0.011)	0.0197 (±0.008)	0.018 (±0.019)	**0.0158 (±0.008)**
BA						
F-measure	**0.5056 (±0.023)**	0.5000 (±0.016)	0.3784 (±0.025)	0.3474 (±0.031)	0.3292 (±0.033)	0.4825 (±0.020)
Recall	**0.3980 (±0.028)**	0.3958 (±0.019)	0.2612 (±0.013)	0.2357 (±0.033)	0.2147 (±0.029)	0.3331 (± 0.022)
Precision	0.6946 (±0.034)	0.6786 (±0.026)	0.6863 (±0.027)	0.6607 (±0.021)	0.7054 (±0.033)	**0.8749 (±0.019)**
$\rho_{\mathbf{W}}$	0.4487 (±0.017)	0.4486 (±0.035)	0.4221 (±0.027)	0.3054 (±0.015)	0.4078 (±0.023)	**0.5127 (±0.027)**
Error	**0.0142 (±0.011)**	0.0143 (±0.009)	0.0177 (±0.013)	0.0192 (±0.011)	0.0207 (±0.007)	0.0225 (±0.015)

Table 3.2: Comparison of the considered algorithms, incl. Algorithm 1, Dong [18], Kalofolias [19], Sar-TV [35], Sar-ESA [35], and Humbert [94], where $K = 15$, $N = 30$, and $M = 300$.

RG						
F-measure	0.8517 (±0.021)	0.9135 (±0.036)	0.7946 (±0.019)	0.6642 (±0.025)	0.9154 (±0.018)	**0.9441 (±0.027)**
Recall	0.7479 (±0.022)	0.8432 (±0.018)	0.6666 (±0.023)	0.4972 (±0.021)	0.8442 (±0.019)	**0.8954 (±0.031)**
Precision	0.9889 (±0.018)	0.9967 (±0.032)	0.9889 (±0.009)	0.9999 (±0.0001)	0.9999 (±0.0001)	0.9984 (±0.001)
$\rho_{\mathbf{W}}$	0.9156 (±0.017)	0.9350 (±0.013)	0.9299 (±0.0171)	0.9442 (±0.016)	0.9203 (±0.021)	**0.9921 (±0.019)**
Error	0.0091 (±0.001)	0.0082 (±0.002)	0.0190 (±0.007)	0.0179 (±0.006)	0.0106 (±0.003)	0.0077 (±0.002)
ER						
F-measure	0.8523 (±0.026)	0.8666 (±0.019)	0.5556 (±0.031)	0.6221 (±0.028)	0.8485 (±0.016)	**0.9385 (±0.031)**
Recall	0.8009 (±0.019)	0.8301 (±0.022)	0.4005 (±0.025)	0.4728 (±0.033)	0.8025 (±0.018)	**0.8978 (±0.036)**
Precision	0.9107(±0.012)	0.9124 (±0.021)	0.9020 (±0.029)	0.9092 (±0.031)	0.9001 (±0.015)	**0.9838 (±0.034)**
$\rho_{\mathbf{W}}$	0.8984 (±0.018)	0.9201 (±0.021)	0.9289 (±0.025)	0.9221 (±0.022)	0.9042 (±0.011)	**0.9864 (±0.017)**
Error	0.0064 (±0.002)	0.0059 (±0.004)	0.0145 (±0.003)	0.0185 (±0.008)	**0.0017 (±0.004)**	0.0117 (±0.002)
BA						
F-measure	0.7257 (±0.021)	0.7835 (±0.029)	0.5368 (±0.037)	0.4250 (±0.033)	0.7899 (±0.017)	**0.9021 (±0.021)**
Recall	0.6164 (±0.021)	0.7002 (±0.031)	0.3806 (±0.019)	0.2772 (±0.017)	0.7020 (±0.025)	**0.8465 (±0.033)**
Precision	0.8836 (±0.022)	0.9006 (±0.031)	0.9107 (±0.019)	0.9109 (±0.015)	0.9031 (±0.024)	**0.9655 (±0.019)**
$\rho_{\mathbf{W}}$	0.7198 (±0.022)	0.7374 (±0.012)	0.8076 (±0.019)	0.8074 (±0.018)	0.8162 (±0.027)	**0.9752 (±0.018)**
Error	0.0107 (±0.008)	**0.0097 (±0.007)**	0.0162 (±0.013)	0.0191 (±0.011)	0.0169 (±0.009)	0.0127 (±0.012)

K, reaches its peak at $K = 15$, and then decreases. This is because, with a wider signal bandwidth K, more use knowledge can be obtained from the observations. However, when $K > 15$, the useful information extracted from the observed signals decreases. We see that the correlation coefficient first increases with K and then stabilizes after $K = 15$, when the number of observations is 300. This is because the observed signals are sparse.

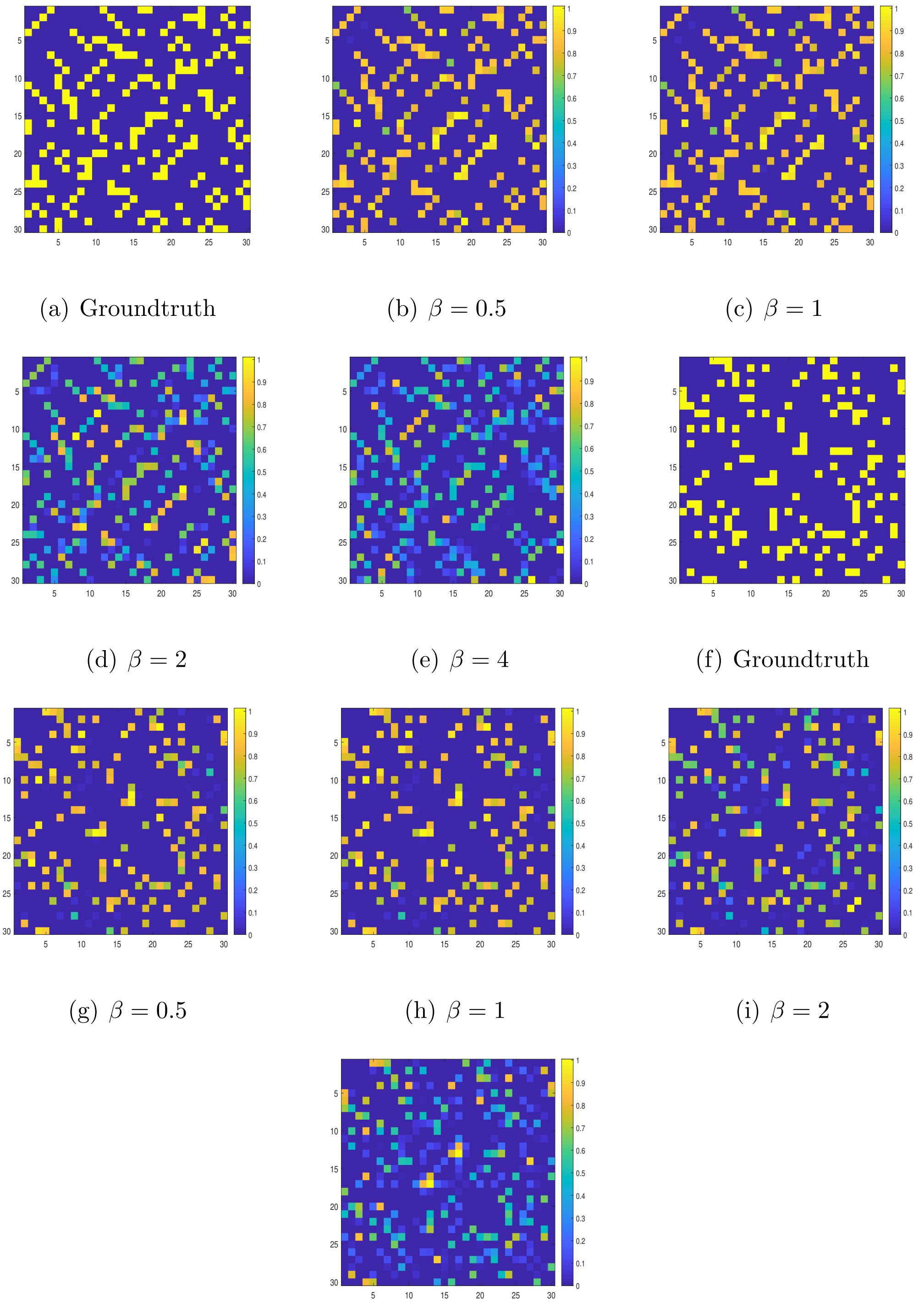
Fig. 3.2 shows the weighted adjacent matrix of the graphs learned by Algorithm 1 under different values of the coefficient β, where the Random Geometric model and Erdős-Rènyi model are considered with $K = 15$, $N = 30$, and $M = 300$. We see that the settings of $\beta = 0.5$ and $\beta = 1$ allow Algorithm 1 to achieve better results than the setting of larger β values. No obvious difference can be observed between the learned graphs under the settings of $\beta = 0.5$ and $\beta = 1$. For this reason, we set $\beta = 0.5$ in the rest of this chapter.

3.4.2 Attention-Deficit Hyperactivity Disorder (ADHD) Data

We apply Algorithm 1 to analyze brain functional connectivity networks and demonstrate the effectiveness of the algorithm to existing techniques with respect to reliability and efficiency. The considered ADHD dataset contains 42 right-handed, boy subjects aged between 11 and 16 years old[1]. To analyze the connectivity of the brain functional networks, we divide a brain into 90 anatomical ROIs using an anatomical automatic labeling template [127], where each node accounts for an ROI with 232 signals from the ADHD dataset. The observed brain signals yield $\mathbf{Y} \in \mathbb{R}^{90\times 232}$.

By running Algorithm 1, we can find that the correlation coefficient $\rho_{\mathbf{W}}$ takes its peak at $K = 62$. For this reason, we take $K = 62$ when executing Algorithm 1 to construct the brain network, i.e., the weighted adjacent coefficients. The brain network is displayed by the BrainNet Viewer toolbox [128]. We consider that the ROIs belong to eight anatomical regions, i.e., frontal lobe (dark blue), orbital surface (blue-green), temporal lobe (green), parietal lobe (blue), occipital lobe (orange),

[1] The dataset is obtained from the ADHD-200 global competition database (https://www.nitrc.org/projects/neurobureau/). The dataset contains 17 ADHD subjects and 25 TD subjects that are analyzed and compared in this section.



(a) Groundtruth

(b) $\beta = 0.5$

(c) $\beta = 1$

(d) $\beta = 2$

(e) $\beta = 4$

(f) Groundtruth

(g) $\beta = 0.5$

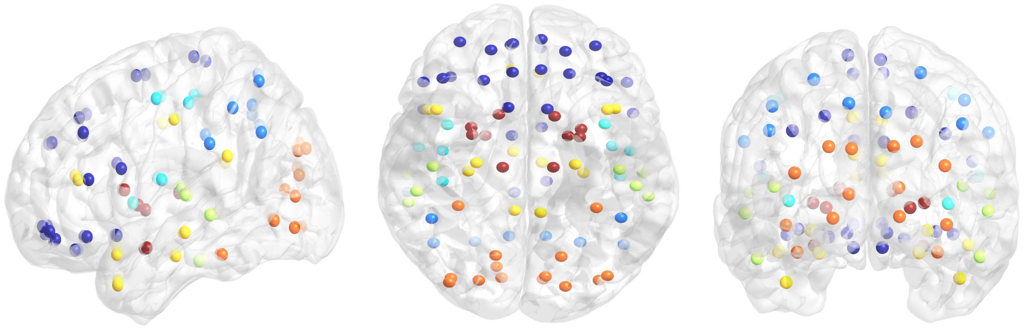
(h) $\beta = 1$

(i) $\beta = 2$

(j) $\beta = 4$

Figure 3.2: The learned graphs with different coefficient β under the Random Geometric model and Erdős-Rènyi model, where $K = 15$, and $N = 30$.

limbic lobe (yellow), insula (light green), and sub cortical gray nuclei (red); see Fig. 3.3.

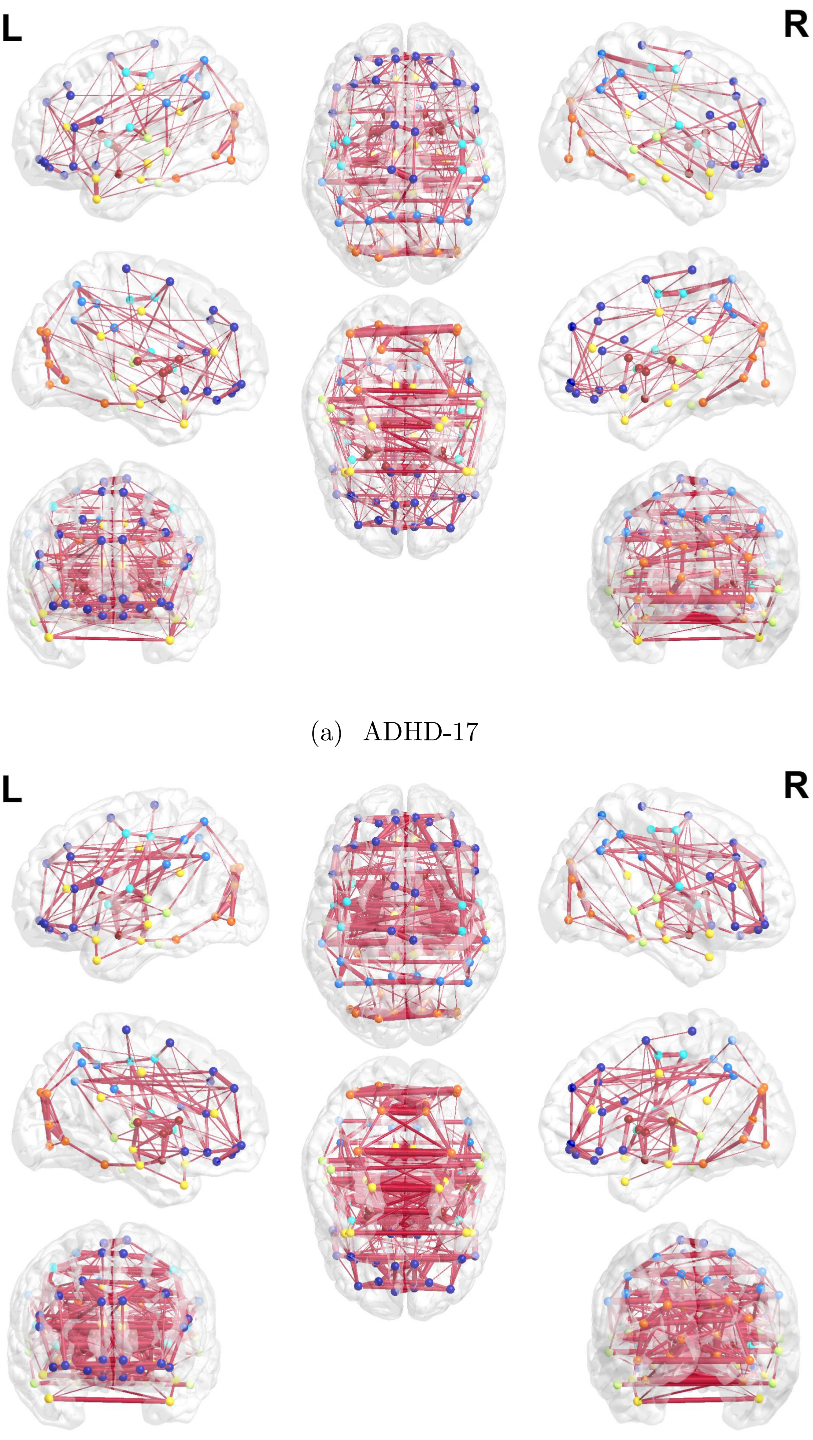


(a) Sagittal View (b) Axial View (c) Coronal View

Figure 3.3: Colored nodes distribution from sagittal, axial, and coronal views. Different colors of nodes represent different blocks of ROIs.

Fig. 3.4 exhibits the differences in the brain functional networks between ADHD subject 17 (ADHD-17) and typical developing (TD) subject 23 (TD-23), which are visible from the sagittal, axial, and coronal views of the brain. We see that both ADHD and TD subjects exhibit an efficient small-world brain network structures and have highly similar hub distributions. Nevertheless, ADHD-17 exhibits considerable asymmetry in the hemispheric brain anatomical network topology, as shown in the sagittal, axial, and coronal views in Figs. 3.4. We also see that the ADHD-17 exhibits decreased structural connectivity and connection strength within the same anatomical regions, especially in the frontal lobe (dark blue nodes), temporal lobe (green nodes), parietal lobe (blue nodes), and occipital lobe (orange nodes), compared to TD-23. Moreover, ADHD-17 has fewer connections between different anatomical regions than TD-23, which is consistent with existing neuroanatomical studies [129].

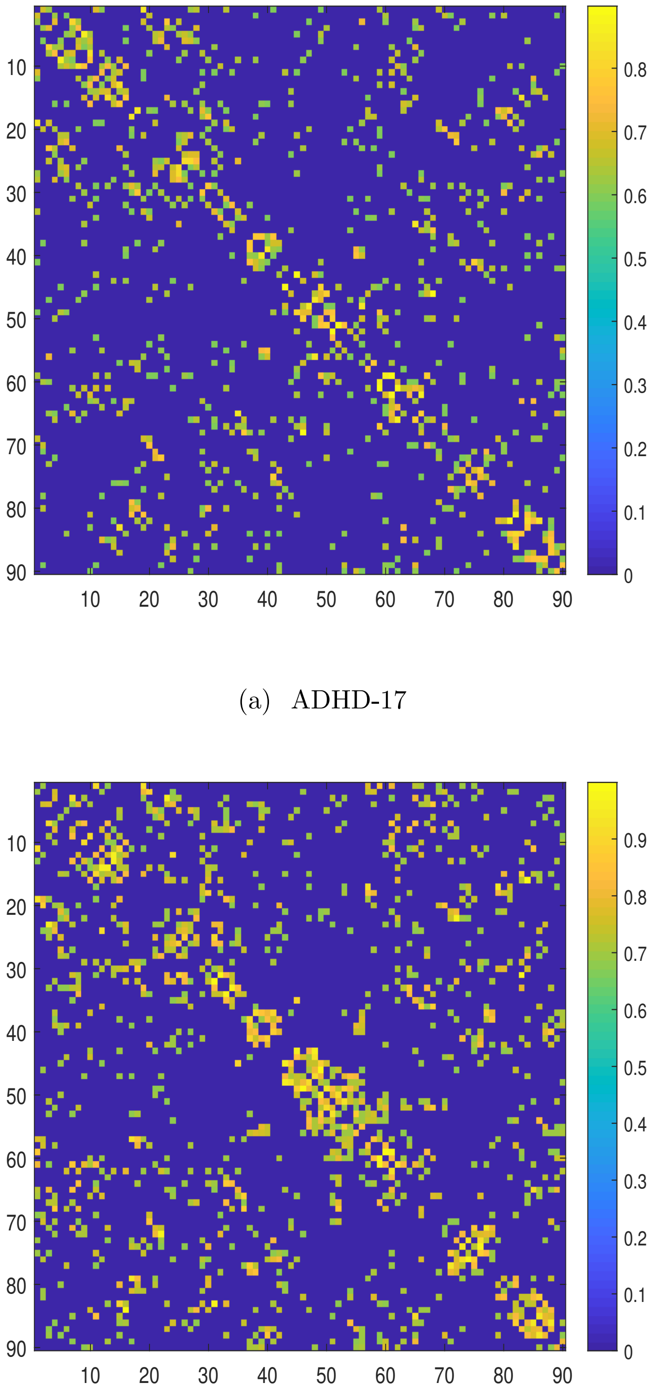
To better display the difference in the connectivity of the brain networks between the ADHD and TD subjects, we transform the brain functional network into the weighted adjacent matrices of ADHD-17 and TD-23 in Fig. 3.5. We find that ADHD-17 exhibits a considerably varying degree of decreased connectivity, compared to TD-23. All changes in ADHD-17 confirm that the brain network topologies of ADHD-17 are disrupted in their distributed neural networks, which is likely related to inattention and hyperactivity. The observed decreased structural connectivity between the local functional networks, and the increased asymmetry in ADHD-17 are consistent with existing neuroanatomical studies, e.g., [129]–[133].



(a) ADHD-17

(b) TD-23

Figure 3.4: Connected brain networks from sagittal, axial, and coronal views depicted by BrainNet Viewer. The colored nodes represent different blocks of ROIs, and the width of the red line is the intensity of the connectivity between two ROIs. (a) The learned graphs are based on ADHD-17 with Algorithm 1. (b) The learned graphs are based on TD-23.



(a) ADHD-17

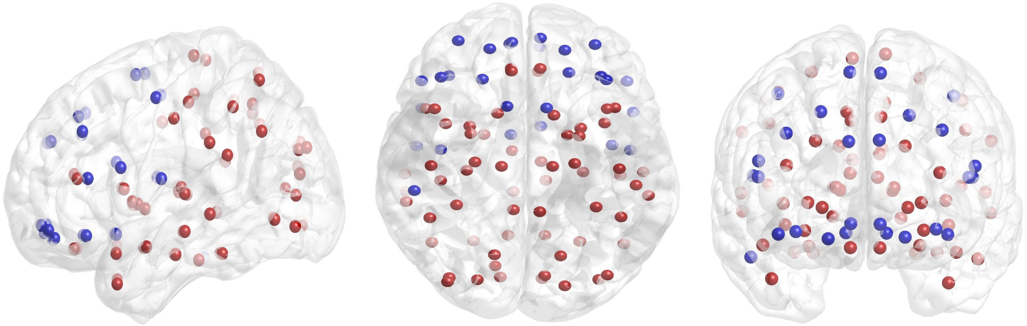
(b) TD-23

Figure 3.5: The weighted adjacent matrix of the learned graph based on the ADHD-17 and TD-23.

To further gauge the accuracy of our algorithm, we reconstruct the complete signals of a brain functional network, i.e., $\mathbf{Y}$, based on only part of the signals in the test dataset and the graph Laplacian inferred from the training dataset. The time-series brain signals are taken from a randomly selected ADHD subject, i.e., ADHD-17, in the dataset. At the training stage, we learn the graph Laplacian of ADHD-17 using Algorithm 1. At the testing stage, the amount of the observable brain signals, e.g., the amount of brain signal waveforms, is set to be the signal bandwidth, i.e., K. We rely on the signals observed at 62 of the total 90 brain regions, and the learned graph Laplacian, to recover the remaining brain signals of 28 ROIs.

As shown in Fig. 3.6, the brain signals at the red nodes are the observed signals, and the signals at the dark blue nodes are unknown and to be reconstructed. We use a greedy algorithm to reconstruct the signals[10]. Figs. 3.7 – 3.9 illustrate the ground-truth brain signals of ADHD-7 at node 9, node 13, and node 23 using the blue line, and the correspondingly reconstructed version using the orange line. Node 9, node 13, and node 23 are not among the 62 observable nodes. We also reconstruct the brain signals at node 9 of ADHD-24 and ADHD-33, as shown in Figs. 3.10 and 3.11, respectively. We see that the reconstructed signals are consistent with the ground-truth signals, indicating the effectiveness of Algorithm 1. For comparison, we also reconstruct the ADHD signals by running current advanced graph learning methods, i.e., Sar-ESA [35], Sar-TV [35], and Humbert [94], under the same experimental settings as in Figs. 3.7(b)–3.7(d). Their reconstructed signals are much less accurate than Algorithm 1.

We adopt the coefficient of determination, denoted by R^2, to quantify the reconstruction efficiency of the considered algorithms, i.e., $R^2 = 1 - \frac{\Sigma_{i=1}^{N}\Sigma_{j=1}^{M}(P_{ij}-Y_{ij})^2}{\Sigma_{i=1}^{N}\Sigma_{j=1}^{M}(P_{ij}-\bar{Y}_i)^2}$, where P_{ij} and Y_{ij} are the reconstructed and ground-truth brain signals, respectively; and $\bar{Y}_i$ is the mean ground-truth brain signals at the i-th node. Fig. 3.12 shows that Algorithm 1 is superior to the state-of-the-art algorithms in the signal reconstruction accuracy, under all diverse values of K. As shown in Fig. 3.12, apart from the scenario with $K = 62$ observable nodes, we also consider the scenarios with fewer numbers of observable nodes, i.e., $K = 22$, 32, 42, and 52.



(a) Sagittal View (b) Axial View (c) Coronal View

Figure 3.6: Nodes distribution from sagittal, axial, and coronal views. The brain signals at the red nodes are the observed signals, and the signals at the dark blue nodes are unknown and to be reconstructed.

3.5 Conclusion

This chapter developed a new AO-based graph learning technique to learn the graph topology of band-limited, fMRI-based brain signals. To do this, we first revealed that the frequency-domain representation of the band-limited signals is a function of the graph Laplacian, thereby transforming the learning problem to only learn the graph Laplacian by estimating its eigenvectors and eigenvalues in an alternating manner. By unveiling that the feasible solution of the eigenvectors is on a Stiefel manifold, we proposed to solve the eigenvectors using Stiefel manifold dual gradient descent and the eigenvalues using the ADMM. Experiments corroborated that our technique can improve substantially the learning accuracy of fMRI-based brain signals than the state of the art, e.g., by over 30%.

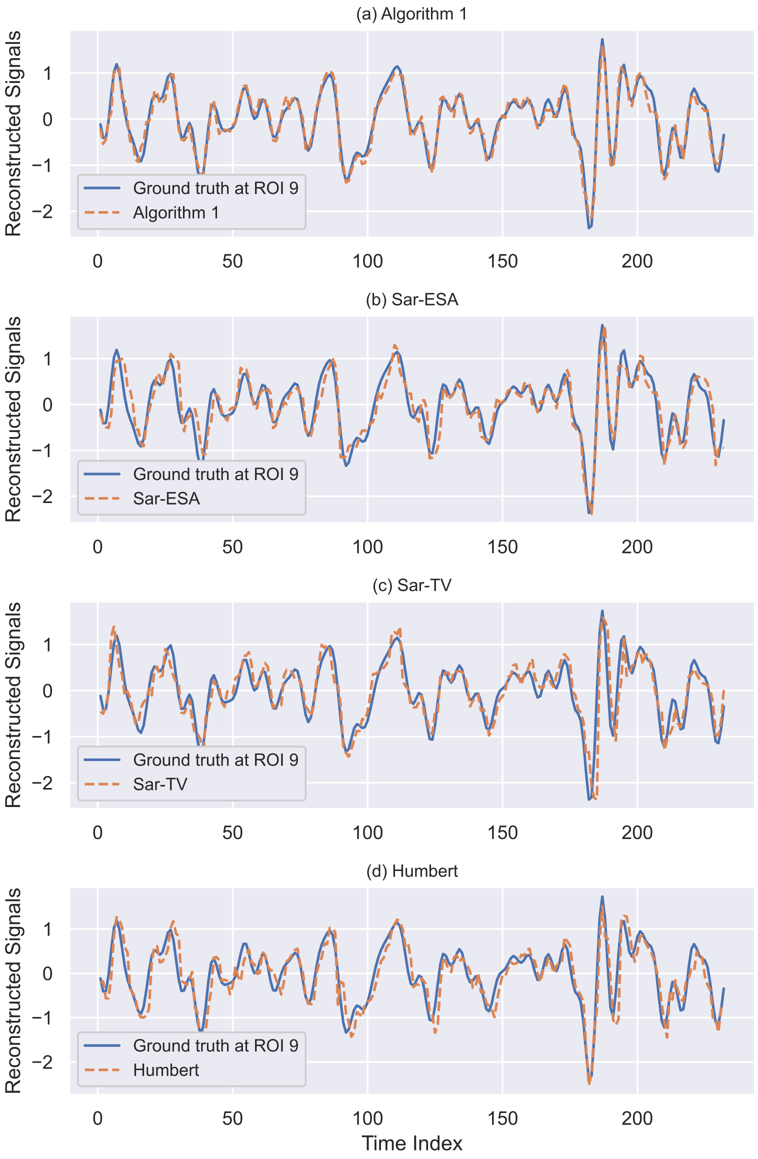


Figure 3.7: The reconstructed signals of the anatomical ROI 9 of ADHD-17 using the different methods.

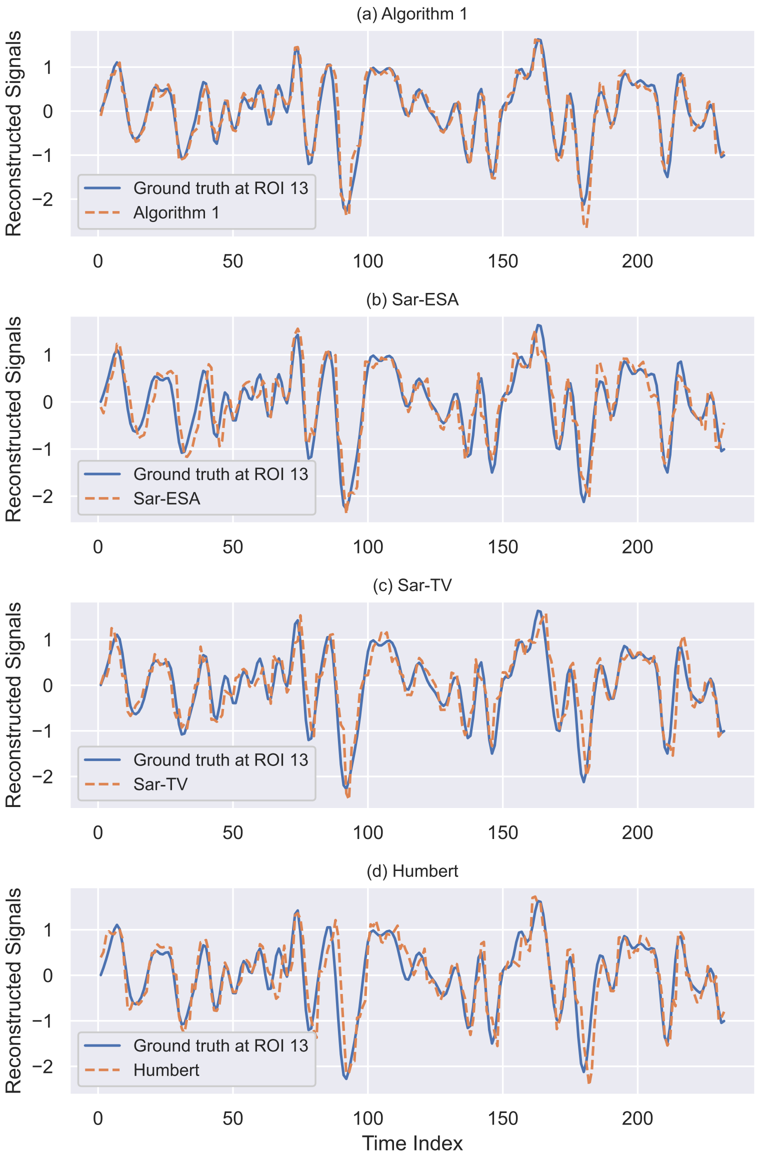


Figure 3.8: The reconstructed signals of the anatomical ROI 13 of ADHD-17 using the different methods.

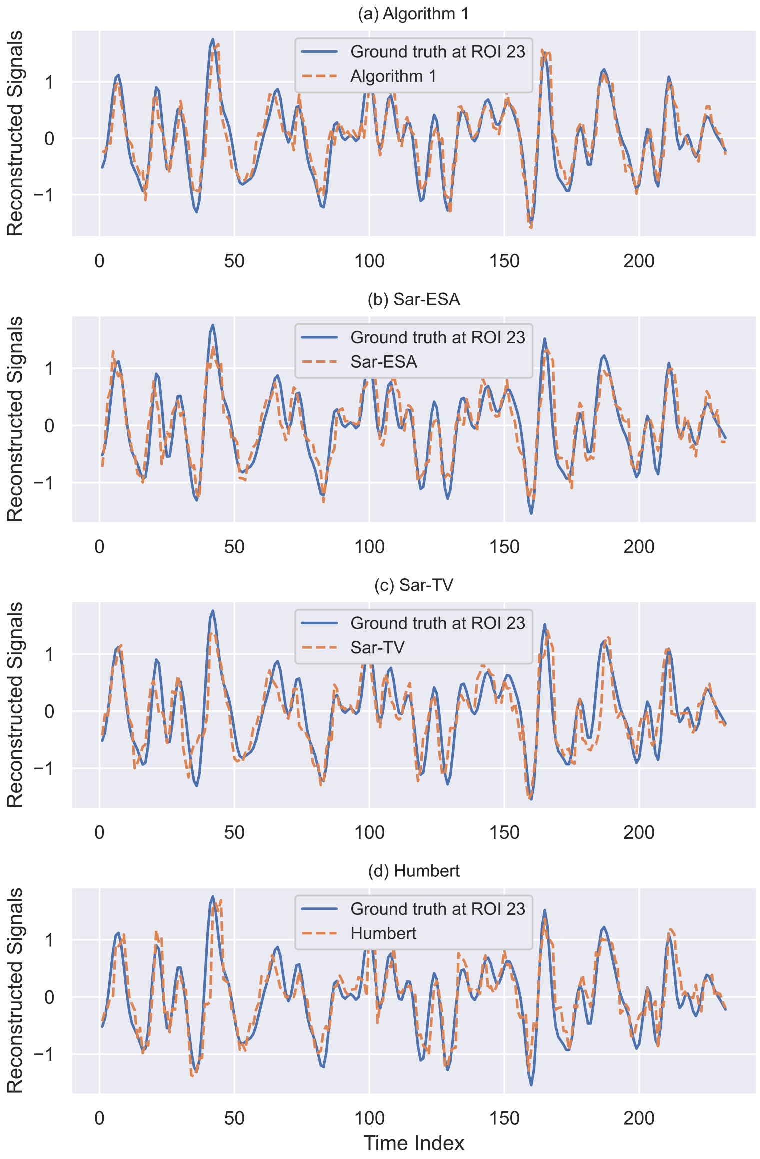


Figure 3.9: The reconstructed signals of the anatomical ROI 23 of ADHD-17 using the different methods.

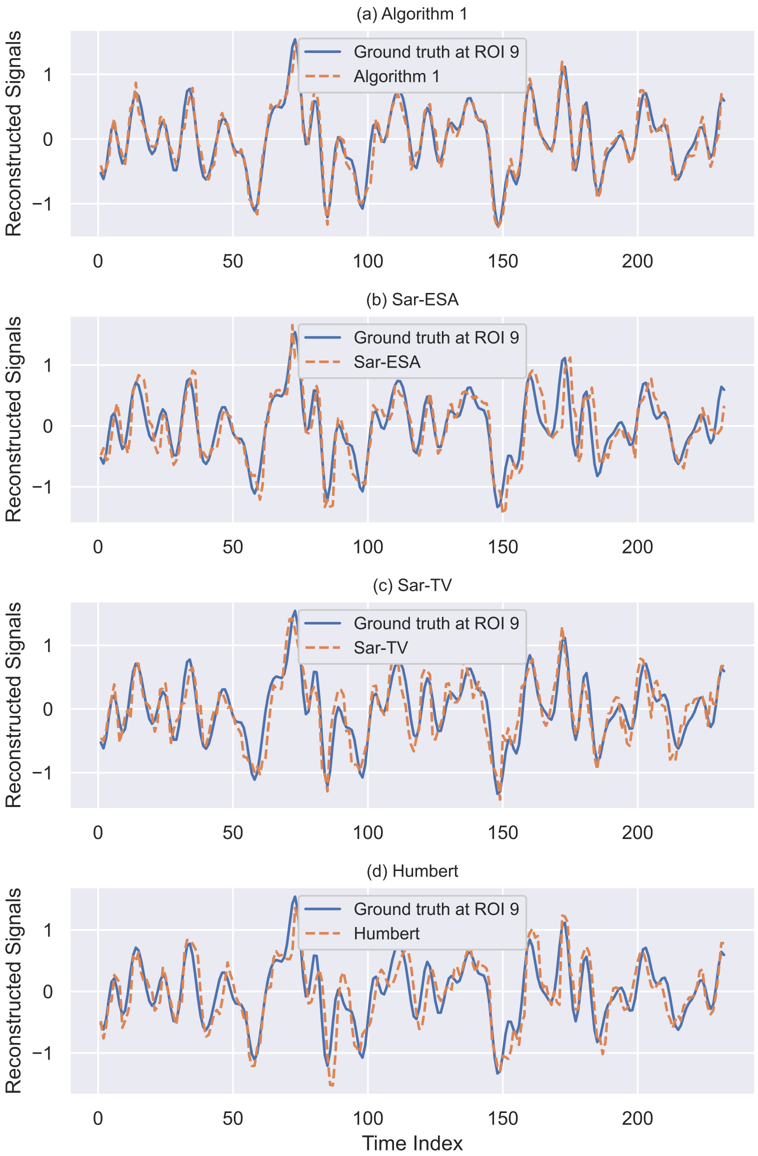


Figure 3.10: The reconstructed signals of the anatomical ROI 9 of ADHD-23 using the different methods.

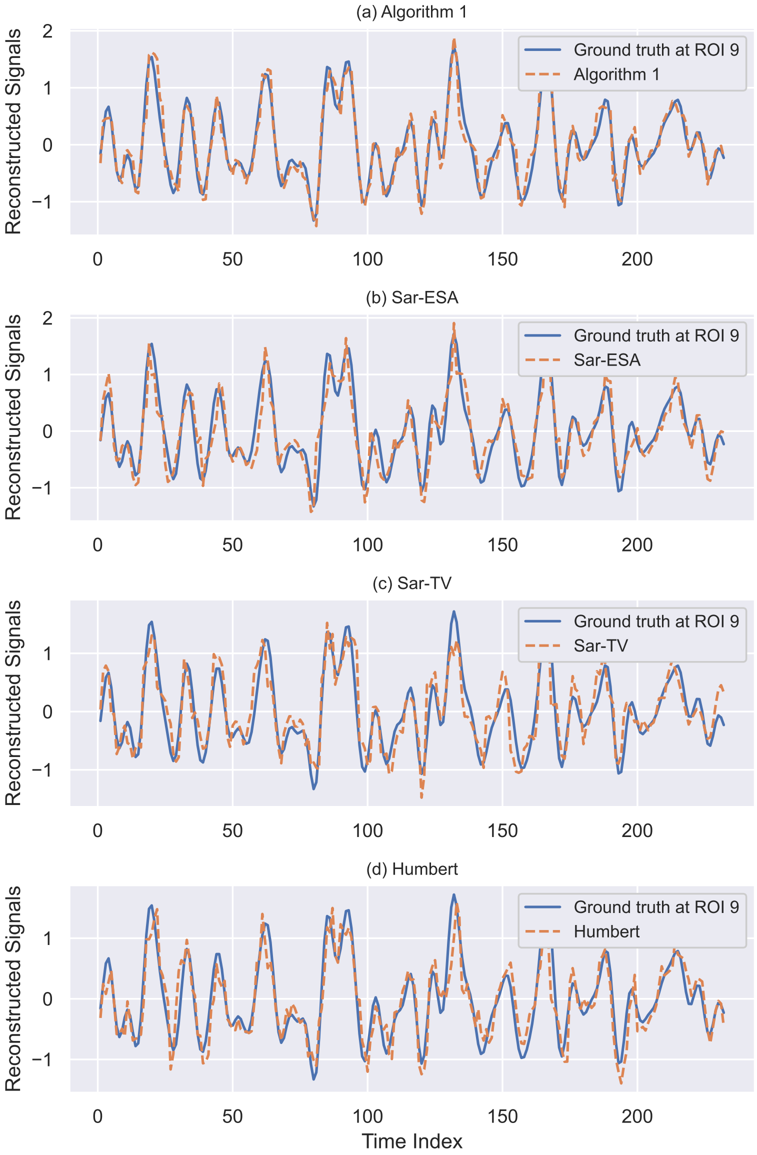


Figure 3.11: The reconstructed signals of the anatomical ROI 9 of ADHD-33 using the different methods.

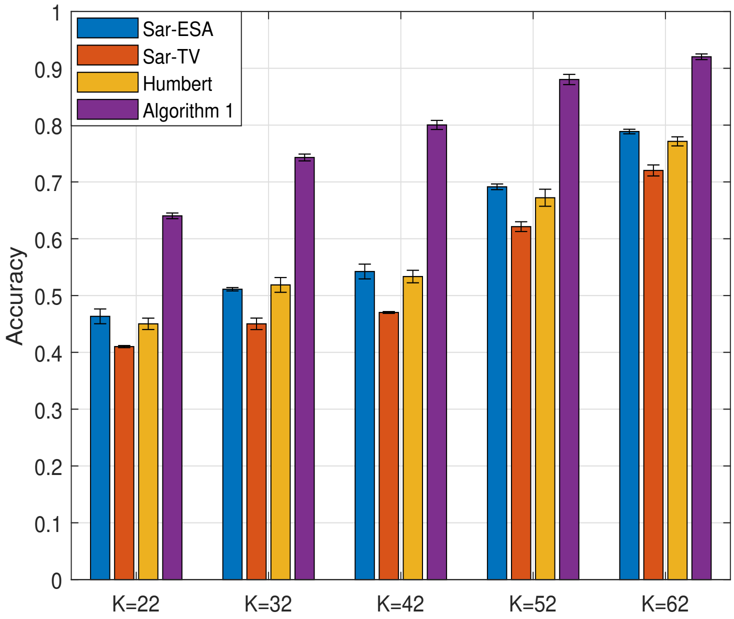


Figure 3.12: Mean efficiency of reconstruction between different methods upon the entire ADHD dataset.

Chapter 4

Graph Learning from Band-Limited Data by Graph Fourier Transform Analysis

4.1 Introduction

Graph learning is the technique used to uncover the latent graph topologies of data [41]. Existing graph learning methods cannot accurately and efficiently infer the graph topology (i.e., graph Laplacian [47]) of band-limited graph signals due to difficulties in joint estimation of both the frequency-domain representation and the GFT basis converting captured data to the frequency domain. In earlier studies, underlying graph topologies were assumed to be known [13], [14]. Graph learning was used to reconstruct the graphs underlying datasets. An overview of graph learning methods from statistical approaches to GSP-based network inference was provided in [39]. Classical graph topology inference methods, such as graph lasso [16] and covariance selection [15], estimated the covariance matrices of graph signals. More recent graph learning techniques enforced smoothness to graph signals prior to topology inference [18]–[22], [42], [43]. In other words, it has been typically assumed that the frequency-domain representations of graph signals have unlimited bandwidths, e.g., for mathematical tractability. The more general band-limited graph signals were studied in [35], where a two-step strategy approximately estimated an orthonormal sparsifying transform with AO and then recovered the graph Laplacian matrix with convex optimization. However, the fidelity of the topologies inferred from the band-limited graph signals was penalized due to the use of the AO-based approximation. In [134], a different problem was considered to learn the subspace clustering of graph signals by learning a similarity matrix, as opposed to the graph Laplacian. The AO was used to estimate the similarity matrix and a cluster/label matrix in an alternating manner.

This chapter presents a new graph learning technique that learns the graph topology of observed band-limited graph signals. The graph Laplacian can be used to analyze network properties or reconstruct missing data resulting from faulty measurements.

- We reveal analytically the intrinsic relation between the unknown frequency-domain representation of general band-limited graph signals and the GFT basis that can transform the signals to the frequency domain;

- A new analytical expression is derived to determine analytically the GFT basis of observed band-limited signals (as opposed to being numerically and approximately solved in [35]);

- Given the GFT basis and the statistical knowledge of the graph Laplacian, the estimation of the eigenvalues of the graph Laplacian is convex and efficiently derived using ADMM.

Numerical experiments show that, under three popular graph models, our new technique can efficiently infer the topology of graph signals and improve the estimation precision. Experiments are executed on an actual temperature dataset and an ADHD dataset. The new technique provides reasonable inference on real-world datasets and superb capability of recovering corrupted data in the datasets in comparison to the current state of the art.

4.2 System Model

The considered network comprises N nodes, at which a vector of signals, denoted by $\mathbf{y}_m \in \mathbb{R}^{N\times 1}$, are observed at the m-th sample $(m = 1, \cdots, M)$. M is the observed sample size. Let $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_M] \in \mathbb{R}^{N\times M}$ collect M observations of the band-limited signals of the network. As considered in [13] and [35], we aim to infer the graph topology from $\mathbf{Y}$. Specifically, we wish to characterize the network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ captured by a weighted and undirected adjacency matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$ with N vertices. $\mathcal{V} = \{1, \cdots, N\}$ is the set of N vertices, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges.The weighted adjacency matrix $\mathbf{W}$ collects all the edges with $W_{ij} = W_{ji} \neq 0 \ \forall (i, j) \in \mathcal{E}$.

Let $\mathbf{D} \triangleq \operatorname{diag}(\mathbf{W1})$ define the degree matrix containing the node degrees at its diagonal. Also, suppose that each node is connected to at least one other node, ensuring no isolated nodes are in the graph. In other words, none of the diagonal elements is zero in $\mathbf{D}$. Then, according to [84], the combinatorial graph Laplacian of $\mathcal{G}$ is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W}. \tag{4.1}$$

The graph Laplacian is a semi-definite matrix with positive elements along its main diagonal and non-positive elements anywhere else [84]. By eigenvalue decomposition, $\mathbf{L}$ is rewritten as:

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \tag{4.2}$$

where $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \cdots, \lambda_N)$ is the diagonal matrix containing the Laplacian eigenvalues. $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_N]$ is an orthonormal matrix collecting all eigenvectors.

To infer the topological knowledge of $\mathcal{G}$, the Laplacian matrix $\mathbf{L}$ needs to be estimated. As done in [13], [35], and [119], we enforce $\mathbf{Y}$ to be band-limited over graph $\mathcal{G}$, e.g., the observation signals are sparse in the canonical domain [13], [35], [119]. GFT [10] has been utilized to decompose $\mathbf{Y}$ into orthonormal components $\mathbf{U}$ in the Laplacian $\mathbf{L}$ [84]. For any $m \in \{1, \cdots, M\}$, the GFT of the observed signal $\mathbf{y}_m$, denoted by $\mathbf{s}_m$, projects $\mathbf{y}_m$ onto the subspace spanned by $\mathbf{U}$, i.e., $\mathbf{s}_m = \mathbf{U}^T\mathbf{y}_m$. With the band-limited property of $\mathbf{y}_m$, $\mathbf{s}_m$ is a sparse vector and captures the key characteristics of $\mathbf{y}_m$ in the frequency domain. The band-limited signal is written as $\mathbf{y}_m = \mathbf{U}\mathbf{s}_m$. Let $\mathbf{S} = [\mathbf{s}_1, \cdots, \mathbf{s}_M] \in \mathbb{R}^{N\times M}$ collect all $\mathbf{s}_m, m = 1, \cdots, M$. We have

$$\mathbf{Y} = \mathbf{U}\mathbf{S}. \tag{4.3}$$

The frequency-domain representation $\mathbf{S}$ depends on both $\mathbf{Y}$ and the graph topology, or more explicitly, the GFT basis, $\mathbf{U}$. It is not straightforward to determine $\mathbf{S}$, given $\mathbf{Y}$. $\mathbf{S}$ and $\mathbf{U}$ need to be jointly estimated, as done in the existing literature [35]. With the sparsity of $\mathbf{s}_m$ $(m \in \{1, \cdots, M\})$, we set $\mathbf{S} \in \mathcal{B}_K$ as a K-block sparse matrix with multiple all-zero row-vectors. K specifies the bandwidth of the frequency-domain representation of the observed band-limited graph signal $\mathbf{Y}$. $\mathcal{B}_K$ collects all K-block sparse matrices [10]:

$$\mathcal{B}_K \triangleq \{\mathbf{S} \in \mathbb{R}^{N\times M}, \mathbf{S}(i,:) = \mathbf{0}, i \notin \mathcal{K} \subseteq \mathcal{V},\ K = |\mathcal{K}|\}, \tag{4.4}$$

where $\mathbf{S}(i,:)$ is the i-th row of $\mathbf{S}$, and $\mathcal{K} \in \mathcal{V}$.

4.3 Problem Statement

We estimate the Laplacian $\mathbf{L}$ and, in turn, the topological knowledge of graph $\mathcal{G}$ substantiating the observation $\mathbf{Y}$. Given the band-limitedness of $\mathbf{Y}$, we cast the problem as

$$\min_{\mathbf{L},\mathbf{U}\in\mathbb{R}^{N\times N},\mathbf{S}\in\mathbb{R}^{N\times M}} \|\mathbf{Y} - \mathbf{U}\mathbf{S}\|_F^2 + f(\mathbf{L}, \mathbf{Y}) \tag{4.5a}$$

$$\text{s.t.} \quad \mathbf{U}\mathbf{U}^T = \mathbf{I}_N, \tag{4.5b}$$

$$\mathbf{S} \in \mathcal{B}_K, \tag{4.5c}$$

$$\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T,\ \mathbf{L} \in \mathbb{L}, \tag{4.5d}$$

$$\mathbf{u}_1 = \frac{1}{\sqrt{N}}\mathbf{1}, \tag{4.5e}$$

where $||\cdot||_F$ stands for the Frobenius norm. The objective (4.5a) consists of two terms. The first term of (4.5a) penalizes any discrepancy between $\mathbf{U}\mathbf{S}$ and $\mathbf{Y}$. The second term characterizes a regularized function [18], [35]. The details of $f(\mathbf{L}, \mathbf{Y})$ are provided in Section 4.5. Constraint (4.5b) ensures $\mathbf{U}$ to be a unitary matrix complying with (4.2). Constraint (4.5c) imposes the K-block sparsity of $\mathbf{S}$ in (4.4). Constraint (4.5d) guarantees that $\mathbf{L}$ refers to a valid Laplacian matrix that satisfies the necessary conditions and properties, and $\mathbb{L}$ collects all valid candidates to $\mathbf{L}$ [84], i.e.,

$$\mathbb{L} = \{\mathbf{L} \succeq \mathbf{0} | \mathbf{L}\mathbf{1} = \mathbf{0}, L_{ij} \leq 0, i \neq j\}, \tag{4.6}$$

where $\mathbf{0}$ stands for an all-zero vector. Since $\mathbf{L1} = \mathbf{0}$ in (4.6), 0 is an eigenvalue of $\mathbf{L}$, leading to $\mathbf{u}_1 = \frac{1}{\sqrt{N}}\mathbf{1}$ in (4.5e). $\mathbf{u}_1$ is the first column of $\mathbf{U}$.

Remark 1 *Problem (4.5) is non-convex due to the orthonormal property in (4.5b) and the sparse property in (4.5c). We decouple* (4.5) *into the two phases. The first phase is that, given the observation* $\mathbf{Y}$*, we estimate* $\mathbf{U}$*, as described in Section 4.4. Based on the estimated* $\mathbf{U}$*, the second phase estimates the eigenvalues of* $\mathbf{L}$*,* $\mathbf{\Lambda}$*, as delineated in Section 4.5.*

4.4 Closed-form Expression for GFT Basis

Given the observed signal $\mathbf{Y}$, we solve problem (4.5) by first estimating the GFT basis $\mathbf{U}$ to minimize $\|\mathbf{Y} - \mathbf{US}\|_F^2$ subject to $\mathbf{UU}^T = \mathbf{I}_N$, $\mathbf{S} \in \mathcal{B}_K$, and $\mathbf{u}_1 = \frac{1}{\sqrt{N}}\mathbf{1}$ (and then estimating the eigenvalues $\mathbf{\Lambda}$ to minimize the regularizer $f(\mathbf{Y}, \mathbf{L})$ given the obtained $\mathbf{U}$, as will be discussed in Section 4.5). The reason for beginning with the GFT basis, $\mathbf{U}$, is the paramount importance of $\mathbf{U}$ in graph learning and of the capability to recover $\mathbf{U}$ from $\mathbf{Y}$ in the absence of the *a-priori* knowledge about the statistical distribution of $\mathbf{L}$.

According to the definition $\mathbf{Y} = \mathbf{U}\,\mathbf{S}$ and the orthonormality of the unitary matrix $\mathbf{U}$ in (4.5b), we first solve [35, eq. (8)]

$$\min_{\mathbf{U}\in\mathbb{R}^{N\times N},\mathbf{S}\in\mathbb{R}^{N\times M}} \|\mathbf{Y} - \mathbf{US}\|_F^2 = \left\|\mathbf{U}^T\mathbf{Y} - \mathbf{S}\right\|_F^2, \text{ s.t. (4.5b), (4.5c), (4.5e).} \tag{4.7}$$

Despite its convex objective, problem (4.7) is non-convex due to the non-convexity of (4.5b) and (4.5c). We reorganize (4.7) as

$$\min_{\mathbf{U}\in\mathbb{R}^{N\times N}} \min_{\mathbf{S}\in\mathcal{B}_K} \sum_{i=1}^{N} \left\|\mathbf{u}_i^T\mathbf{Y} - \mathbf{S}(i,:)\right\|_2^2, \text{ s.t. (4.5b), (4.5e),} \tag{4.8}$$

which can be rewritten as

$$\min_{\mathbf{U}\in\mathbb{R}^{N\times N},\mathcal{K}} \left(\min_{\mathbf{S}\in\mathcal{B}_K} \sum_{i\in\mathcal{K}} \left\|\mathbf{u}_i^T\mathbf{Y} - \mathbf{S}(i,:)\right\|_2^2 + \sum_{i\notin\mathcal{K}} \left\|\mathbf{u}_i^T\mathbf{Y}\right\|_2^2 \right), \text{ s.t. (4.5b), (4.5e).} \tag{4.9}$$

By assessing the objective of (4.9), we find that the optimal index set, $\mathcal{K}$, collects the indices to the K largest entries of $\{\left\|\mathbf{u}_i^T\mathbf{Y}\right\|\}_i^N$, and

$$\mathbf{S}(i,:) = \begin{cases} \mathbf{u}_i^T\mathbf{Y}, & \text{if } i \in \mathcal{K}; \\ \mathbf{0}, & \text{if } i \notin \mathcal{K}. \end{cases} \tag{4.10}$$

Only the $(N-K)$ smallest entries of $\{||\mathbf{u}_i^T\mathbf{Y}||\}_i^N$ remain in the objective of (4.9) after $\mathbf{S}$ is optimized to suppress $\sum_{i\in\mathcal{K}}\left\|\mathbf{u}_i^T\mathbf{Y}-\mathbf{S}(i,:)\right\|_2^2$ in (4.10). The objective is minimized with respect to $\mathbf{S}$. We proceed to find the optimal $\mathbf{U}$, denoted by $\mathbf{U}^*$, to minimize the objective in (4.7).

By plugging (4.10) into the objective of (4.9), (4.7) becomes

$$\mathbf{U}^*=\underset{\mathbf{U},\mathcal{K}}{\arg\min}\sum_{i\notin\mathcal{K}}\left\|\mathbf{u}_i^T\mathbf{Y}\right\|_2^2=\underset{\mathbf{U},\mathcal{K}}{\arg\min}\left\|\mathbf{U}_{\mathcal{K}^{\mathsf{c}}}^T\mathbf{Y}\right\|_F^2=\underset{\mathbf{U},\mathcal{K}}{\arg\max}\left\|\mathbf{U}_{\mathcal{K}}^T\mathbf{Y}\right\|_F^2. \tag{4.11}$$

Here, $\mathcal{K}^{\mathsf{c}}$ is the complementary set of $\mathcal{K}$, i.e., $\mathcal{K}^{\mathsf{c}}=\mathcal{V}\setminus\mathcal{K}$. $\mathbf{U}_{\mathcal{K}}$ and $\mathbf{U}_{\mathcal{K}^{\mathsf{c}}}$ are the matrices collating the column-vectors of $\mathbf{U}$ indexed by $\mathcal{K}$ and $\mathcal{K}^{\mathsf{c}}$, respectively.

Despite the non-convexity of (4.11) in $\mathcal{K}$, we notice that (4.11) is to find the K-dimensional space on which $\mathbf{Y}$ has the largest orthogonal projection:

$$\underset{\mathbf{U},\mathcal{K}}{\arg\max}\left\|\mathbf{U}_{\mathcal{K}}^T\mathbf{Y}\right\|_F^2=\underset{\mathbf{U},\mathcal{K}}{\arg\max}\ \mathrm{tr}\big(\mathbf{Y}^T\mathbf{U}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T\mathbf{Y}\big)=\underset{\mathbf{U},\mathcal{K}}{\arg\max}\ \mathrm{tr}\big(\mathbf{P}_{\mathbf{U}_{\mathcal{K}}}\mathbf{Y}\mathbf{Y}^T\big), \tag{4.12}$$

where $\mathbf{P}_{\mathbf{U}_{\mathcal{K}}}=\mathbf{U}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T$ is the orthogonal projection on the subspace spanned by the column-vectors of $\mathbf{U}_{\mathcal{K}}$.

Lemma 1 *The use of the orthogonal projection,* $\mathbf{P}_{\mathbf{U}_{\mathcal{K}}}=\mathbf{U}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T$*, preserves the orthogonality constraint* (4.5b) *in* (4.12)*.*

Proof 1 *See Appendix B.1.*

By applying (4.12), problem (4.7) can be rewritten as

$$\mathbf{U}^*=\arg\max\nolimits_{\mathbf{U},\mathcal{K}}\ \mathrm{tr}\left(\mathbf{P}_{\mathbf{U}_{\mathcal{K}}}\mathbf{Y}\mathbf{Y}^T\right),\quad \text{s.t.}\quad (4.5\text{e}). \tag{4.13}$$

Theorem 1 *Considering both the cases of* $\mathbf{u}_1\notin\mathbf{U}_{\mathcal{K}}$ *and* $\mathbf{u}_1\in\mathbf{U}_{\mathcal{K}}$*, the closed-form optimal solution* $\mathbf{U}^*=[\mathbf{U}_{\mathcal{K}}^*,\mathbf{U}_{\mathcal{K}^{\mathsf{c}}}^*]$ *to problem (4.7) is given by*

$$\mathbf{U}^*=\mathrm{eigen}\left[\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T\right], \tag{4.14}$$

where eigen$[\mathbf{X}]$ *provides the eigenvectors of* $\mathbf{X}$*.*

Proof 2 *The solution to (4.13) is derived in the two cases:*

1. *In the case of* $\mathbf{u}_1\notin\mathbf{U}_{\mathcal{K}}$*: Let* $\mathbf{P}_{\mathbf{U}_{\mathcal{K}\setminus\{1\}}}$ *represent the orthogonal projection of the subspace of* $\mathbf{U}_{\mathcal{K}}$*, where* $\mathbf{P}_{\mathbf{U}_{\mathcal{K}\setminus\{1\}}}=\mathbf{P}_{\mathbf{U}_{\mathcal{K}}}\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T$*. The objective of* (4.13) *is rewritten as*

$$\max\nolimits_{\mathbf{U},\mathcal{K}}\ \mathrm{tr}(\mathbf{P}_{\mathbf{U}_{\mathcal{K}}}\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T). \tag{4.15}$$

Lemma 2 *If* $\mathbf{u}_1 \notin \mathbf{U}_{\mathcal{K}}$*, the solution* $\mathbf{U}_{\mathcal{K}}^*$ *comprises the eigenvectors corresponding to the* K *largest eigenvalues of* $(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T)\mathbf{Y}\mathbf{Y}^T(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T)^T$.

Proof 3 *See Appendix B.2.*

2. *In the case of* $\mathbf{u}_1 \in \mathbf{U}_{\mathcal{K}}$*: By writing* $\mathbf{U}_{\mathcal{K}} = \left[\mathbf{u}_1, \mathbf{U}_{\mathcal{K}\setminus\{1\}}\right]$*, the objective of (4.13) can be rewritten as*

$$\arg\max_{\mathbf{U},\mathcal{K}} \operatorname{tr}\left(\mathbf{P}_{\mathbf{U}_{\mathcal{K}}}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\right) = \arg\max\nolimits_{\mathbf{U},\mathcal{K}}\operatorname{tr}(\mathbf{P}_{\mathbf{U}_{\mathcal{K}\setminus\{1\}}}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}+\mathbf{u}_1\mathbf{u}_1^{\mathrm{T}}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}) \tag{4.16a}$$

$$= \arg\max\nolimits_{\mathbf{U},\mathcal{K}}\operatorname{tr}(\mathbf{P}_{\mathbf{U}_{\mathcal{K}}}\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T), \tag{4.16b}$$

where $\mathbf{u}_1\mathbf{u}_1^T\mathbf{Y}\mathbf{Y}^T$ *is a constant in* (4.16a) *and suppressed.Apparently,* (4.16b) *is identical to* (4.15)*. The solution* $\mathbf{U}_{\mathcal{K}}^*$ *comprises the eigenvectors corresponding to the* $(K-1)$ *largest eigenvalues of* $\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T$*, and* $\mathbf{u}_1$*, which can be proved in the same way as* ***Lemma 2****.*

Remark 2 (4.14) *is derived analytically by solving exactly the challenging non-convex problem* (4.7)*. The computational cost of* (4.14) *is only* $\mathcal{O}(MN^2)$*. The closed-form solution is achieved by revealing unprecedentedly the relation between the GFT coefficient* $\mathbf{S}$ *and the GFT basis* $\mathbf{U}$*, as done in* (4.10)*; then reformulating losslessly the joint estimation of* $\mathbf{S}$ *and* $\mathbf{U}$ *in* (4.7) *to the estimation of* $\mathbf{U}$ *only in* (4.11)*; and finally solving* $\mathbf{U}^*$ *with non-trivial analysis, as shown in the proof of* ***Theorem 1****. In contrast, the existing techniques, e.g., the one developed in [35], solved problem* (4.7) *numerically and approximately by using the AO. Given the non-convexity of problem* (4.7)*, the AO algorithm in [35] could only solve* $\mathbf{S}$ *and* $\mathbf{U}$ *in an alternating manner. Specifically, at the k-th iteration of the algorithm, given* $\mathbf{U}^{(k-1)}$*,* $\mathbf{S}^{(k)} = \arg\min_{\mathbf{S}\in\mathbb{R}^{N\times M}}\|(\mathbf{U}^{(k-1)})^T\mathbf{Y}-\mathbf{S}\|_F^2$*, s.t.* $\mathbf{S}\in\mathcal{B}_K$*, and then given* $\mathbf{S}^{(k)}$*,* $\mathbf{U}^{(k)}$ *is obtained by solving problem* $\mathbf{U}^{(k)} = \arg\min_{\mathbf{S}\in\mathbb{R}^{N\times M}}\|\mathbf{U}^T\mathbf{Y}-\mathbf{S}^{(k)}\|_F^2$*, s.t.* $\mathbf{U}\mathbf{U}^T = \mathbf{I}_N$*. Not only does this provide a suboptimal solution to* (4.7)*, but incurs a high computational cost of* $\mathcal{O}(MN^2)$ *per iteration.*

Remark 3 *While the analysis of* $\mathbf{U}$ *and* $\mathbf{S}$ *depends on a given* K*, the result of the analysis, i.e.,* $\mathbf{U}^* = \operatorname{eigen}[(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T)\mathbf{Y}\mathbf{Y}^T(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T)^T]$ *in* (4.14)*, turns out to be applicable to any* K*,* $1\leq K\leq N-1$*. To this end, we can set* $K=N-1$ *initially to evaluate all frequency-domain components of the observed signal* $\mathbf{Y}$ *and analytically derive* $\mathbf{U}^*$ *and* $\mathbf{S} = (\mathbf{U}^*)^T\mathbf{Y}$*. Then, we assess the norm of each row of* $\mathbf{S}$*, i.e.,*

$||\mathbf{S}(i,:)||$, *and decide K either to be the number of rows with their norms (or the total of their norms) exceeding a preconfigured threshold, or by sorting the norms in descending order and identifying K with an elbow method.*

4.5 Laplacian Eigenvalues Estimation

With the *a-priori* knowledge of statistics or distribution of $\mathbf{L}$, the regularizer $f(\mathbf{L},\mathbf{Y})$ can capture the desired characteristics of the topology, e.g., the sparsity of $\mathbf{L}$. Given the GFT basis $\mathbf{U}^* \in \mathbb{R}^{N\times N}$ in (4.14) and the definition $\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, to estimate the eigenvalues of $\mathbf{L}$ is in essence to estimate the N eigenvalues, $\boldsymbol{\Lambda}$, as done in this section. Typically, $\mathbf{L}$ follows an exponential distribution [84]. The MAP estimate of $\mathbf{L}$ is written as [84]

$$\min_{\mathbf{L}} \ \mathrm{tr}(\mathbf{L}\mathbf{Y}\mathbf{Y}^T) - M\log|\mathbf{L}| + \alpha\left\|\mathrm{vec}(\mathbf{L})\right\|_1, \ \text{s.t.} \ \mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T, \mathbf{L}\in\mathbb{L}. \tag{4.17}$$

In this sense, $f(\mathbf{L},\mathbf{Y}) = \mathrm{tr}(\mathbf{L}\mathbf{Y}\mathbf{Y}^T) - \log|\mathbf{L}| + \alpha\left\|\mathrm{vec}(\mathbf{L})\right\|_1$ provides a reasonable regularizer for (4.5a). $||\cdot||_1$ stands for ℓ_1-norm, and $|\cdot|$ denotes pseudo-determinant. α is a tunable regularization parameter. With the non-positivity of $\mathbf{L}$ and $\mathbf{L}\mathbf{1} = \mathbf{0}$, $\left\|\mathrm{vec}(\mathbf{L})\right\|_1 = 2\ \mathrm{tr}(\mathbf{L})$. By using the linearity of trace, $f(\mathbf{L},\mathbf{Y})$ is rewritten as

$$f(\mathbf{L},\mathbf{Y}) = \mathrm{tr}\left(\mathbf{L}\mathbf{Y}\mathbf{Y}^T + 2\alpha\mathbf{L}\right) - M\log|\mathbf{L}| = \mathrm{tr}\left(\mathbf{L}\mathbf{T}\right) - M\log|\mathbf{L}|, \tag{4.18}$$

where $\mathbf{T} = \mathbf{Y}\mathbf{Y}^T + 2\alpha\mathbf{I}$. We note that $f(\mathbf{L},\mathbf{Y}) = \mathrm{tr}\left(\mathbf{L}\mathbf{T}\right) - M\log|\mathbf{L}|$ is still intractable due to the singular pseudo-determinant of $|\mathbf{L}|$. To circumvent this impasse, we exploit the equivalence of $\mathrm{tr}\left(\mathbf{L}\mathbf{T}\right) - M\log|\mathbf{L}|$ and $\mathrm{tr}\left(\mathbf{L}\mathbf{T}\right) - M\log\det(\mathbf{L}+\mathbf{J})$ with $\mathbf{J} = \mathbf{u}_1\mathbf{u}_1^T = \frac{1}{N}\mathbf{1}\mathbf{1}^T$ [84, Prop. 1], and rewrite (4.17) as

$$\begin{aligned}&\min_{\mathbf{L}} \mathrm{tr}\left(\mathbf{L}\mathbf{T}\right) - M\log\det(\mathbf{L}+\mathbf{J})\\ &\text{s.t.} \ \mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T, \ \mathbf{L}\succeq\mathbf{0}, \mathbf{L}\mathbf{1} = \mathbf{0}, \ \mathbf{I}\odot\mathbf{L}\geq\mathbf{0}, \ \mathbf{A}\odot\mathbf{L}\leq\mathbf{0},\end{aligned} \tag{4.19}$$

where the constraint is from (4.6) and $\mathbf{A} = \mathbf{1}\mathbf{1}^T - \mathbf{I}$.

With the optimal $\mathbf{U}^*$ obtained in (4.14), we rewrite $\mathbf{L}$ in the following two cases:

1. *In the case of* $\mathbf{u}_1 \notin \mathbf{U}^*_{\mathcal{K}}$, the eigenvectors of $\mathbf{L}$, i.e., $\mathbf{U}^*$, is arranged as $[\mathbf{u}_1, \mathbf{U}^*_{\mathcal{K}}, \mathbf{U}^*_{\mathcal{K}^c\setminus\{1\}}]$. By taking eigenvalue decomposition, $\mathbf{L}$ is written as

$$\mathbf{L} = \left[\mathbf{u}_1, \mathbf{U}^*_{\mathcal{K}}, \mathbf{U}^*_{\mathcal{K}^c\setminus\{1\}}\right]\boldsymbol{\Phi}_1\left[\mathbf{u}_1, \mathbf{U}^*_{\mathcal{K}}, \mathbf{U}^*_{\mathcal{K}^c\setminus\{1\}}\right]^T \tag{4.20a}$$

$$= \left[\mathbf{U}^*_{\mathcal{K}}, \mathbf{U}^*_{\mathcal{K}^c\setminus\{1\}}\right]\boldsymbol{\Phi}_2\left[\mathbf{U}^*_{\mathcal{K}}, \mathbf{U}^*_{\mathcal{K}^c\setminus\{1\}}\right]^T, \tag{4.20b}$$

where $\boldsymbol{\Phi}_1 = \text{blkdiag}\left(0, \boldsymbol{\Lambda}_{\mathcal{K}}, \boldsymbol{\Lambda}_{\mathcal{K}^c\setminus\{1\}}\right)$ is a block-diagonal matrix with $\boldsymbol{\Lambda}_{\mathcal{K}} \succeq \mathbf{0}$ and $\boldsymbol{\Lambda}_{\mathcal{K}} \in \mathbb{R}^{K\times K}$ and collects all the eigenvalues of $\mathbf{L}$ associated with the eigenvectors $[\mathbf{u}_1, \mathbf{U}^*_{\mathcal{K}}, \mathbf{U}^*_{\mathcal{K}^c\setminus\{1\}}]$; and $\boldsymbol{\Phi}_2 = \text{blkdiag}\left(\boldsymbol{\Lambda}_{\mathcal{K}}, \boldsymbol{\Lambda}_{\mathcal{K}^c\setminus\{1\}}\right)$ with $\boldsymbol{\Lambda}_{\mathcal{K}^c\setminus\{1\}} \succeq \mathbf{0}$ and $\boldsymbol{\Lambda}_{\mathcal{K}^c\setminus\{1\}} \in \mathbb{R}^{(N-K-1)\times(N-K-1)}$.

2. *In the case of* $\mathbf{u}_1 \in \mathbf{U}^*_{\mathcal{K}}$, $\mathbf{U}^*$ is arranged as $[\mathbf{u}_1, \mathbf{U}^*_{\mathcal{K}\setminus\{1\}}, \mathbf{U}^*_{\mathcal{K}^c}]$ and $\mathbf{L}$ is

$$\mathbf{L} = \left[\mathbf{u}_1, \mathbf{U}^*_{\mathcal{K}\setminus\{1\}}, \mathbf{U}^*_{\mathcal{K}^c}\right] \boldsymbol{\Phi}_1 \left[\mathbf{u}_1, \mathbf{U}^*_{\mathcal{K}\setminus\{1\}}, \mathbf{U}^*_{\mathcal{K}^c}\right]^T \tag{4.21a}$$

$$= \left[\mathbf{U}^*_{\mathcal{K}\setminus\{1\}}, \mathbf{U}^*_{\mathcal{K}^c}\right] \boldsymbol{\Phi}_2 \left[\mathbf{U}^*_{\mathcal{K}\setminus\{1\}}, \mathbf{U}^*_{\mathcal{K}^c}\right]^T, \tag{4.21b}$$

where $\boldsymbol{\Phi}_1 = \text{blkdiag}\left(0, \boldsymbol{\Lambda}_{\mathcal{K}\setminus\{1\}}, \boldsymbol{\Lambda}_{\mathcal{K}^c}\right)$ with $\boldsymbol{\Lambda}_{\mathcal{K}\setminus\{1\}} \succeq \mathbf{0}$ and $\boldsymbol{\Lambda}_{\mathcal{K}\setminus\{1\}} \in \mathbb{R}^{(K-1)\times(K-1)}$ collects all the eigenvalues of $\mathbf{L}$ associated with the eigenvectors $[\mathbf{u}_1, \mathbf{U}^*_{\mathcal{K}^c\setminus\{1\}}, \mathbf{U}^*_{\mathcal{K}}]$; and $\boldsymbol{\Phi}_2 = \text{blkdiag}\left(\boldsymbol{\Lambda}_{\mathcal{K}\setminus\{1\}}, \boldsymbol{\Lambda}_{\mathcal{K}^c}\right)$ with $\boldsymbol{\Lambda}_{\mathcal{K}^c} \succeq \mathbf{0}$ and $\boldsymbol{\Lambda}_{\mathcal{K}^c} \in \mathbb{R}^{(N-K)\times(N-K)}$.

To unify the presentation of the two cases, we define 1) If $\mathbf{u}_1 \notin \mathbf{U}^*_{\mathcal{K}}$, then $\mathbf{V}_1 = \mathbf{U}^*_{\mathcal{K}}$, $\mathbf{V}_2 = \mathbf{U}^*_{\mathcal{K}^c\setminus\{1\}}$, $\boldsymbol{\Lambda}_1 = \boldsymbol{\Lambda}_{\mathcal{K}}$, and $\boldsymbol{\Lambda}_2 = \mathbf{U}_{\mathcal{K}^c\setminus\{1\}}$; 2) If $\mathbf{u}_1 \in \mathbf{U}^*_{\mathcal{K}}$, then $\mathbf{V}_1 = \mathbf{U}^*_{\mathcal{K}^c\setminus\{1\}}$, $\mathbf{V}_2 = \mathbf{U}^*_{\mathcal{K}}$, $\boldsymbol{\Lambda}_1 = \boldsymbol{\Lambda}_{\mathcal{K}^c\setminus\{1\}}$, and $\boldsymbol{\Lambda}_2 = \boldsymbol{\Lambda}_{\mathcal{K}}$. As a result, the Laplacian matrix $\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ in (4.19) can be written as

$$\mathbf{L} = [\mathbf{V}_1, \mathbf{V}_2]\, \boldsymbol{\Phi}_2\, [\mathbf{V}_1, \mathbf{V}_2]^T, \tag{4.22}$$

where $\boldsymbol{\Phi}_2 = \text{blkdiag}(\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2)$. Since $[\mathbf{V}_1, \mathbf{V}_2]^T\mathbf{u}_1 = \mathbf{0}$, $\mathbf{L1} = \mathbf{0}$ in (4.19) is preserved in (4.22). By plugging (4.22), $\text{tr}(\mathbf{LT})$ can be rewritten as

$$\text{tr}(\mathbf{LT}) = \text{tr}(\mathbf{V}_1\boldsymbol{\Lambda}_1\mathbf{V}_1^T\mathbf{T}) + \text{tr}(\mathbf{V}_2\boldsymbol{\Lambda}_2\mathbf{V}_2^T\mathbf{T}) = \text{tr}(\boldsymbol{\Lambda}_1\widetilde{\mathbf{T}}_1) + \text{tr}(\boldsymbol{\Lambda}_2\widetilde{\mathbf{T}}_2), \tag{4.23}$$

where $\widetilde{\mathbf{T}}_1 = \mathbf{V}_1^T\mathbf{T}\mathbf{V}_1$ and $\widetilde{\mathbf{T}}_2 = \mathbf{V}_2^T\mathbf{T}\mathbf{V}_2$. By plugging (4.22) and $\mathbf{J} = \frac{1}{N}\mathbf{1}\mathbf{1}^T$, $\log\det(\mathbf{L}+\mathbf{J})$ can be rewritten as

$$\begin{aligned}\log\det\left(\mathbf{L}+\mathbf{J}\right) &= \log\det\left(\left[\frac{\mathbf{1}}{\sqrt{N}}, \mathbf{V}_1, \mathbf{V}_2\right]\begin{bmatrix}1 & \\ & \boldsymbol{\Phi}_2\end{bmatrix}\left[\frac{\mathbf{1}}{\sqrt{N}}, \mathbf{V}_1, \mathbf{V}_2\right]^T\right) \\ &= \log\det\left(\text{blkdiag}\left(1, \boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2\right)\right) \\ &= \log\det\left(\boldsymbol{\Lambda}_1\right) + \log\det\left(\boldsymbol{\Lambda}_2\right).\end{aligned} \tag{4.24}$$

Constraint $\mathbf{L1} = \mathbf{0}$ in (4.19) is preserved in (4.22), and so is it in (4.23) and (4.24). The equality constraint $\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ in problem (4.19) is retained throughout the analysis, i.e., in (4.22) and then (4.23) and (4.24). By substituting (4.23) and (4.24), problem (4.19) becomes

$$\min_{\mathbf{\Lambda}_1,\mathbf{\Lambda}_2} \left[\mathrm{tr}(\mathbf{\Lambda}_1\widetilde{\mathbf{T}}_1) + \mathrm{tr}(\mathbf{\Lambda}_2\widetilde{\mathbf{T}}_2) - M\log\det(\mathbf{\Lambda}_1) - M\log\det(\mathbf{\Lambda}_2)\right] \tag{4.25a}$$

$$\text{s.t.} \quad \mathbf{\Lambda}_1 \succeq \mathbf{0}, \mathbf{\Lambda}_2 \succeq \mathbf{0}, \tag{4.25b}$$

$$\mathbf{I} \odot [\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T] \geq \mathbf{0}, \tag{4.25c}$$

$$\mathbf{A} \odot [\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T] \leq \mathbf{0}. \tag{4.25d}$$

Here, $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ may not have full rank (since $\mathbf{U}_{\mathcal{K}}^*$ and $\mathbf{U}_{\mathcal{K}^c}^*$ are organized against $(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^T)\mathbf{Y}\mathbf{Y}^T(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^T)^T$ in (4.14), not against the Laplacian matrix $\mathbf{L}$). $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ can have zero diagonal elements, if the graph $\mathcal{G}$ is not a connected graph. Problem (4.25) is not continually differentiable, and cannot be solved using CVX. By defining $\mathbf{C} = \mathbf{L}$, (4.25) is rewritten as

$$\min_{\mathbf{\Lambda}_1,\mathbf{\Lambda}_2} \left[\mathrm{tr}(\mathbf{\Lambda}_1\widetilde{\mathbf{T}}_1) + \mathrm{tr}(\mathbf{\Lambda}_2\widetilde{\mathbf{T}}_2) - M\log\det(\mathbf{\Lambda}_1) - M\log\det(\mathbf{\Lambda}_2)\right] \tag{4.26a}$$

$$\text{s.t.} \quad \mathbf{\Lambda}_1 \succeq \mathbf{0}, \mathbf{\Lambda}_2 \succeq \mathbf{0}, \tag{4.26b}$$

$$\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T - \mathbf{C} = \mathbf{0}, \tag{4.26c}$$

$$\mathbf{I} \odot \mathbf{C} \geq \mathbf{0}, \ \mathbf{A} \odot \mathbf{C} \leq \mathbf{0}, \tag{4.26d}$$

which can be solved using the ADMM with details provided in Appendix B.3.

Convergence Analysis. The ADMM solves constrained optimization problems [135] with the following structure: $\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x}) + g(\mathbf{y}), \text{ s.t. } \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{y} = \mathbf{E}$, where $\mathbf{x} \in \mathbb{R}^{N\times 1}$ and $\mathbf{y} \in \mathbb{R}^{N\times 1}$ denotes the variables being optimized; $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times N}$ and $\mathbf{E} \in \mathbb{R}^{N\times N}$ are known; $f(\cdot)$ and $g(\cdot)$ represent convex objectives. Therefore, the convergence of the ADMM is guaranteed under the following two conditions: 1) Both $f(\cdot)$ and $g(\cdot)$ are closed, proper and convex sets; hence, the subproblems arising from the updates of $\mathbf{x}$ and $\mathbf{y}$ are solvable; and 2) The Lagrangian has a saddle point. Problem (4.26) satisfies the two conditions. First, $\mathrm{tr}(\mathbf{\Lambda}_1\widetilde{\mathbf{T}}_1) - M\log\det(\mathbf{\Lambda}_1)$ and $\mathrm{tr}(\mathbf{\Lambda}_2\widetilde{\mathbf{T}}_2) - M\log\det(\mathbf{\Lambda}_2)$ in (4.26) correspond to $f(\cdot)$ and $g(\cdot)$, respectively. $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ are strictly convex with proper, closed, and convex sets. The first condition is satisfied. The primary variables $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ are solvable. Second, the Lagrangian of problem (4.25), i.e., $\mathcal{L}$ in (B.5), has a saddle point, since the Hessian of the Lagrangian is indefinite at the stationary point. The second condition is satisfied. As a result, the use of ADMM to solve problem (4.26) is convergent.

Complexity Analysis. For every iteration of ADMM, the complexity of evaluating the primal variable $\mathbf{\Lambda}_1$ is dominated by matrix multiplications and eigenvalue decomposition with a cost of $\mathcal{O}(K^3)$. Likewise, the cost of evaluating the primal variable $\mathbf{\Lambda}_2$ is $\mathcal{O}((N-K-1)^3)$. The complexity of evaluating the primal variable $\mathbf{C}$ is dominated by matrix multiplications with a cost of $\mathcal{O}(N^2K+N^2(N-K-1))$. The update of the dual variable $\mathbf{Z}$ incurs the complexity of $\mathcal{O}(N^2K + N^2(N-K-1))$. Due to $K < N$, the overall cost is $\mathcal{O}(N^3)$ per iteration.

Remark 4 *The proposed approach can be extended to the case with observation noises. In this case, the observation signal, denoted by $\hat{\mathbf{Y}}$, is $\hat{\mathbf{Y}} = \mathbf{Y} + \mathbf{n}$, where $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ is the additive white Gaussian noise, and σ^2 is the variance of the additive white Gaussian noise. When the underlying signals $\mathbf{Y}$ are ergodic and M is sufficiently large, the expectation of $\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T$ over the additive Gaussian observation noises is given by*

$$\mathbb{E}(\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T) = \mathbf{Y}\mathbf{Y}^T + M\sigma^2\mathbf{I}, \tag{4.27}$$

which indicates that $\mathbb{E}(\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T)$ and $\mathbf{Y}\mathbf{Y}^T$ have the same eigenvectors.

Without observation noises, the GFT basis $\mathbf{U}^*$ in (4.14) only depends on the eigenvectors of $\mathbf{Y}\mathbf{Y}^T$. With the observation noises, the GFT basis is

$$\mathbf{U}^* = \text{eigen}\left[\left(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^T\right)\mathbb{E}(\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T)\left(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^T\right)^T\right]. \tag{4.28}$$

Since $\mathbb{E}(\frac{1}{M}\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T - \sigma^2\mathbf{I}) = \frac{1}{M}\mathbf{Y}\mathbf{Y}^T$ (i.e., $\frac{1}{M}\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T - \sigma^2\mathbf{I}$ is an asymptotic unbiased estimate of $\frac{1}{M}\mathbf{Y}\mathbf{Y}^T$ as $M \to \infty$) based on (4.27) and $\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T - M\sigma^2\mathbf{I}$ and $\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T$ have the same eigenvectors, $\mathbf{U}^*$ can be approximated by

$$\mathbf{U}^* \approx \text{eigen}\left[\left(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^T\right)\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T\left(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^T\right)^T\right]. \tag{4.29}$$

Regarding the eigenvalues of $\mathbf{L}$, i.e., $\mathbf{\Lambda}$, the estimation with noise-free observations is formulated in (4.17). With noisy observation $\hat{\mathbf{Y}}$, an asymptotic unbiased estimate of the covariance matrix of $\mathbf{Y}$, i.e., $\lim\limits_{M\to\infty}\frac{1}{M}\mathbf{Y}\mathbf{Y}^T$, is $\frac{1}{M}\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T - \sigma^2\mathbf{I}$, since $\lim\limits_{M\to\infty}\mathbb{E}\left(\frac{1}{M}\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T - \sigma^2\mathbf{I}\right) = \lim\limits_{M\to\infty}\mathbb{E}\left(\frac{1}{M}\mathbf{Y}\mathbf{Y}^T\right)$ [136]. We can replace $\mathbf{Y}\mathbf{Y}^T$ in (17) with $\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T - M\sigma^2\mathbf{I}$. In turn, $\mathbf{T}$ in (4.19) is updated to $\hat{\mathbf{T}} = \hat{\mathbf{Y}}\hat{\mathbf{Y}}^T + (2\alpha - M\sigma^2)\mathbf{I}$ under noisy observations and the rest of the estimation steps developed in this section apply.

It is worth mentioning that **Remark 4** is based on the assumption that the observation noise follows a white Gaussian distribution. If the noise is colored with an unknown and non-diagonal covariance matrix, according to matrix perturbation theory [137], small perturbations on (the elements of) a matrix may result in considerable changes in its eigenvalues and eigenvectors. In this sense, the proposed algorithm could be sensitive to non-Gaussian observation noises.

4.6 Simulations and Experimental Results

Extensive experiments are performed to evaluate the new graph learning technique. The ADMM algorithm is initialized by setting $\mathbf{C}^{(0)}$ and $\mathbf{Z}^{(0)}$ to two symmetric unit matrices, and the step size ρ to 1. We stop the algorithm upon reaching a predefined maximum number of iterations, i.e., 10^4, or the difference of the objective (4.26) is smaller than $\epsilon = 10^{-5}$ between two consecutive iterations. For comparison purposes, we consider the state of the art: Dong's algorithm [18], Saboksayr's algorithm [43], Sardellitti's Total Variation (TV) and Estimated-Signal-Aid (ESA) algorithms [35], and Egilmez's algorithm [84], [85].

4.6.1 Results on Synthetic Data

We consider three different random graphs generated separately from Random Geometric model [124], Erdős-Rènyi model [125], and Barábasi-Alber model [126]. Regarding the Random Geometric model, we generate the graph with six connections per node. For the Erdős-Rènyi model, we generate a graph by randomly connecting labeled nodes. Each edge is included in the graph independently with a probability of 0.3, regardless of any other edge. For the Barábasi-Albert model, we generate the graph with two initial nodes and then add more nodes with two edges for each node addition. For each of the three graph models, we generate over 100 independent random graphs. We then compute the average results of these graphs and plot them in this section.

After generating a graph from the graph models, we can obtain the ground-truth Laplacian of the graph, denoted by $\mathbf{L}_0$. We take the singular value decomposition of $\mathbf{L}_0$ and obtain the ground-truth GFT, denoted by $\mathbf{U}_0$. Then, we generate the observed band-limited graph signal $\mathbf{Y} = \mathbf{U}_0\mathbf{S}_0$ with $\mathbf{S}_0 = [\mathbf{s}_{0,1}, \cdots, \mathbf{s}_{0,M}] \in \mathbb{R}^{N\times M}$. Specifically, $\mathbf{s}_{0,m} \sim \mathcal{N}(0, \mathbf{\Lambda}^{\dagger})$, where $\mathrm{diag}(\mathbf{\Lambda}) = (\lambda_1, \cdots, \lambda_K, 0, \cdots, 0)$. The precision matrix of $\mathbf{s}_{0,m}$ is defined to be the eigenvalue matrix of $\mathbf{L}$ with the largest $(N - K)$ values set to 0, as considered in [119]. The key difference between the band-limited signals and smooth signals is the degenerate values in some dimensions of the multivariate Gaussian signal $\mathbf{S}$. The smooth signals can be considered as a specific scenario of the band-limited signals with $K = N - 1$. Our algorithm designed for general band-limited signals can be applied to smooth signals.

With reference to [35], the performance metrics are F-measure, Recall, Precision score, and the percentage of recovery errors. Let $\mathcal{E}_g$ and $\mathcal{E}_r$ denote the sets of ground-truth and recovered graphs, respectively. Precision measures the proportion of the identified edges in the recovered graphs among the ground-truth graphs, i.e., Precision $= \mathcal{E}_g \cap \mathcal{E}_r/\mathcal{E}_r$. Recall evaluates the proportion of edges from the ground-truth graphs that are correctly identified among the recovered graphs, i.e., Recall $= \mathcal{E}_g \cap \mathcal{E}_r/\mathcal{E}_g$. F-measure combines both Precision and Recall into a single metric to evaluate the overall accuracy of the recovered edges. Specifically, F-measure $=$ $2\cdot$Precision$\cdot$Recall/(Precision + Recall). The correlation coefficient $\rho_{\mathbf{W}}(\mathbf{W}_0, \mathbf{W})$ (or $\rho_{\mathbf{W}}$) between a recovered graph and its ground-truth is defined as $\rho_{\mathbf{W}}(\mathbf{W}_0, \mathbf{W}) = \frac{\sum_{ij} W_{0ij} W_{ij}}{\sqrt{\sum_{ij} W_{0ij}^2}\sqrt{\sum_{ij} W_{ij}^2}}$ [35], where $\mathbf{W}_0$ and $\mathbf{W}$ are the weighted adjacency matrices of the ground-truth and recovered graphs, respectively; and W_{0ij} and W_{ij} are the (i, j)-th elements of $\mathbf{W}_0$ and $\mathbf{W}$, respectively. The estimation error, or "Error", is defined as Error $= \|\mathbf{A} - \mathbf{A}_0\|_F / (N(N-1))$, where $\mathbf{A}$ and $\mathbf{A}_0$ are the binary adjacency matrices of the recovered and ground-truth graphs, respectively.

Fig. 4.1(a) illustrates the convergence behaviors of the proposed algorithm under three different models. Fig. 4.1(b) plots the correlation coefficient of the algorithm, i.e., $\rho_{\mathbf{W}}$, with different values of the regularization parameter α and signal bandwidth K under the Random Geometric model. We see that $\rho_{\mathbf{W}}$ reaches its peak at $\alpha = 0.9$ and $K = 15$; i.e., the optimal regularization parameter is $\alpha = 0.9$ when the signal bandwidth is $K = 15$. Likewise, we can obtain the optimal α under the Erdős-Rènyi and Barábasi-Albert models.

Tables 4.1 and 4.2 compare the proposed algorithm and the benchmarks under the

Table 4.1: Comparison of the considered algorithms, including the new algorithm, Dong's [18], Saboksayr's (Sab.) [43], Sar-TV [35], Sar-ESA [35], and Egilmez's [84], [85], where $K = 3$, $N = 30$, and $M = 300$.

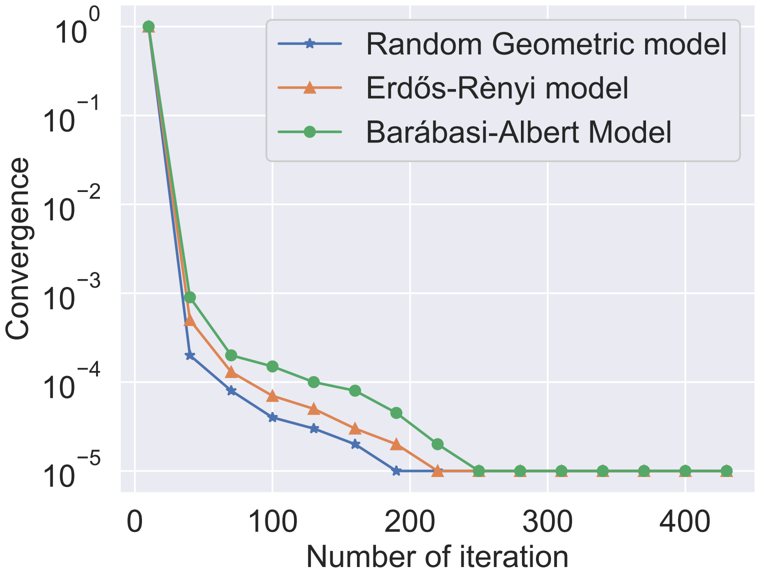
	Dong	**Sab.**	**TV**	**ESA**	**Egilmez**	**Ours**
RG						
Fmeasure	0.5478	0.5499	0.5536	0.5255	0.4770	**0.5597**
Recall	0.4500	0.4476	**0.4627**	0.3913	0.3217	0.4213
Precision	0.7000	0.7129	0.6889	0.8000	**0.9222**	0.8333
$\rho_{\mathbf{W}}$	0.6324	0.5357	0.5996	0.5600	0.6115	**0.6471**
Error	0.0189	0.0185	0.0220	0.0219	0.0261	**0.0177**
ER						
Fmeasure	0.4632	0.4971	0.3784	0.3033	0.2876	**0.5283**
Recall	0.5000	**0.5096**	0.2612	0.2200	0.1725	0.3889
Precision	0.4314	0.4852	0.6863	0.6275	0.8627	**0.8235**
$\rho_{\mathbf{W}}$	0.5611	0.5761	0.4623	0.4496	0.5670	**0.6215**
Error	**0.0170**	0.0199	0.0174	0.0197	0.0200	0.0189
BA						
Fmeasure	**0.5056**	0.5014	0.3784	0.3474	0.3803	0.4808
Recall	**0.3980**	0.3999	0.2612	0.2357	0.2368	0.3289
Precision	0.6946	0.6717	0.6863	0.6607	0.9643	**0.8929**
$\rho_{\mathbf{W}}$	0.4495	0.4486	0.4221	0.3054	0.4151	**0.5135**
Error	**0.0142**	0.0173	0.0177	0.0192	0.0216	0.0169

considered graph models, where each random graph consists of $N = 30$ vertices. $M = 300$. We take $K = 3$ in Table 4.1 and $K = 15$ in Table 4.2. For the proposed algorithm, different values of the regularized parameter α are tried and tested to obtain the best one, as done in Fig. 4.1(b). For fair comparisons, α is also individually tested and optimized for each benchmark. The tables show that the proposed algorithm consistently outperforms all the other techniques considered, indicating its superiority in terms of performance. It achieves high scores in both Fmeasure and correlation coefficient $\rho_{\mathbf{W}}$ under all random graph models. In particular, Table 4.1 shows that the proposed algorithm yields at least 20% improvement in terms of Fmeasure under the Erdős-Rènyi model, compared to Sardellitti's TV and ESA algorithms. By comparing Tables 4.1 and 4.2, we see that the proposed algorithm increasingly outperforms the benchmarks, as K grows.

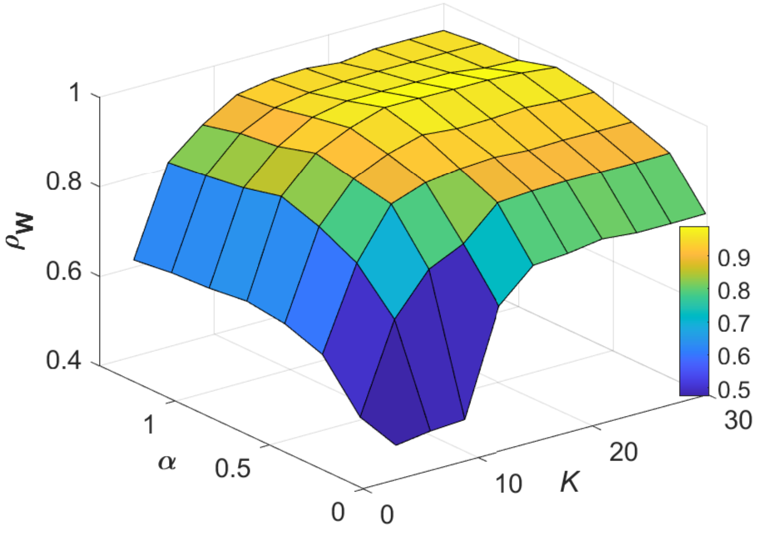
Table 4.2: Comparison of the considered algorithms, where $K = 15$, $N = 30$, and $M = 300$.

	Dong	**Sab.**	**TV**	**ESA**	**Egilmez**	**Ours**
RG						
Fmeasure	0.8517	0.9111	0.7946	0.6642	0.9301	**0.9401**
Recall	0.7479	0.8427	0.6666	0.4972	0.8693	**0.8912**
Precision	0.9889	0.9914	0.9889	0.9999	1.0000	**1.0000**
$\rho_{\mathbf{W}}$	0.9156	0.9342	0.9299	0.9442	0.9641	**0.9895**
Error	0.0091	0.0102	0.0190	0.0179	0.0136	**0.0075**
ER						
Fmeasure	0.8523	0.8671	0.5556	0.6221	0.8963	**0.9254**
Recall	0.8009	0.8183	0.4005	0.4728	0.8425	**0.8907**
Precision	0.9107	0.9221	0.9020	0.9092	0.9575	**0.9610**
$\rho_{\mathbf{W}}$	0.8984	0.9207	0.9289	0.9221	0.9244	**0.9840**
Error	0.0064	0.0059	0.0145	0.0185	0.0115	**0.0083**
BA						
Fmeasure	0.7257	0.7857	0.5368	0.4250	0.4576	**0.8728**
Recall	0.6164	0.6986	0.3806	0.2772	0.3000	**0.8359**
Precision	0.8836	0.8976	0.9107	0.9109	0.9643	**0.9131**
$\rho_{\mathbf{W}}$	0.7198	0.7371	0.8076	0.8074	0.7962	**0.9641**
Error	0.0117	0.0153	0.0162	0.0191	0.0184	**0.0105**

Fig. 4.2(a) evaluates the average correlation coefficients of the considered algorithms against K under the Random Geometric model, where $N = 30$ and $M = 300$. Our algorithm achieves the largest correlation coefficient over the wide spectrum of K. We see that the correlation coefficient first increases with K, reaches its peak at $K = 15$, and then stabilizes. This is because a wider signal bandwidth K allows for a broader range of frequencies to be captured in the observed signals. As a result, more detailed and diverse information can be extracted. When K exceeds 15, the information extracted from the observed signals no longer increases because the observed signals are sparse and the setting of $K \geq 15$ overkills. Fig. 4.2(b) compares the correlation coefficients between different methods under the Erdős-Rènyi model, as M increases. $K = 15$. We see that the correlation coefficients stabilize when $M > 250$. Fig. 4.2(c) compares the computational complexities of the methods with the increase of M, where the Random Geometric model is considered, $K = 15$, and the average of 1,000 independent tests is plotted. We see that Saboksayr's method



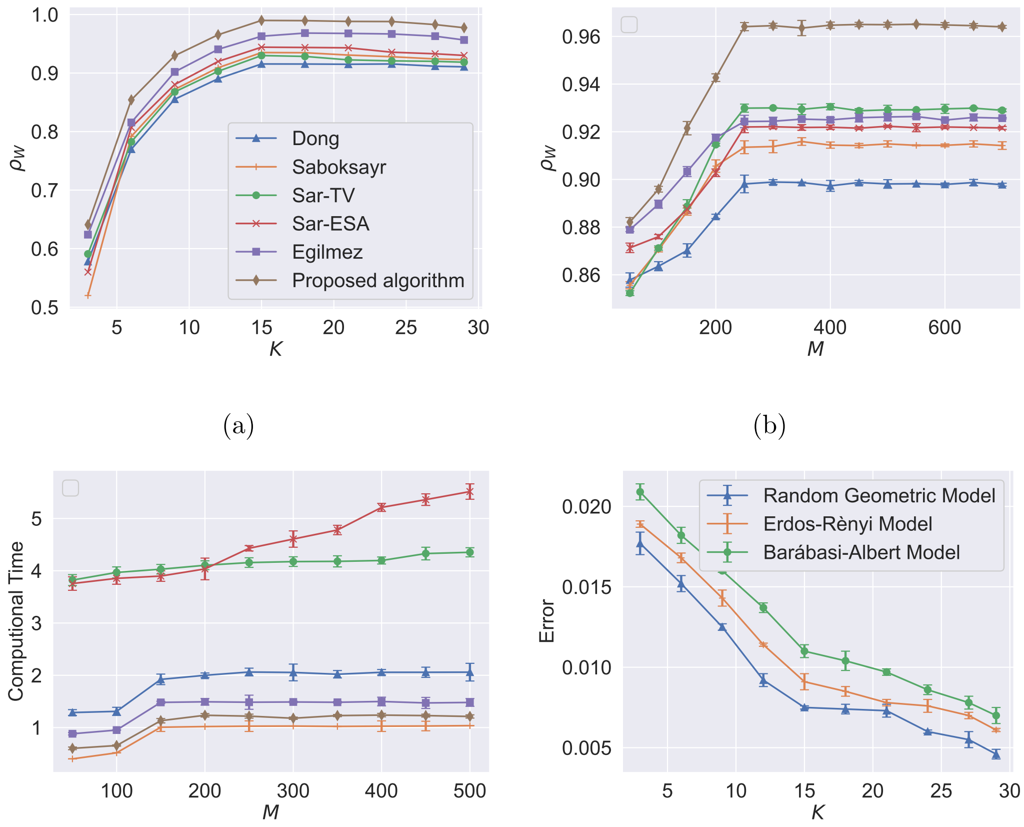
(a)



(b)

Figure 4.1: (a) The convergence of the proposed algorithm. (b) The correlation coefficient vs K and α under the Random Geometric model, where $N = 30$ and $M = 300$.

is the fastest, followed by our algorithm. This indicates that our algorithm provides superb detection accuracy with considerably low complexity. Fig. 4.2(d) illustrates that the Error score consistently decreases as the value of K increases across all three models.



(a)

(b)

(c)

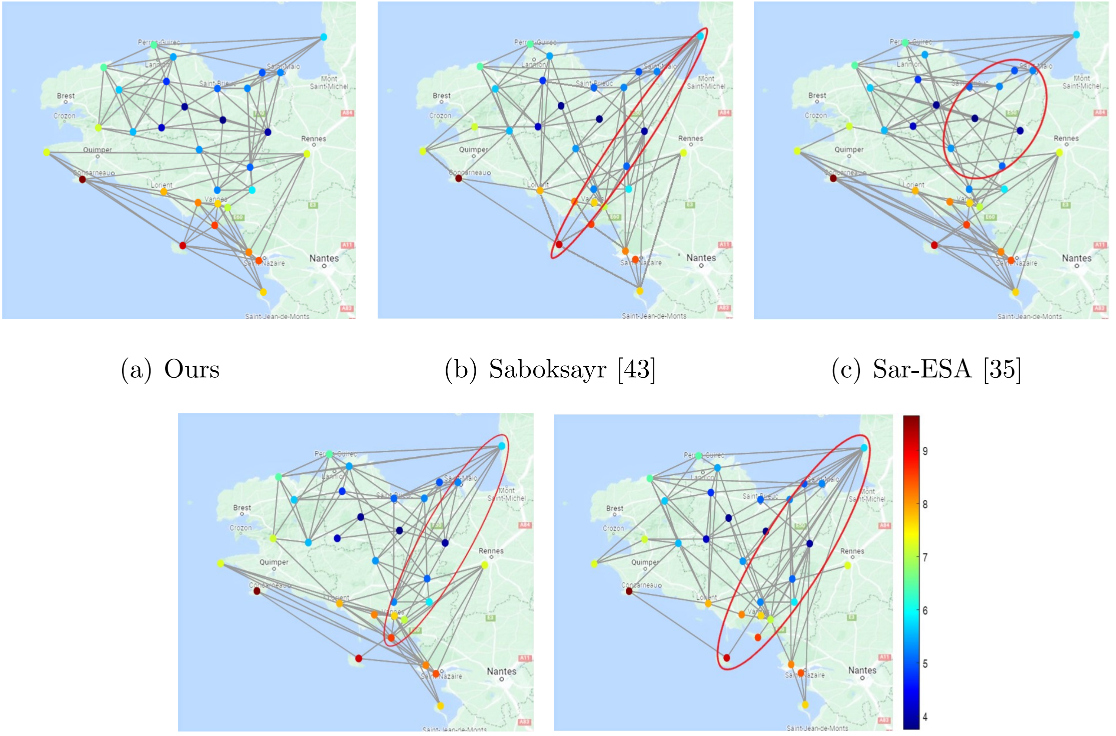
(d)

Figure 4.2: (a) The correlation coefficient with the increased signal bandwidth K under the Random Geometric model, where $N = 30$ and $M = 300$. (b) The correlation coefficient with the increase of M under the Erdős-Rènyi model, where $K = 15$ and $N = 30$. (c) Runtime (in seconds) with the increase of M under the Random Geometric model, where $K = 15$ and $N = 30$. (d) The estimation error of the proposed algorithm with the increased signal bandwidth K, where $N = 30$ and $M = 300$.

4.6.2 Experiment on Temperature Measurement Data

We assess the performance of the proposed graph learning algorithm using a temperature dataset collected from 32 weather stations in Brittany, France[1]. The dataset collects hourly temperatures during a period of 31 days, resulting in a total of $24 \times 31 = 744$ observations. We take the 32 weather stations as the vertices of a graph and the temperatures measured at the stations as the observation $\mathbf{Y} \in \mathbb{R}^{32\times 744}$. Given $\mathbf{Y}$, we first analytically determine the GFT basis, $\mathbf{U}$, using (4.14) in Section 4.4 and then recover the graph using the ADMM algorithm developed in Section

[1] https://github.com/BaolingShan/Temperature-dataset-in-Brittany-France



(a) Ours

(b) Saboksayr [43]

(c) Sar-ESA [35]

(d) Sar-TV[35]

(e) Egilmez [84], [85]

Figure 4.3: The learned graphs with different methods. The color of a node indicates the average temperature in $^{\circ}C$ measured at the node during all the periods of observation.

4.5. In the recovered graph, the vertices indicate the weather stations. The weight of an edge can then indicate the degree of similarity between the temperatures of two weather stations represented by the connected vertices.

Fig. 4.3 shows the graphs learned, where the average temperature of each weather station is color-coded. Fig. 4.3(a) corresponds to the proposed algorithm, where $K = 17$ is obtained as described in **Remark 3**, i.e., by first setting $K = N - 1$ to generate and sort $\|\mathbf{S}(i,:)\|$, i.e., $\|\mathbf{S}(1,:)\| \geq \|\mathbf{S}(2,:)\| \geq \cdots \geq \|\mathbf{S}(N,:)\|$, and then selecting the smallest K such that $\frac{\sum_{i=1}^{K}\|\mathbf{S}(i,:)\|}{\sum_{i=1}^{N}\|\mathbf{S}(i,:)\|} \geq 99\%$. The results of the existing algorithms are plotted in Figs. 4.3(b)–4.3(d). For Sar-ESA and Sar-TV, $K = 17$ is obtained in the same way as in our algorithm. Saboksayr's and Egilmez's do not rely on the value of K. For a fair comparison, we configure all graphs to have consistent sparsity, i.e., each graph has 90 edges, by fine-tuning the respective sparsity hyperparameters of the considered algorithms. We see that our algorithm provides more reasonable learning results, as it connects the weather stations with close av-

Table 4.3: The reconstruction accuracy of Saboksayr's (Sab.) [43], Sar-ESA [35], Sar-TV [35], Egilmez's [84], [85], and the proposed algorithm upon temperature dataset.

Methods	**Sab.**	**Sar-ESA**	**Sar-TV**	**Egilmez**	**Ours**
Accuracy	0.7227	0.8269	0.7343	0.7061	**0.9301**
Runtime (s)	15	203	156	37	23

erage temperatures and disconnects those with substantially different temperatures. By contrast, the existing approaches connect the weather stations excessively with substantially different average temperatures in Figs. 4.3(b), 4.3(d) and 4.3(e); or fail to connect some nearby weather stations with similar average temperatures in Fig. 4.3(c).

The reliability of the proposed algorithm is evaluated by using a greedy algorithm [10] to reconstruct the temperature data of some of the stations over a random period of time based on the learned Laplacian matrix and the observed signals of the other stations. The number of observable temperature signals (or weather stations) is set to be equal to the signal bandwidth K. In other words, we select 17 nodes to recover the signal waveforms of the other 15 nodes. Fig. 4.4(a) shows the ground-truth temperatures observed at Lorient station (the solid blue line) and the reconstructed version (the orange dash line). The Lorient station is one of the 15 unobserved stations. We see that the reconstructed signals are consistent with the ground-truth signals, indicating the reliability of the new graph-learning algorithm. For comparison, we also reconstruct the temperature signals by running Saboksayr's [43], Sar-ESA [35], Sar-TV [35], and Egilmez's [84], [85], as shown in Figs. 4.4(b)-4.4(e), where the reconstructed signals are less accurate. We adopt the coefficient of determination, denoted by R^2, to quantify the reconstruction accuracy of the algorithms. $R^2 = 1 - \|\mathrm{vec}(\hat{\mathbf{Y}}_{\mathrm{R}}) - \mathrm{vec}(\mathbf{Y}_{\mathrm{R}})\|_{\mathbf{2}}^{\mathbf{2}} / \|\mathrm{vec}(\hat{\mathbf{Y}}_{\mathrm{R}}) - \bar{y}_{\mathrm{R}}\|_2^2$. $\hat{\mathbf{Y}}_{\mathrm{R}}$ and $\mathbf{Y}_{\mathrm{R}}$ are the reconstructed signals and their ground-truths. $\bar{y}_{\mathrm{R}}$ is the average of the ground-truths. $\mathrm{vec}(\cdot)$ denotes vectorization. Table 4.3 shows our algorithm provides the best reconstruction accuracy with the second shortest runtime.

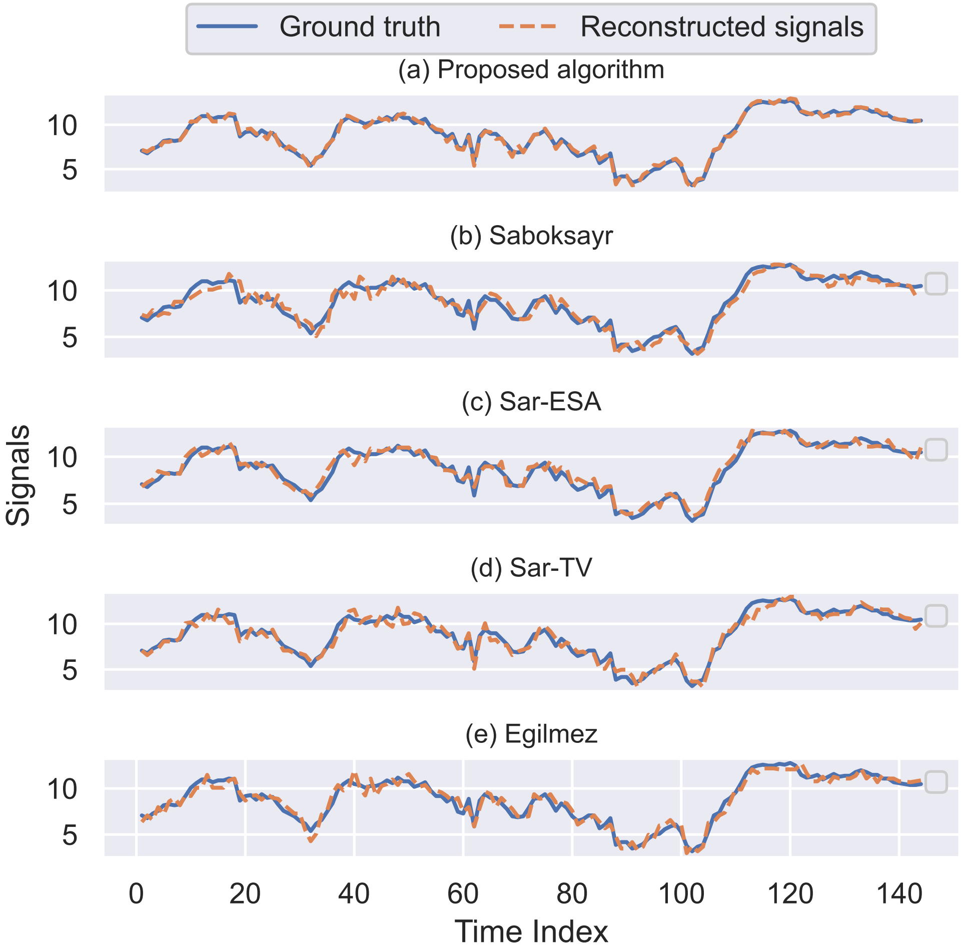


Figure 4.4: The reconstructed signals of temperatures at Louargat.

4.6.3 Attention-Deficit Hyperactivity Disorder (ADHD) Data

We apply the proposed algorithm to analyze brain functional networks. ADHD is a prevalent behavioral problem diagnosed with inattention and hyperactivity [130]. Latest neuroanatomical and neuropsychological studies indicated these behavioral disturbances are related to changes in functional connectivity among brain regions [132].

The considered ADHD dataset contains 42 right-handed, boy subjects with an average age of 13.2 years[2]. Without loss of generality, we randomly select subject 10 in this experiment. For comparison, typically developing (TD) subject 1 is selected in this experiment. We divide a brain into 90 ROIs using AAL template [131], where each ROI is represented by a node with 232 signals from the ADHD dataset. Therefore, the observed graph signal yield $\mathbf{Y} \in \mathbb{R}^{90\times 232}$. Following the proposed al-

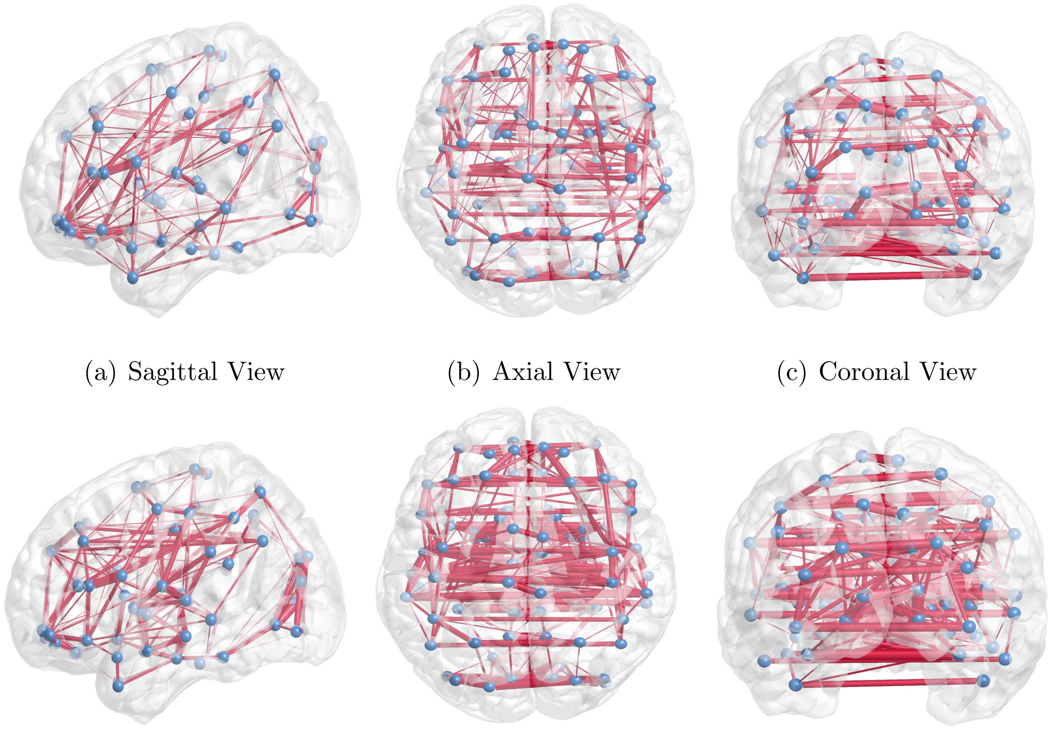
[2] The ADHD dataset is available at $https://www.nitrc.org/frs/?group\_id = 383$

gorithm, we can determine $K = 62$. By executing the algorithm, the brain network connecting the ROIs is constructed in the form of weighted adjacent coefficients. The brain network is depicted by the BrainNet Viewer toolbox [128].

As shown in Fig. 4.5, the differences of the brain functional network between the ADHD subject and the TD subject are visible from the sagittal, axial, and coronal views of the brain. We see that the ADHD subject exhibits decreased structural connectivity, compared to the TD subject. We also observe considerable asymmetry in the hemispheric brain anatomical network topology in the ADHD subject, as shown in Figs. 4.5(b), 4.5(c), 4.5(e), and 4.5(f). This confirms that the brain network topologies of the ADHD subject are disrupted in their distributed neural networks, which is likely related to inattention and hyperactivity. The observed decreased connectivity and increased asymmetry in the ADHD subject are consistent with the existing neuroanatomical studies [130]–[133].

We reconstruct the signals of a part of the brain network based on the learned Laplacian matrix and the signals of the other part of the brain network. The number of observable signals, i.e., the number of brain signal waveforms, is set as K. We select $K = 62$ nodes to recover the signal waveforms of the other 28 nodes. Fig. 4.6(a) plots the ground-truth ADHD brain signals at node 7 of subject 10 using the blue line, and the reconstructed signals using the orange line. Node 7 is not among the 62 observable nodes. The reconstructed signals concur with the ground truth, confirming the accuracy of our algorithm. We also reconstruct the ADHD signals by running the existing algorithms; see Figs. 4.6(b)-4.6(e). The existing algorithms are much less accurate than the proposed algorithm.

Table 4.4 quantifies the reconstruction accuracy of all 28 unobserved nodes using R^2. Our algorithm provides a much finer accuracy than the existing algorithms. Table 4.4 also compares the run time between the four algorithms. Our algorithm is substantially faster than Sar-ESA, Sar-TV, and Egilmez. Despite our algorithm taking slightly longer time than Saboksayr's, the latter provides much worse accuracy; see Fig. 4.6.



(a) Sagittal View (b) Axial View (c) Coronal View

(d) Sagittal View (e) Axial View (f) Coronal View

Figure 4.5: Connected brain networks from sagittal, axial and coronal views depicted by BrainNet Viewer. The blue nodes represent the ROIs, and the thickness of the red line is the strength of the connectivity between two ROIs. (a), (b), (c) The learned graphs based on the ADHD subject 10 with the proposed algorithm. (d), (e), (f) The learned graph based on the TD control subject 1.

Table 4.4: The reconstruction accuracy of Saboksayr's (Sab.) [43], Sar-ESA [35], Sar-TV [35], Egilmez's [84], [85], and the proposed algorithm upon all ADHD subjects.

Methods	**Sab.**	**Sar-ESA**	**Sar-TV**	**Egilmez**	**Ours**
Accuracy	0.7121	0.7981	0.7202	0.7004	**0.9315**
Runtime (s)	9	113	96	20	13

4.7 Conclusion

In this chapter, we developed a new graph learning technique to efficiently infer the graph structure of observed band-limited graph signals. A closed-form derivation was analytically derived for the graph GFT basis from the observed signals. With the GFT basis, we further applied ADMM to solve the eigenvalues of the graph Laplacian with substantially lower computational complexity. Simulations and experiments showed that our method outperforms the status quo in accuracy and efficiency.

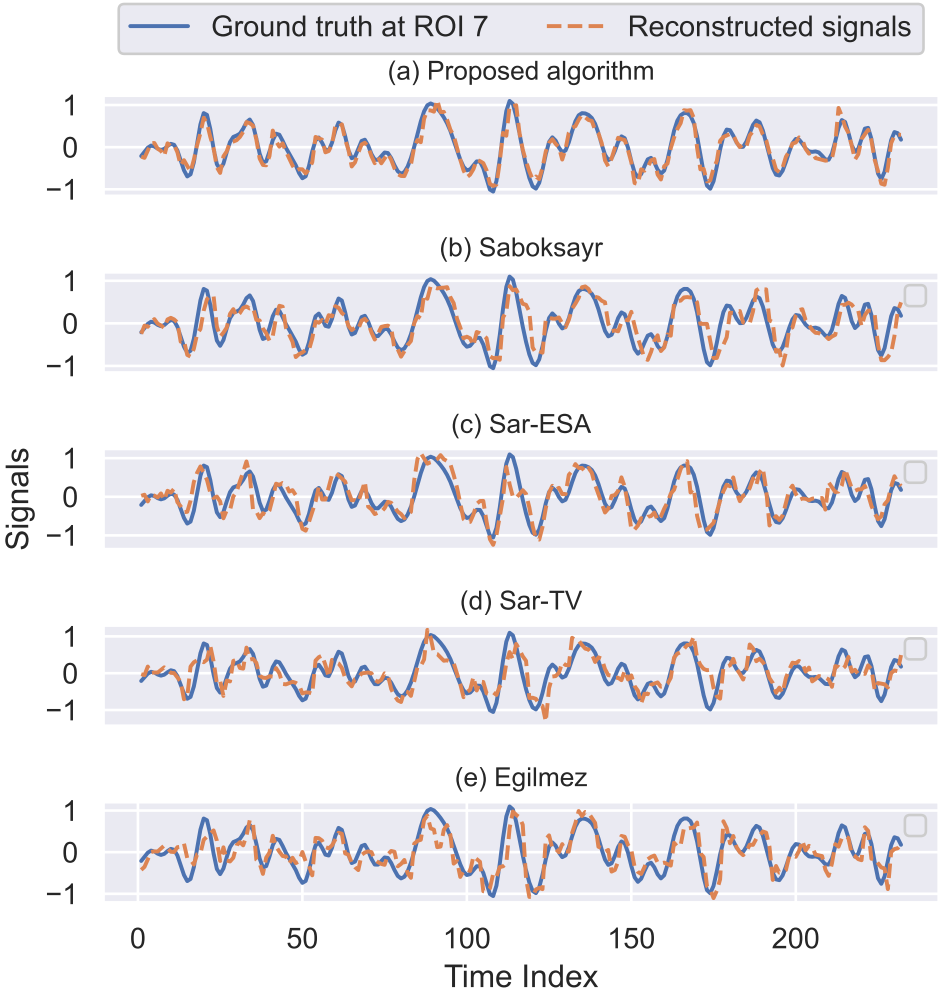


Figure 4.6: The reconstructed signals of the anatomical ROI 7 of subject 10.

Chapter 5

Novel Graph Topology Learning for Spatio-Temporal Analysis of COVID-19 Spread

5.1 Introduction

Global health, economic, and social challenges are escalating since the ongoing Coronavirus disease 2019 (COVID-19) pandemic. As of April 2022, Europe had 192.09 million confirmed cases and over two million deaths [1]. The SARS-CoV-2 virus has undergone numerous genetic changes since its discovery [138]. While some of these changes do not affect the virus's behavior, others may affect how easily it is transmitted. Changes beneficial to the virus tend to spread more quickly, which means that variants harboring them gradually replace other circulating variants [139]. In November 2020, SARS-CoV-2 Alpha was first identified in the United Kingdom, which was estimated to be 50% higher transmissibility than the original strain. From July 2021 to October 2021, SARS-CoV-2 Delta prevailed in Europe. The SARS-CoV-2 Omicron variant took over from the SARS-CoV-2 Delta variant in Europe in November 2021. Earlier studies demonstrated that Omicron can, to a degree, evade the protective effects of antibodies induced by vaccinations or natural

[1] https://coronavirus.jhu.edu/data

infections. Large portions of the European population are susceptible to infection, leading to sharp increases in COVID-19 cases and unprecedented community spread.

Comprehending the spatio-temporal characteristics of the virus transmission is the key to controlling the transmission of the pandemic. Studies show that the global spread of COVID-19 did not process uniformly [50], [140]. An outbreak's size and condition are influenced by the characteristics of virus spread [141]. Unfortunately, it is difficult to implement evidence-based policies for COVID-19 due to a lack of adequate evidence in policy-making and research [51]. While it is possible to estimate the growth rates of confirmed cases and deaths [52], the relationships between pairs of countries are still unknown as far as the COVID-19 development is concerned. Datasets about ongoing situations in different countries are likely to show spatial-temporal patterns since virus spread tends to follow geographic trends. A spatial-temporal analysis of confirmed COVID-19 cases may also shed light on its evolution. The record of pandemic evolution in Europe is known to be complex, variable, and non-linear. Consequently, it is essential to uncover hidden information about SARS-CoV-2 as new virus variants emerge.

One way to understand the spreading dynamic of the pandemic is to generate and analyze COVID-19 pandemic diffusion graph topologies with the graph-theoretic metrics [54]–[57]. In addition to illustrating spatial and temporal connections between places, spatio-temporal maps can potentially indicate changes in pandemic risks [105].

Existing studies have examined the spread of epidemics as a complex system by assessing the degree of correlation or synchronization between time-series data [59]. A deeper understanding of the spread dynamics of the new variants of SARS-CoV-2 requires new methods beyond assessing correlation or synchronization. There is a need to explore the latent structures among the data and reveal the relationships between different countries to understand the spatio-temporal spread of the virus.

This chapter aims to uncover the hidden knowledge that underpins the evolution of the pandemic, examine the underlying relationship among countries, and understand the spreading pattern of SARS-CoV-2 variants, e.g., by taking Europe as an example.

A new effective graph learning method is proposed to infer the graph Laplacian of the COVID-19 data, where we first obtain the closed-form expression of the graph Laplacian eigenvector, also known as GFT basis. Given the eigenvectors, we transform the inference of graph Laplacian to a readily solvable, convex problem of estimating its eigenvalues. With the graph Laplacian estimated, we perform an in-depth spatio-temporal exploration of COVID-19 data and shed insights into the COVID-19 spread in Europe.

5.2 Materials and System Model

The analysis is based on the open-access dataset of daily identified COVID-19 cases reported officially by different countries, territories, and regions, and published by the WHO[2]. The daily data for COVID-19 in European countries are updated every day. We collect the data from January 2020 to April 2022 and divide this period into four based on the statistics from the WHO, as shown in Fig. 5.1. The first period is the early stage of the pandemic outbreak, between March 2020 and October 2020, when the original strain of the virus dominated the spread. The second period is from November 2020 to May 2021, when the Alpha variant was dominant. The third stage is from June 2021 to October 2021, when the Delta variant broke out. The fourth stage is from November 2021 to April 2022, when the Omicron variant rapidly replaced the Delta and became the dominating variant in most European countries.

For each period, we analyze the SARS-CoV-2 time series data of the 44 European countries published by the WHO, by extracting a graph with 44 nodes from the data. The graph topology is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V} = 1, 2, \ldots, N$ denotes the set of N vertices, with $N = 44$ being the number of countries. The set of edges, denoted by $\mathcal{E}$, is a subset of $\mathcal{V} \times \mathcal{V}$. The weighted adjacency matrix of the graph $\mathcal{G}$, denoted by $\mathbf{W} \in \mathbb{R}^{N \times N}$, indicates the extent to which two countries are correlated with respect to COVID-19 spread. $W_{ij} = W_{ji} \neq 0$ for $\forall (i, j) \in \mathcal{E}$. Each node in $\mathcal{G}$ represents a European country and corresponds to the time series recording daily confirmed cases per million people in the country.

[2] https://covid19.who.int/WHO-COVID-19-global-data.csv

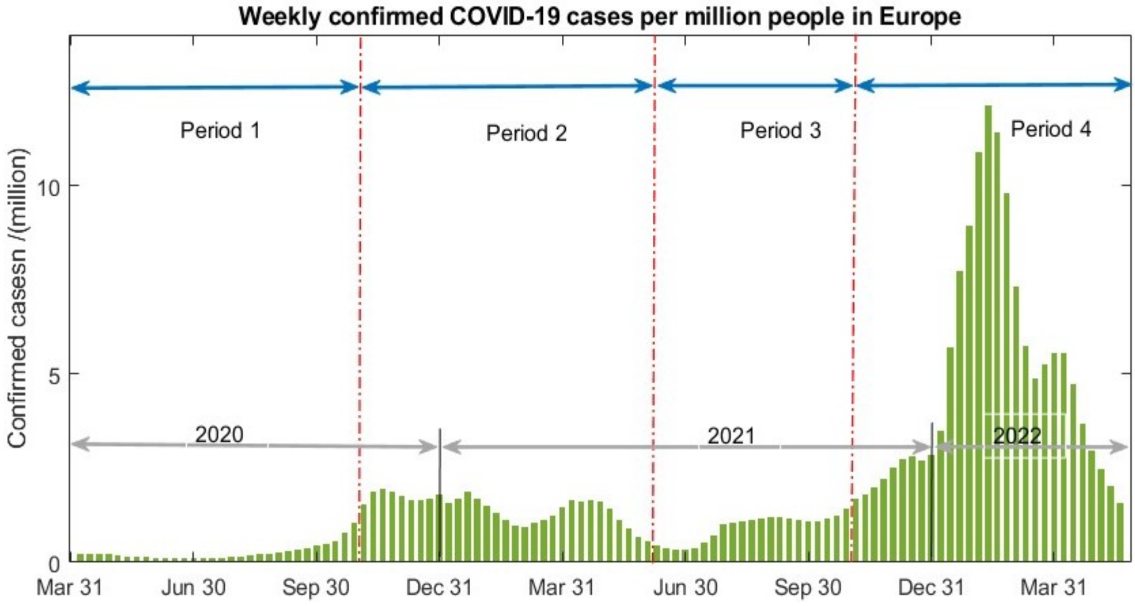


Figure 5.1: Weekly confirmed COVID-19 cases per million people.

For each of the periods, we use $\mathbf{x}_p \in \mathbb{R}^{N\times 1}, \forall p \in \{1, \cdots, P\}$ to denote the COVID-19 records of the N countries on the p-th day of the period, where P is the number of days in the period. The COVID-19 data of the European countries during the period are arranged in an $N \times P$ matrix, denoted by $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \in \mathbb{R}^{N\times P}$. Here, $\mathbf{X}$ is band-limited and its frequency-domain representation has finite bandwidth; in other words, the virus spreads across countries, rather than breaks out simultaneously in all countries.

To derive information about the underlying structure of $\mathcal{G}$, it is necessary to estimate the graph Laplacian $\mathbf{L}$. According to the definition in [84], graph Laplacian is a semi-definite matrix with positive elements along its main diagonal and non-positive elements anywhere else, which can be rewritten as:

$$\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T = \mathbf{U}\text{diag}(\boldsymbol{\lambda})\mathbf{U}^T. \tag{5.1}$$

Here, $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \cdots, \lambda_N)$ is a diagonal matrix consisting of the non-negative eigenvalues of the Laplacian, and $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_N]$ is an orthonormal matrix comprising the corresponding eigenvectors. According to [10], the GFT is the projection of $\mathbf{X}$ on $\mathbf{U}$. The GFT of the COVID-19 data $\mathbf{x}_p, \forall p \in \{1, \cdots, P\}$ on p-th day, denoted by $\mathbf{s}_p$, is given by

$$\mathbf{s}_p = \mathbf{U}^T\mathbf{x}_p. \tag{5.2}$$

Let $\mathbf{S} = [\mathbf{s}_1, \cdots, \mathbf{s}_p] \in \mathbb{R}^{N \times P}$. Then,

$$\mathbf{X} = \mathbf{US}. \tag{5.3}$$

With the sparsity of $\mathbf{s}_p, \forall p$, we define $\mathbf{S}$ as a K-block sparse matrix with rows consisting of multiple all-zero vectors. Here, K indicates the frequency-domain bandwidth of the COVID-19 data $\mathbf{X}$. K is obtained empirically in prior or enumerated to find its proper value [142]. The set $\mathcal{B}_K$ contains all K-block sparse matrices, defined as $\mathcal{B}_K \triangleq \{\mathbf{S} \in \mathbb{R}^{N \times P}, \mathbf{S}(i,:) = \mathbf{0}, \forall i \notin \mathcal{K} \subseteq \mathcal{V},\ K = |\mathcal{K}|\}$. Here, $\mathbf{S}(i,:)$ denotes the i-th row of $\mathbf{S}$, and the set $\mathcal{K} \subseteq \mathcal{V}$ collates the indexes to the K most significant frequency components of the $\mathbf{X}$.

5.3 Proposed Graph inference for COVID-19 Spread Analysis

COVID-19 data analysis plays a crucial role in identifying the most influential countries or regions in the spread of the virus and understanding how the virus spreads among countries. In this section, we develop a new graph learning method, which accurately and efficiently extracts the underlying graph topological information of the COVID-19 data, reveals the fine-grained similarity (or correlation) between different countries in the virus spread process, and helps identify the most influential countries that present strong representativeness. More specifically, the technique extracts the graph Laplacian matrix $\mathbf{L}$ of the COVID-19 data in each period by first deriving $\mathbf{U}$ and then solving $\mathbf{\Lambda}$ efficiently using convex optimization techniques. By applying the graph extracted and centrality measures, we identify the influential countries that it is significant in the study of the COVID-19 spread.

5.3.1 Graph Topology Extraction

First, we estimate $\mathbf{L}$ and hence, the graph struucture $\mathcal{G}$ substantiating the COVID-19 data $\mathbf{X}$. By taking into account the band-limitedness of $\mathbf{X}$, we formulate the problem as

$$\min_{\mathbf{L},\mathbf{U}\in\mathbb{R}^{N\times N},\mathbf{S}\in\mathbb{R}^{N\times P}} \|\mathbf{X}-\mathbf{U}\mathbf{S}\|_F^2 + f(\mathbf{L},\mathbf{X}) \tag{5.4a}$$

$$\text{s.t.} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}_N, \tag{5.4b}$$

$$\mathbf{S} \in \mathcal{B}_K, \tag{5.4c}$$

$$\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T,\ \mathbf{L} \in \mathbb{L},\ \mathrm{tr}(\mathbf{L}) = N, \tag{5.4d}$$

$$\mathbf{u}_1 = \frac{1}{\sqrt{N}}\mathbf{1}. \tag{5.4e}$$

The objective (5.4a) is composed of two terms. The first accounts for data consistency by penalizing any discrepancy between **US** and **X** using a quadratic loss. The second term provides a regularization function. According to [18] and [35], we set

$$f(\mathbf{L},\mathbf{X}) = \mathrm{tr}(\mathbf{X}^T\mathbf{L}\mathbf{X}) + \alpha \left\|\mathrm{vec}(\mathbf{L})\right\|_1.$$

Constraint (5.4b) guarantees that the matrix **U** is unitary, satisfying the decomposition in (5.1); constraint (5.4c) enforces that the GFT coefficient matrix **S** is K-block sparse; constraint (5.4d) ensures that **L** complies with the requirement of a legitimate graph Laplacian, and $\mathbb{L}$ contains all legitimate candidates for **L** [84]:

$$\mathbb{L} = \{\mathbf{L} \succeq \mathbf{0} | \mathbf{L}\mathbf{1} = \mathbf{0}, L_{ij} = L_{ji} \leq 0, \forall i \neq j\}. \tag{5.5}$$

According to $\mathbf{L}\mathbf{1} = \mathbf{0}$ in (5.5), we conclude that 0 is an eigenvalue of **L** and corresponds to the eigenvector $\mathbf{u}_1 = \frac{1}{\sqrt{N}}\mathbf{1}$, i.e., the first column of **U**; see (5.4e).

Remark 5 *To address the non-convexity of (5.4) caused by the non-convex orthonormality constraint in (5.4b) and the sparsity constraint in (5.4c), we decouple and solve* (5.4) *in two phases. Given the COVID-19 data* **X***, we first estimate the GFT basis* **U** *by minimizing* $\|\mathbf{X}-\mathbf{U}\mathbf{S}\|_F^2$ *subject to* $\mathbf{U}^T\mathbf{U} = \mathbf{I}_N$*,* $\mathbf{S} \in \mathcal{B}_K$ *and* $\mathbf{u}_1 = \frac{1}{\sqrt{N}}\mathbf{1}$*. In the second step, we estimate the eigenvalues* $\boldsymbol{\Lambda}$ *by minimizing the regularizer* $\mathrm{tr}(\mathbf{X}^T\mathbf{L}\mathbf{X}) + \alpha \left\|vec(\mathbf{L})\right\|_1$ *with the obtained* **U***.*

Extraction of Eigenvectors

Starting with the GFT basis, **U**, provides a way to identify the intrinsic structure in the COVID-19 data that are related to the underlying pandemic network, even without the *a-priori* information of graph Laplacian **L**. To estimate **U** from **X** satisfies the definition of GFT, i.e., $\mathbf{X} = \mathbf{U}\,\mathbf{S}$.

By utilizing the orthonormality property of $\mathbf{U}$ in (5.4b), we have $\|\mathbf{X}-\mathbf{US}\|_F^2 = \left\|\mathbf{U}^T\mathbf{X}-\mathbf{S}\right\|_F^2$. We start with the first part of problem (5.4), as given by [35, eq. 8]

$$\min_{\mathbf{U}\in\mathbb{R}^{N\times N},\mathbf{S}\in\mathbb{R}^{N\times P}} \left\|\mathbf{U}^T\mathbf{X}-\mathbf{S}\right\|_F^2, \text{ s.t. (5.4b), (5.4c), (5.4e).} \tag{5.6}$$

Despite the convex objective function, problem (5.6) is non-convex due to the orthonormality in (5.4b) and the sparsity in (5.4c). Since both $\mathbf{U}$ and $\mathbf{S}$ are unknown, we reorganize (5.6) as

$$\min_{\mathbf{U}\in\mathbb{R}^{N\times N},} \min_{\mathbf{S}\in\mathcal{B}_K} \sum_{i=1}^{N} \left\|\mathbf{u}_i^T\mathbf{X}-\mathbf{S}(i,:)\right\|_2^2, \text{ s.t. (5.4b), (5.4e),} \tag{5.7}$$

which can be rewritten as

$$\min_{\mathbf{U}\in\mathbb{R}^{N\times N}} \left(\min_{\mathbf{S}\in\mathcal{B}_K} \sum_{i\in\mathcal{K}} \left\|\mathbf{u}_i^T\mathbf{X}-\mathbf{S}(i,:)\right\|_2^2 + \sum_{i\notin\mathcal{K}} \left\|\mathbf{u}_i^T\mathbf{X}\right\|_2^2 \right)$$
$$\text{s.t.} \quad \text{(5.4b), (5.4e).} \tag{5.8}$$

By closely analyzing the objective function of (5.8), it can be noticed that the optimal $\mathcal{K}$ comprises K most significant entries of $\{\left\|\mathbf{u}_i^T\mathbf{X}\right\|\}_i^N$, and satisfies

$$\mathbf{S}(i,:) = \begin{cases} \mathbf{u}_i^T\mathbf{X}, & \text{if } \ i\in\mathcal{K}; \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{5.9}$$

Therefore, the objective of (5.8) is reduced to only include the $(N-K)$ smallest entries of $\{||\mathbf{u}_i^T\mathbf{X}||\}_i^N$, after optimizing $\mathbf{S}$ to suppress $\sum_{i\in\mathcal{K}} \left\|\mathbf{u}_i^T\mathbf{X}-\mathbf{S}(i,:)\right\|_2^2$ using (5.9). To minimize this objective with respect to $\mathbf{S}$, we aim to seek the optimal $\mathbf{U}$, represented as $\mathbf{U}^*$, in (5.6).

Substitute (5.9) into the objective of (5.8). Then , problem (5.6) can be written as

$$\begin{aligned}\mathbf{U}^* &= \arg\min_{\mathbf{U}} \sum_{i\notin\mathcal{K}} \left\|\mathbf{u}_i^T\mathbf{X}\right\|_2^2 = \arg\min_{\mathbf{U}} \left\|\mathbf{U}_{\mathcal{K}^c}^T\mathbf{X}\right\|_F^2 \\ &= \arg\max_{\mathbf{U}} \left\|\mathbf{U}_{\mathcal{K}}^T\mathbf{X}\right\|_F^2, \end{aligned} \tag{5.10}$$

where $\mathcal{K}^c$ denotes the complementary set of $\mathcal{K}$, i.e., $\mathcal{K}^c = \mathcal{V}\setminus\mathcal{K}$; and the matrices $\mathbf{U}_{\mathcal{K}}$ and $\mathbf{U}_{\mathcal{K}^c}$ collate the column-vectors of $\mathbf{U}$ with indexes collected in $\mathcal{K}$ and $\mathcal{K}^c$, respectively.

Table 5.1: Topological characteristics of the learned complex networks.

Metric	**Formula**	**Description**
Degree centrality	$C_d(n_i) = \frac{\sum_1^j e_{ij}}{N-1}$	The number of edges directed towards node i.
Closeness centrality	$C_c(n_i) = \frac{N-1}{\sum_{i \neq j} d_{ij}}$	The average length of the shortest paths from node i to the rest of the nodes.
Betweenness centrality	$C_b(n_i) = \frac{\sum\limits_{i,j \neq v} \frac{\sigma_{ij}(v)}{\sigma_{ij}}}{(N-1)(N-2)}$	The frequency of a node serves as an intermediate relay along the shortest paths.
Average path length	$\frac{\sum_{i \neq j} d_{ij}}{N(N-1)}$	The average length of all the shortest paths in a graph.
Global efficiency	$\frac{N(N-1)}{\sum_{i \neq j} d_{ij}}$	The efficiency of information exchange between all node pairs.

Despite the non-convexity of (5.10), the goal of (5.10) is to identify the K-dimensional subspace in which the COVID-19 data $\mathbf{X}$ has the largest orthogonal projection; i.e.,

$$\arg\max_{\mathbf{U}} \ \left\|\mathbf{U}_{\mathcal{K}}^T \mathbf{X}\right\|_F^2 = \arg\max_{\mathbf{U}} \ \operatorname{tr}\left(\mathbf{P}_{\mathbf{U}_{\mathcal{K}}} \mathbf{X}\mathbf{X}^T\right), \tag{5.11}$$

where $\mathbf{P}_{\mathbf{U}_{\mathcal{K}}} = \mathbf{U}_{\mathcal{K}} \mathbf{U}_{\mathcal{K}}^T$ is the orthogonal projector onto the subspace spanned by $\mathbf{U}_{\mathcal{K}}$. Using (5.11), we reformulate the problem (5.6) as

$$\mathbf{U}^* = \arg\max_{\mathbf{U}} \operatorname{tr}\left(\mathbf{P}_{\mathbf{U}_{\mathcal{K}}} \mathbf{X}\mathbf{X}^T\right), \quad \text{s.t.} \quad (5.4\text{e}). \tag{5.12}$$

Theorem 2 *By examining the two cases of* $\mathbf{u}_1 \notin \mathbf{U}_{\mathcal{K}}$ *and* $\mathbf{u}_1 \in \mathbf{U}_{\mathcal{K}}$*, the optimal solution to problem (5.6), denoted by* $\mathbf{U}^* = [\mathbf{U}_{\mathcal{K}}^*, \mathbf{U}_{\mathcal{K}^c}^*]$*, can be obtained as*

$$\mathbf{U}^* = \text{eigen}\left[\left(\mathbf{I} - \mathbf{u}_1 \mathbf{u}_1^T\right) \mathbf{X}\mathbf{X}^T \left(\mathbf{I} - \mathbf{u}_1 \mathbf{u}_1^T\right)^T\right]. \tag{5.13}$$

Extraction of Eigenvalues

Given the $\mathcal{K}$-band-limited COVID-19 data with the optimal $\mathbf{U}^*$ gained from (5.13), the graph Laplacian $\mathbf{L}$ is written as

$$\mathbf{L} = [\mathbf{U}_{\mathcal{K}}, \mathbf{U}_{\mathcal{K}^c}] \begin{bmatrix} \mathbf{\Lambda}_{\mathcal{K}} & \\ & \mathbf{\Lambda}_{\mathcal{K}^c} \end{bmatrix} [\mathbf{U}_{\mathcal{K}}, \mathbf{U}_{\mathcal{K}^c}]^T . \tag{5.14}$$

where $\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_{\mathcal{K}} & \\ & \mathbf{\Lambda}_{\mathcal{K}^c} \end{bmatrix}$. By plugging (5.14), $\text{tr}(\mathbf{X}^T\mathbf{L}\mathbf{X})$ is written as

$$\begin{aligned} \text{tr}(\mathbf{X}^T\mathbf{L}\mathbf{X}) &= \text{tr}(\mathbf{X}^T(\mathbf{U}_{\mathcal{K}}\mathbf{\Lambda}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T)\mathbf{X} + \mathbf{X}^T(\mathbf{U}_{\mathcal{K}^c}\mathbf{\Lambda}_{\mathcal{K}^c}\mathbf{U}_{\mathcal{K}^c}^T)\mathbf{X}) \\ &= \text{tr}(\mathbf{S}_{\mathcal{K}}^T\mathbf{\Lambda}_{\mathcal{K}}\mathbf{S}_{\mathcal{K}}). \end{aligned} \tag{5.15}$$

Problem (5.4) becomes

$$\begin{aligned} \min_{\mathbf{\Lambda}_{\mathcal{K}}, \mathbf{\Lambda}_{\mathcal{K}^c}, \mathbf{L}} \quad & \text{tr}(\mathbf{S}_{\mathcal{K}}^T\mathbf{\Lambda}_{\mathcal{K}}\mathbf{S}_{\mathcal{K}}) + \alpha \left\|\text{vec}(\mathbf{L})\right\|_1 \\ \text{s.t.} \quad & \mathbf{L} = [\mathbf{U}_{\mathcal{K}}, \mathbf{U}_{\mathcal{K}^c}] \begin{bmatrix} \mathbf{\Lambda}_{\mathcal{K}} & \\ & \mathbf{\Lambda}_{\mathcal{K}^c} \end{bmatrix} [\mathbf{U}_{\mathcal{K}}, \mathbf{U}_{\mathcal{K}^c}]^T , \\ & \mathbf{\Lambda}_{\mathcal{K}} \succeq \mathbf{0}, \mathbf{\Lambda}_{\mathcal{K}^c} \succeq \mathbf{0}, \\ & \mathbf{L}\mathbf{1} = \mathbf{0}, \\ & \text{tr}(\mathbf{L}) = N, \\ & L_{ij} = L_{ji} \leq 0, \forall i \neq j. \end{aligned} \tag{5.16}$$

Since its objective and constraints are convex or affine, problem (5.16) is convex and can be effectively addressed using by CVX toolboxes. With $\mathbf{U}$ and $\mathbf{\Lambda}$ obtained, we can obtain the graph Laplacian $\mathbf{L}$ underlying the European COVID-19 data using (5.1).

5.3.2 Influential Country Identify

Next, given the graph topology $\mathbf{L}$ underlying the COVID-19 data and indicating the propagation of the virus, we proceed to estimate the spread pattern of the four variants among the European countries. As shown in Table 5.1, three node-level metrics, including degree centrality [143], closeness centrality [144], and betweenness centrality [145], are used to measure the influence of individual countries in the

COVID-19 spread, where d_{ij} represents the shortest distance between nodes i and j in the extracted graph, σ_{ij} is the total amount of shortest paths between nodes i and j, and $\sigma_{ij}(v)$ denotes the number of these paths through node v.

- Degree centrality measures the number of connections a node has, helping identify the most connected nodes to the rest of the pandemic networks [143].
- Closeness centrality measures the inverse of the sum of the distances between a node and all other nodes in the network, which helps to identify nodes that are central and easily reachable within the network [144].
- Betweenness centrality quatify the importance of a node in maintaining the shortest paths between other nodes in the network, helping to identify nodes that play a critical role in connecting different parts of the network [145].

The higher centrality a country has, the more influential it is and the more attention it deserves. In other words, the countries ranked high in terms of the centrality measures are likely to present the important COVID-19 spread patterns.

Many other existing methods, such as node embeddings [146], DeepWalk [147], spectral clustering [148], and influence maximization [149], aimed to efficiently find influential nodes in large-scale graphs, e.g., social networks with thousands or even millions of nodes, often still based on the above classical centrality measures. Nevertheless, the graph considered consists of only $N = 44$ vertices (for 44 European countries). Computational complexity is less of a concern.

We also take two network-level metrics in Table 5.1, i.e., average path length [150] and global efficiency [151], to explore the spread of the pandemic.

- Average path length measures the average number of hops needed to get from one node to another node in the network [150]. A short average path length indicates a highly connected network, contributing to the fast spread of the pandemic [152].
- Global efficiency measures the average inverse shortest path length between all pairs of nodes, indicating how quickly the virus can spread [151]. A high

global efficiency indicates a dense and well-connected network with fast virus propagation, while a low global efficiency indicates a fragmented and poorly connected network deterring the virus propagation.

5.4 Method Assessment and Results

In this section, we first experimentally validate the superiority of the proposed technique to existing approaches in graph learning accuracy of the COVID-19 data. Then, we use the technique to conduct an in-depth analysis of COVID-19 data, and shed different insights into pandemic spread from existing techniques. The analysis is based on the open-access WHO dataset of daily identified COVID-19 cases in the 44 European countries.

Apart from the proposed method, we evaluate the state-of-the-art solutions: Saboksayr's algorithm [43], Sardellitti's TV algorithm [35], Sardellitti's ESA algorithm [35], and Humbert's algorithm [36].

- Saboksyr' algorithm [19]: This is a scalable and time-efficient primal-dual algorithm that learns the topological structures of time series represented by the weighted adjacency matrices of graphs. However, this method has no explicit generative model for the observations. In other words, the model's accuracy may not be adequate for numerous real-world datasets that exhibit localized behaviors or exhibit piecewise smoothness.

- Sardellitti's TV graph learning algorithm [35]: The approach involves a two-step scheme: (a) learning the orthonormal and sparse transform of the data using AO, and (b) inferring the Laplacian from the sparsifying transform using convex optimization. The algorithm is reasonably computationally efficient by exploiting convex optimization techniques. However, the effectiveness of the overall process is compromised due to the AO-based approximation in the first step, which penalizes the fidelity of the orthogonal sparsifying transform.

- Sardellitti's ESA graph learning algorithm [35]: Different from Sardellitti's TV graph learning algorithm, this algorithm utilizes the information of the

GFT coefficient matrix of the first step in the second step, where the graph Laplacian is recovered from the sparsifying transform and the GFT coefficients using convex optimization.

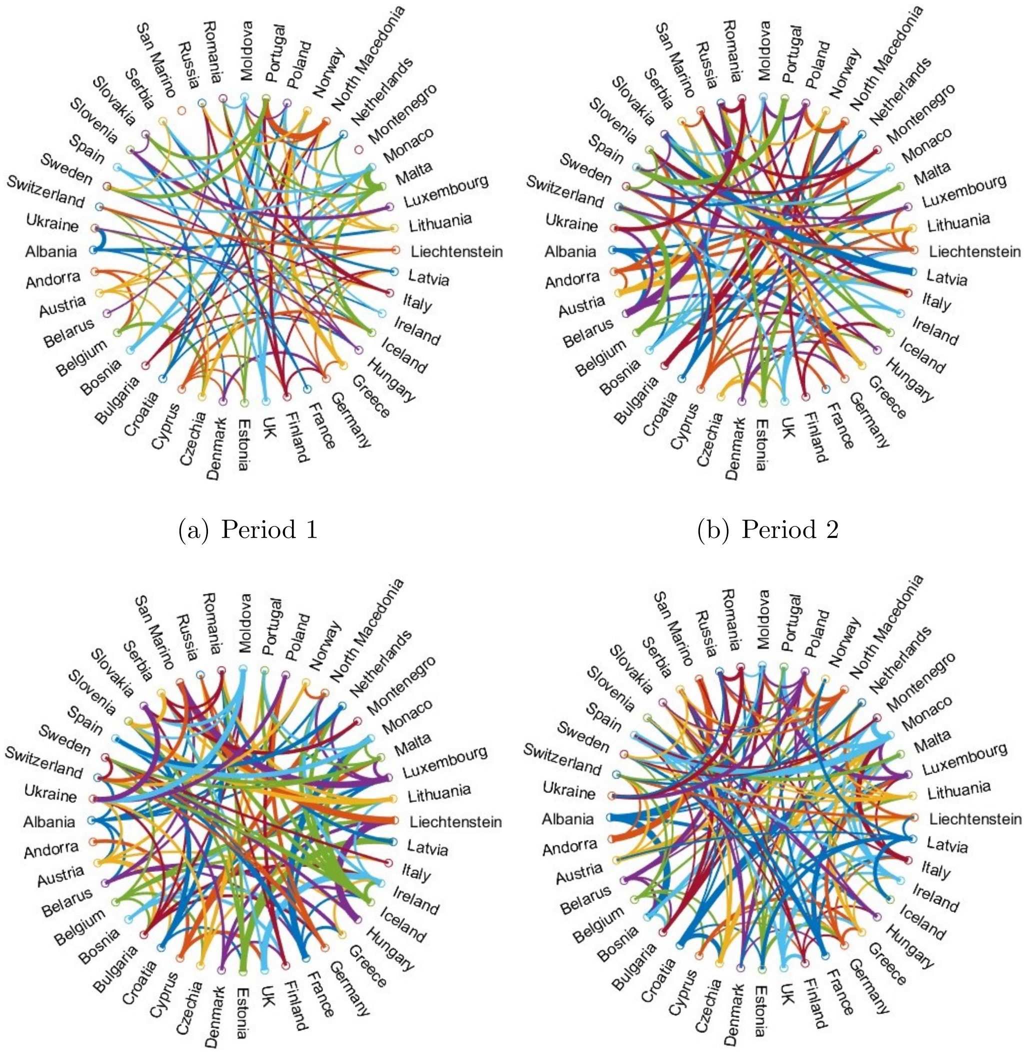
- Humbert's algorithm [36]: This is another AO-based algorithm runs Riemannian manifold gradient descent and linear cone programs in an alternating fashion. However, only suboptimal solutions can be obtained using the AO method. The computational cost is also high.

Apart from the aforementioned advanced graph learning techniques, we also evaluate our proposed algorithm in comparison to the state-of-the-art graph neural network (GNN) [153] when assessing the accuracy of the algorithm. The GNN consists of multiple hidden layers with 50 hidden units per layer. In the training stage, the input to the GNN includes the training data and the weighted correlation matrix of the training set. By contrast, the training set serves as the input for the graph learning algorithms.

5.4.1 Graph Learning-based Analysis of COVID-19 Data

Fig. 5.2 provides the pandemic spread networks of 44 European countries over the four different periods obtained by the proposed algorithm, where the parameters of the algorithm are $K = 26$ and $\alpha = 1$ decided in the way delineated at the beginning of Section 5.4.2. The thickness of an edge measures the similarity of the COVID-19 spread between two countries. The virus spreads in the two countries are more likely to be related if the edge is thicker. The density of the edges indicates the extent to which the COVID-19 spread among countries. It is observed in Fig. 5.2 that the virus spreads are increasingly related among the European countries from Period 1 to Period 4. Not only did the spreads increase between the countries, but the virus spread increasingly widely across more countries.

To better illustrate the correlation of the COVID-19 spread between the European countries, Fig. 5.3 plots the weighted adjacency matrices of the graphs extracted from the COVID-19 data by the proposed algorithm. In the figure, the 44 European countries are sorted alphabetically from Albania to Ukraine along the x- and y-



(a) Period 1

(b) Period 2

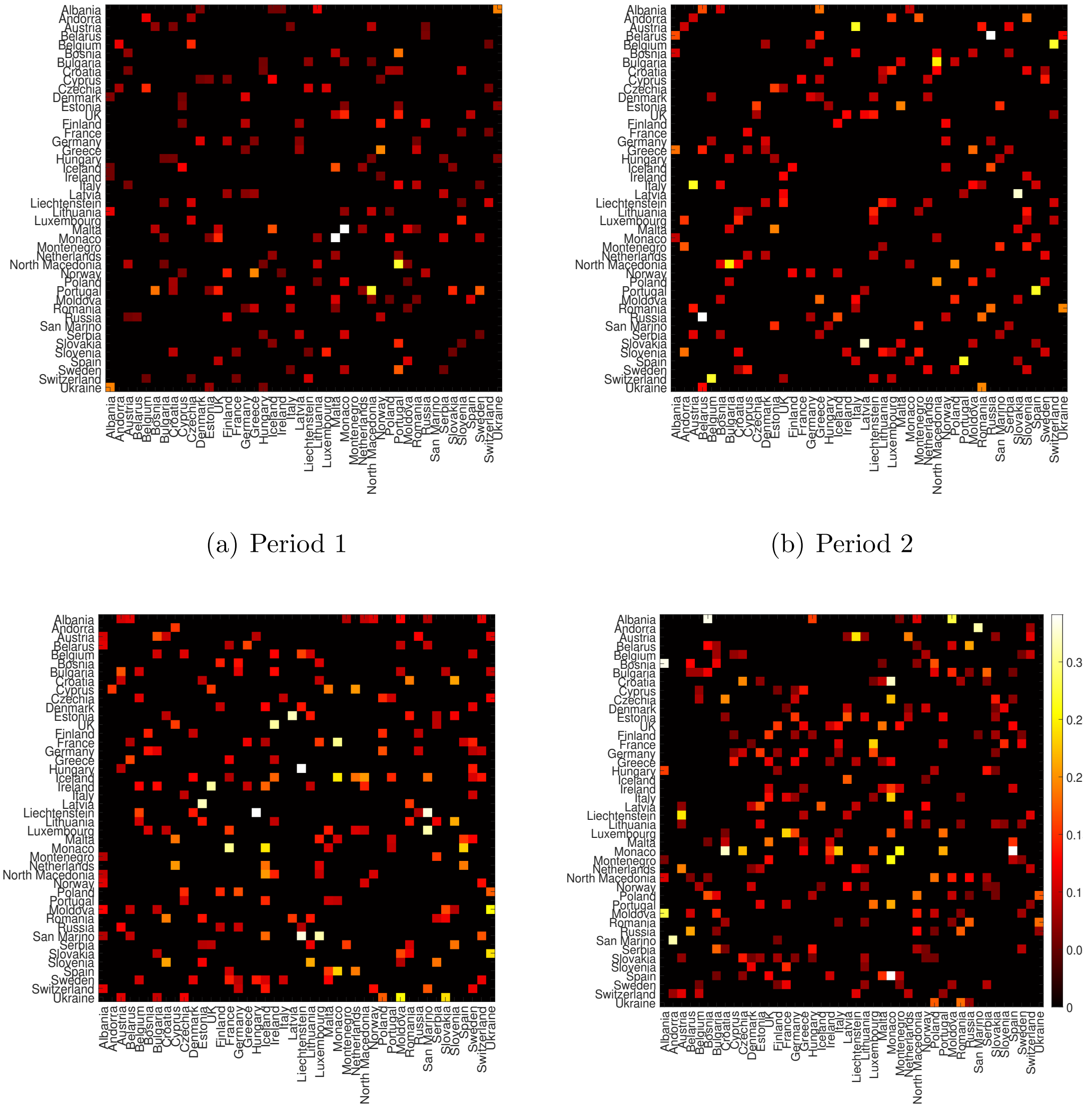
(c) Period 3

(d) Period 4

Figure 5.2: The learned graph of the COVID-19 spread in the 44 European countries during different periods.

axes. The intensity of the color at each pixel stands for the extent of the correlation between the two countries associated with the pixel.

For example, the pixel corresponding to Greece and Norway is lighter than others in Fig. 5.3(a), indicating that Greece and Norway are highly correlated in Period 1. Likewise, Russia and Belarus are highly correlated in Period 2 in Fig. 5.3(b). Nevertheless, the number of light-colored pixels increases overall in both Periods 3 and 4 in Figs. 5.3(c) and 5.3(d), indicating that the Delta and Omicron variants have higher and stronger propagation characteristics in Europe, which is consistent with the finding made in Fig. 5.2.



(a) Period 1

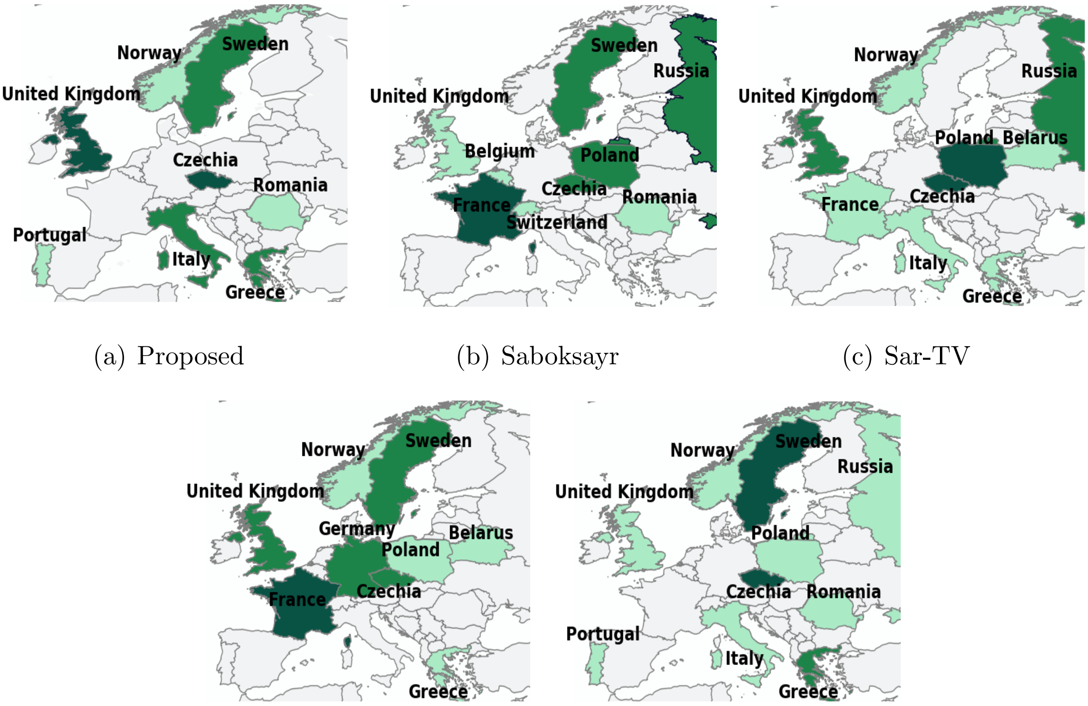
(b) Period 2

(c) Period 3

(d) Period 4

Figure 5.3: The weighted matrix of the learned graph of the spread of COVID-19 in 44 European countries during different periods.

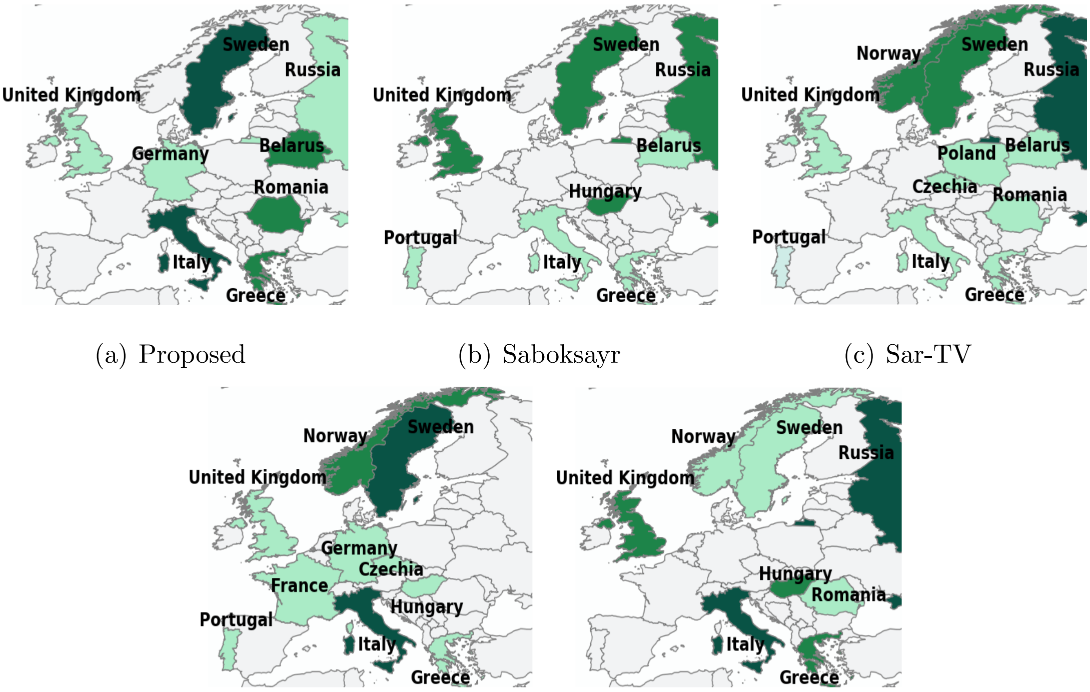
Figs. 5.4–5.7 visualize the top 5 countries that are identified to have been the most influential in the process of the COVID-19 virus spread in Europe, using the proposed approach based on the aforementioned three node-level metrics, i.e., degree centrality, closeness centrality, and betweenness centrality. A darker color indicates a country identified by more centrality measures to be among the top 5 most influential countries. For example, Czechia was influential during Period 1 in the sense of all three centrality measures. This makes sense since the different centrality measures are closely related in nature [105].



(a) Proposed (b) Saboksayr (c) Sar-TV

(d) Sar-ESA (e) Humbert

Figure 5.4: Influential countries identified during Period 1.

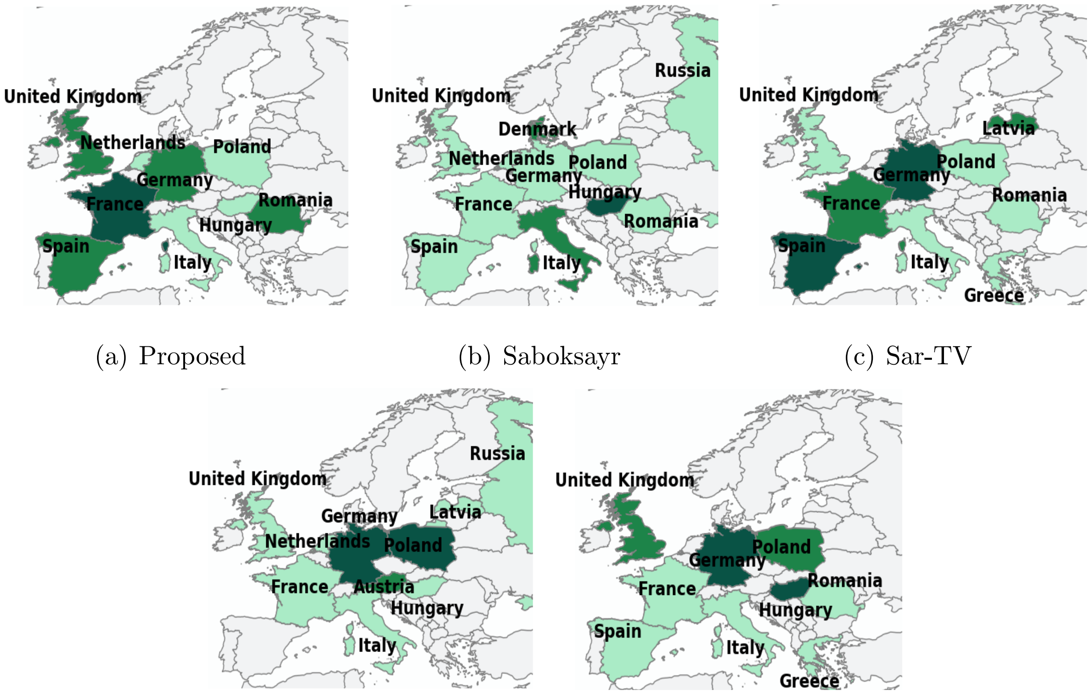


(a) Proposed (b) Saboksayr (c) Sar-TV

(d) Sar-ESA (e) Humbert

Figure 5.5: Influential countries identified during Period 2.

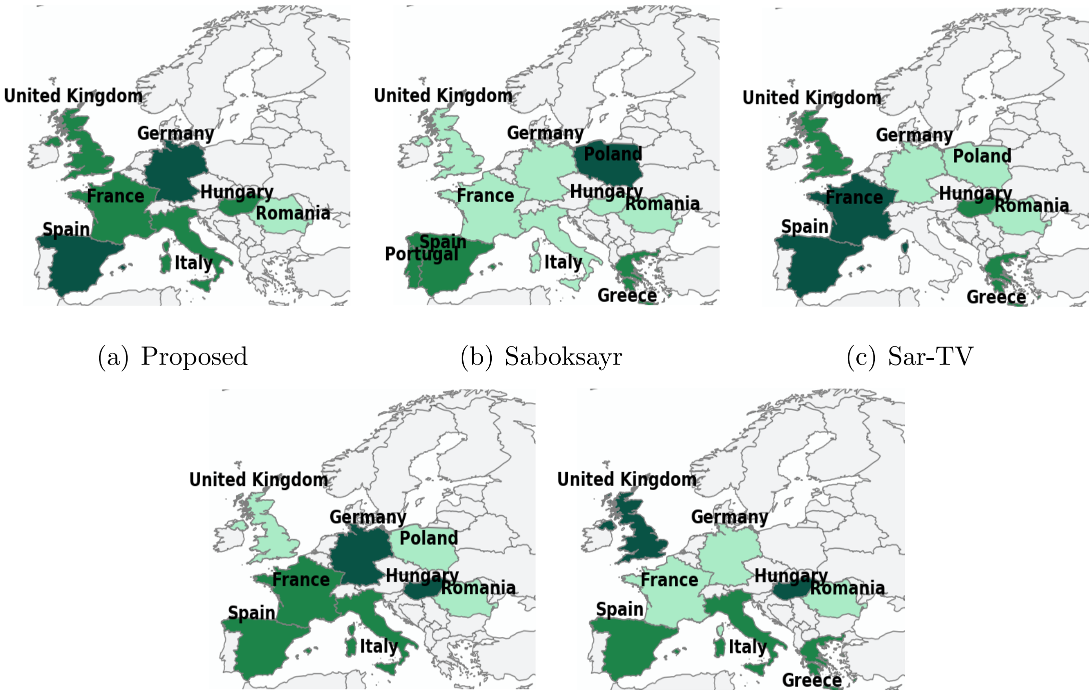
It is obvious in Figs. 5.4–5.7 that the proposed algorithm identifies a different set of the most influential European countries in the COVID-19 spread, compared to the



(a) Proposed (b) Saboksayr (c) Sar-TV

(d) Sar-ESA (e) Humbert

Figure 5.6: Influential countries identified during Period 3.



(a) Proposed (b) Saboksayr (c) Sar-TV

(d) Sar-ESA (e) Humbert

Figure 5.7: Influential countries identified during Period 4.

current advenced graph learning methods. Particularly, our new algorithm helps identify a small and concentrated set of influential countries in every period of

COVID-19 spread; i.e., a country is more likely to be associated with multiple centrality measures. In other words, the influence of a country is more likely to be manifested through multiple measures. Here, the parameters of each method are separately tested and optimized, according to their individual settings.

Fig. 5.8 quantitatively evaluates how different the top 5 most influential countries are identified by the different algorithms. Specifically, we vectorize the 15 most important countries identified using each of the considered algorithms based on the three centrality measures. The similarity between the 15-element vectors produced by any two of the considered graph learning algorithms, measured by the cosine distance $\frac{\mathbf{V}_1^T\mathbf{V_2}}{|\mathbf{V}_1|\cdot|\mathbf{V}_2|}$, quantifies the similarity between the algorithms, where $\mathbf{V}_1$ and $\mathbf{V}_2$ are the two 15-element column vectors, and $|\cdot|$ stands for the norm.

As shown in Fig. 5.8, our new graph learning algorithm yields the highest similarity to Humbert's [36] in terms of their identified important countries (under three centrality measures), followed by Sar-ESA [35], Sar-TV [35], and Saboksayr's [43]. The similarities of the proposed algorithm to the existing algorithms are consistent with the graph learning (and reconstruction) accuracy of the algorithms, as will be shown in Fig. 5.12. Note that the ground truth regarding the most important countries is unavailable in practice. Given the best graph learning accuracy of the proposed algorithm and the consistent rankings between the accuracies and the similarities of the existing algorithms, it is reasonable to conclude that the countries identified by the proposed algorithm are more accurate and can contribute to more effective study and response to the pandemic.

Fig. 5.9 plots the average path length and global efficiency of the graph recovered by the proposed graph learning algorithm in the four periods of the COVID-19 pandemic. It is observed that the average path length decreases while the global efficiency increases during the four periods. The Omicron variant (i.e., Period 4) corresponds to the shortest average path length and the highest global efficiency, indicating that the Omicron variant has a higher level of global reachability and infectivity. In contrast, the original strains in the early stage of the pandemic, i.e., Period 1, have higher average path lengths and smaller global efficiencies. This is consistent with the finding in Figs. 5.2 and 5.3. The reason can be that during

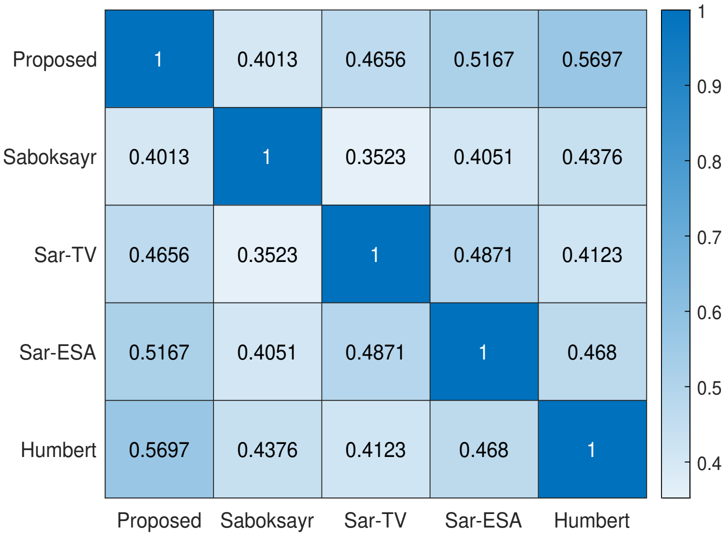


Figure 5.8: The correlation of different algorithms.

Period 1, the countries responded to the outbreak with stay-at-home or workplace closure, effectively slowing down the increase in confirmed cases.

5.4.2 Accuracy Validation of Proposed Graph Learning

Without the ground truth of the graphs underlying the COVID-19 data, we resort to assessing the learning accuracy of the proposed algorithm by obfuscating part of the data and assessing the reconstruction accuracy of this part of data based on the learned graphs and the rest of the data.

Suppose that the number of observable countries is K $(K \leq N)$, i.e., the signal bandwidth. Based on the inferred graphs, e.g., those in Fig. 5.3, and the COVID-19 data of K randomly selected European countries, we reconstruct the quantity of identified cases per million population in the remaining $(N-K)$ countries. The recovered graph signals, denoted by $\hat{\mathbf{x}}_p$, can be obtained as [44]

$$\hat{\mathbf{x}}_P = \mathbf{U}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T\mathbf{\Psi}^T\mathbf{\Psi}\mathbf{D}^2\mathbf{\Psi}^T\mathbf{y}_P, \tag{5.17}$$

where $\mathbf{y}_p \in \mathbb{R}^K$ is sampled $K \times 1$-dimensional COVID-19 data on the p-th day, which is chosen from $\mathbf{x}_p$ randomly and independently [44]. $\mathbf{\Psi} \in \mathbb{R}^{K\times N}$ stands for a sampling operator. $\mathbf{\Psi}_{ij} = 1$ if $j = \mathcal{K}_i$; and 0, otherwise. Here, $\mathcal{K}_i$ is the i-th element of $\mathcal{K}$, indicating the i-th of the $K = |\mathcal{K}|$ European countries with COVID-19 data available. $\mathbf{D} \in \mathbb{R}^{N\times N}$ is a diagonal rescaling matrix with $D_{ii} = 1/\sqrt{K\pi_i}$

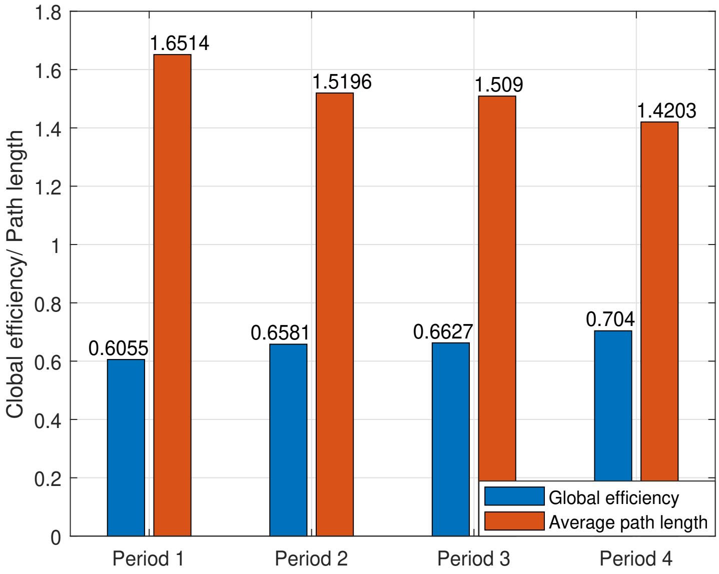


Figure 5.9: Average path length and global efficiency corresponding to different periods of COVID-19.

and π_i being the probability of choosing the i-th $K \times N$-dimensional sample of the K countries on the p-th day of the considered period. Since a uniform sampling process is considered, the sampling metric for every node is $\pi_i = 1/N$.

The RMSE and the R^2 are adopted to quantify the accuracy of the recovered data with respect to the ground-truth COVID-19 data, as given by

$$\text{RMSE} = \sqrt{\sum_{i=1}^{N} (\hat{x}_{pi} - x_{pi})^2 / N}; \tag{5.18}$$

$$R^2 = 1 - \frac{\|\hat{\mathbf{x}}_p - \mathbf{x}_p\|_2^2}{\|\hat{\mathbf{x}}_p - \bar{\mathbf{x}}_p\|_2^2}. \tag{5.19}$$

Here, $\hat{\mathbf{x}}_p$ and $\bar{\mathbf{x}}_p$ are the reconstructed signals and the average of the ground-truths of $\mathbf{x}_p$.

Fig. 5.10 plots the correlations of determination, i.e., R^2, of the proposed algorithm with different regularizer α and data bandwidth K under the pandemic network during Period 1. We see that R^2 reaches its peak at $\alpha = 1$ and $K = 26$; indicating that the optimal regularizer is $\alpha = 1$ for a data bandwidth of $K = 26$. We can similarly determine the optimal values of α for Periods 2 to 4. Fig. 5.11 shows the R^2 of the considered graph learning algorithms in four different periods, where $K = 26$.

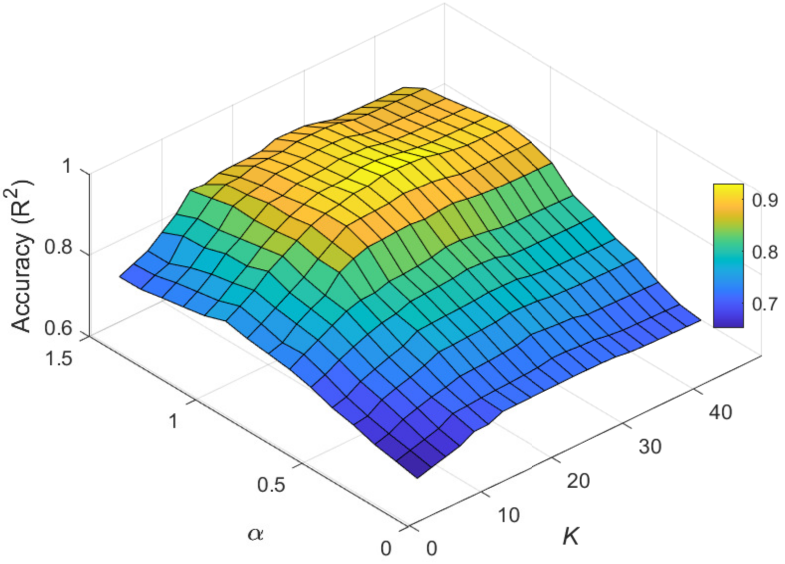


Figure 5.10: The accuracy vs. bandwidth K and α in Period 1.

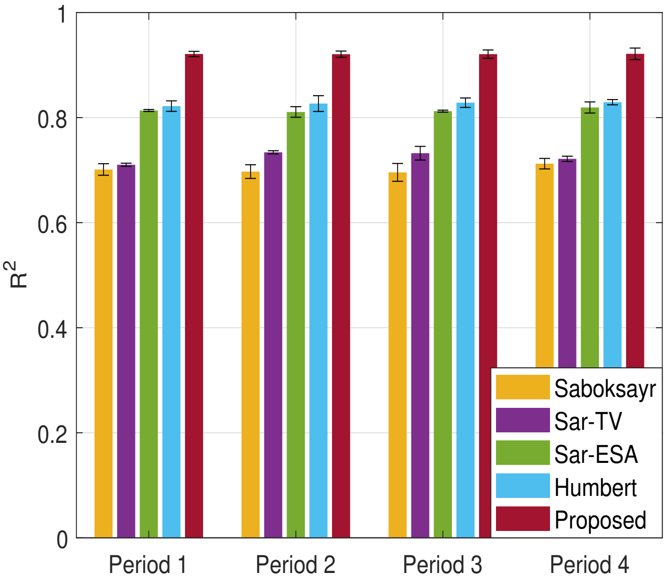


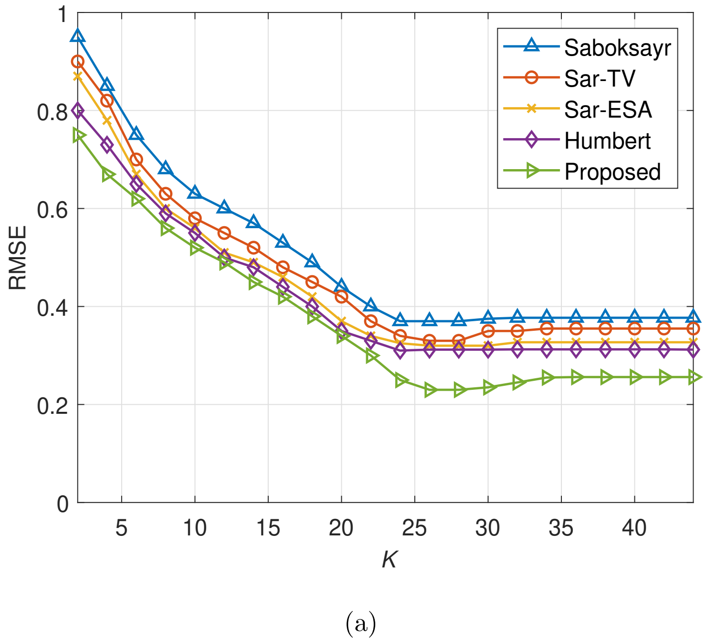
Figure 5.11: Efficiency of reconstruction of different methods upon four periods when $K = 26$.

The proposed algorithm obtains the largest R^2. For example, the improvements of the algorithm are about 29.36%, 27.71%, 12.46%, and 11.11%, compared to Saboksayr's [43], Sar-TV [35], Sar-ESA [35], and Humbert's [36], respectively. To ensure a fair comparison, the parameters are individually tested and optimized for each benchmark in these figures.

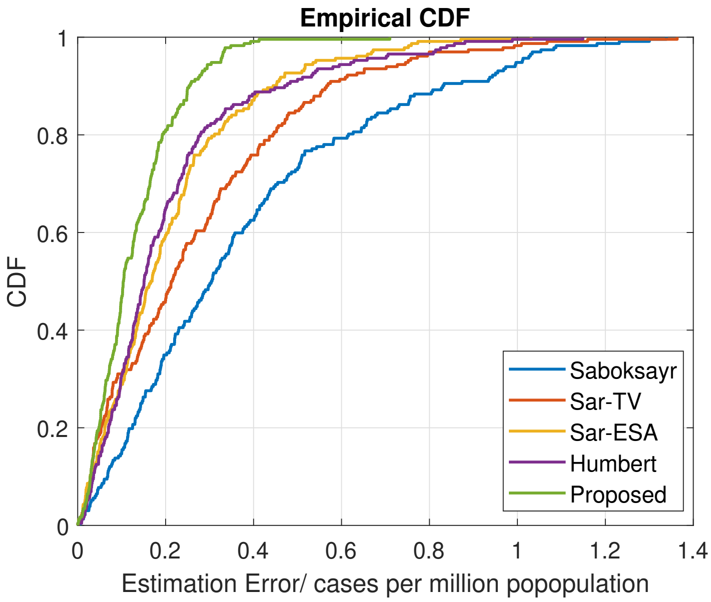
Fig. 5.12(a) shows the RMSE of the considered algorithms with the increase in the signal bandwidth K. We see that under all the considered algorithms, the RMSEs decrease quickly with the growth of K and then converge to constant values. Our proposed algorithm has the smallest RMSE under all values of K. It has the minimum RMSE around 0.23 at $K = 26$ and achieves performance improvements by about 60.87%, 43.48%, 34.78%, and 33.33%, compared to Saboksayr's [43], Sar-TV [35], Sar-ESA [35], and Humbert's [36], respectively. Fig. 5.12(b) plots the cumulative distribution function (CDF) of the errors undergone by the considered algorithms. As depicted, our new algorithm exhibits significantly lower estimation errors compared to the other algorithms. In particular, over 80% of the estimation errors are smaller than 0.2 case per million population under our algorithm. By contrast, 38.3%, 48.4%, 59.5%, and 64.6% of the estimation errors are smaller than 0.2 case per million population under Saboksayr's [43], Sar-TV [35], Sar-ESA [35], and Humbert's [36], respectively.

Next, we proceed to assess the accuracy (R^2) of the considered graph learning algorithms when predicting future missing data based on the graph topologies extracted in the past. In addition to the graph learning techniques, we also consider the state-of-the-art GNN [153]. The COVID-19 dataset of each period is divided into a training set (e.g., the first 80% of the dataset) and a test set (e.g., the remaining 20% of the dataset). In the training phase, the graph learning algorithms extract the graph topology of the training set. In the test phase, the test data of Ukraine is assumed to be missing and is predicted based on the graph topologies extracted from the training set and the available test data of the other countries. By adjusting the ratio between the training and test sets, we show the robustness of the algorithms to the small training set.

As shown in Figs. 5.13(a)–5.13(d), the graph learning methods, including our proposed algorithm, outperform the GNN under different ratios between the training and test sets. When the training set is set to 80% and the testing set is 20%, our algorithm achieves the highest R^2 values with the improvements of about 70.49%, 75.85%, 70.99%, and 68.11% in the four periods, compared to the GNN. Notice that the R^2 value of the GNN can yield negative values, particularly in cases where the



(a)

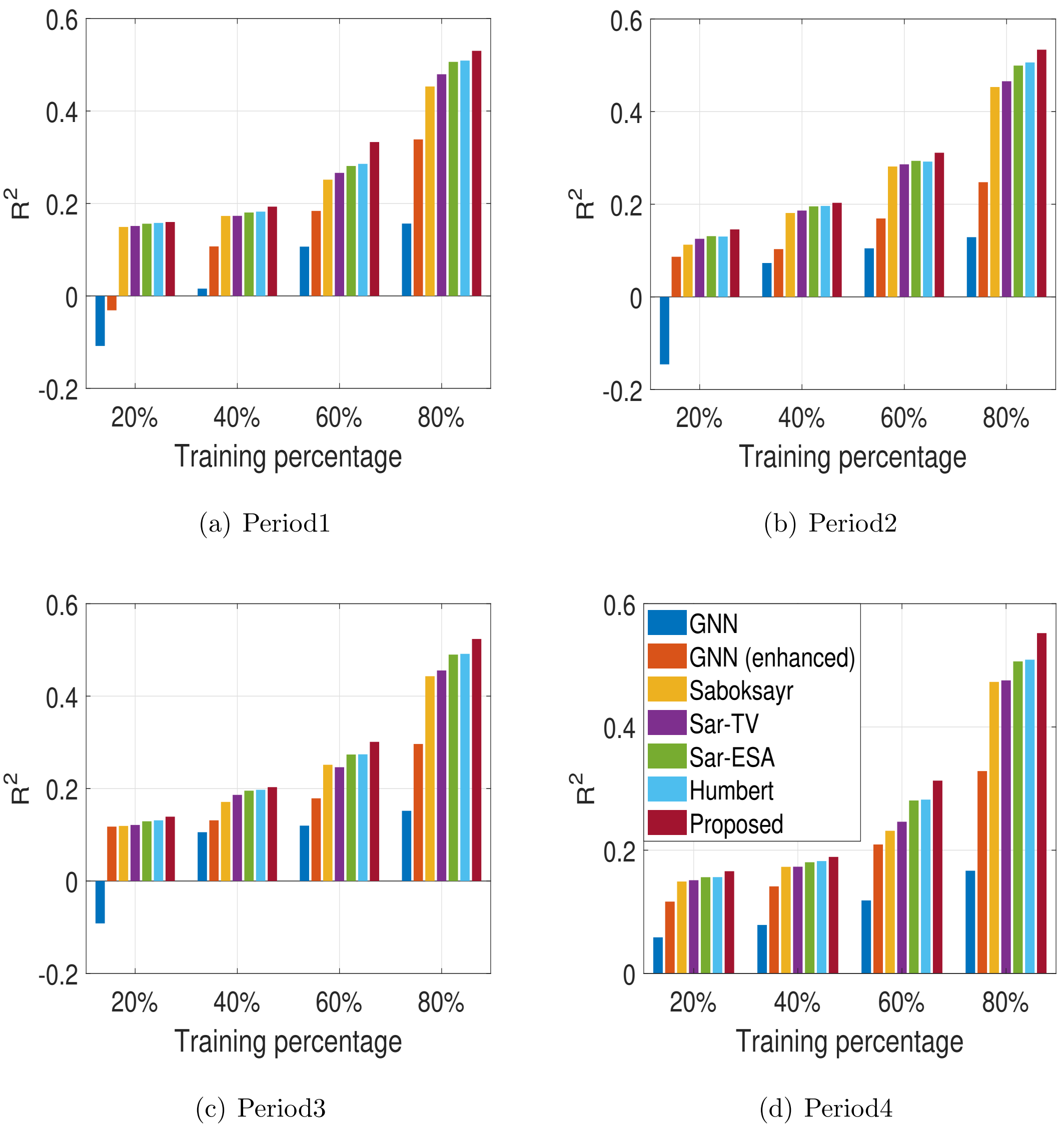


(b)

Figure 5.12: (a) The RMSE vs. the bandwidth K. (b) The CDFs of estimation error under different graph learning methods.

training set is limited. This is the case when even the mean of the data can provide a better fit to the data than the fitted function, e.g., the GNN, when the training set is limited, i.e., 20%.

On the other hand, our new algorithm can enhance the state-of-the-art GNN by providing more accurate graph topologies, compared to a direct calculation of adjacency



(a) Period1

(b) Period2

(c) Period3

(d) Period4

Figure 5.13: Efficiency of reconstruction of different methods of different periods.

matrices (as done in the GNN [153]). By inputting the weighted adjacency matrices of the graphs learned by the algorithm, the GNN can be enhanced and consistently outperform the original GNN in the experiments. Nevertheless, the enhanced GNN still falls short compared to the state-of-the-art graph learning techniques, primarily due to the limited size of the training set, i.e., the COVID-19 data set.

5.5 Conclusion

In this chapter, we proposed a new graph-learning technique to analyze the evolution of the COVID-19 pandemic and reveal the underlying relationship and spreading pattern among different countries. The new technique estimates the graph Laplacian of the COVID-19 data by first deriving the closed-form expression for its eigenvectors and then estimating its eigenvalues with convex optimization. Based on the COVID-19 data, the accuracy of the estimated graph Laplacian was shown to outperform the existing approaches by 33.3% in RMSE and 11.11% in correlation of determination. The new technique helped identify a different set of the most influential and representative European countries, in contrast to the previous techniques. Given the superior accuracy of the algorithm, the set of identified influential countries is expected to be sensible and deserves dedicated research efforts to help understand the COVID-19 spread.

Chapter 6

Preserving the Privacy of Latent Information for Graph-Structured Data

6.1 Introduction

Graph data, characterized by latent graph structures, plays a crucial role in various fields, including physics, [154], [155], biology, transportation [156]–[159], energy [160], engineering [161]–[163], and social science [164], [165]. Illustrative instances of such data encompass brain signals like EEG signals [6], and blood-oxygen-level-dependent time series derived from fMRI on the brain [7]. Moreover, social network data from platforms like Facebook, Twitter, and WeChat offer further examples, revealing social graphs and significant volumes of potentially sensitive and private user information [8], [9].

Privacy is a significant concern for some graph-structured data [166], for example, brain network data obtained by fMRI [7]. The graph-structured brain data can be held by the Department of Neurology in a hospital, and shared with and used by other departments or clinics for big data analytics (e.g., detecting or modeling changes in blood flow that occur with brain activity) or educational purposes.

On the one hand, the latent graph structures of brain network data could expose personal health conditions, such as ADHD [33] and AD [34], under graph interference attacks (based on graph learning techniques [21], [35], [36]). The health conditions could be exploited to reveal the identities of the patients [37], [38]. The latent stimuli of the graph-structured data, which are the input to the latent graphs and derive the output of the graph-structured data observed, are also part of the private information [39]. The bandwidth and waveform shapes of the stimuli can be used as the identifiers of individuals.

On the other hand, minimizing the perturbation on the observed graph-structured data, e.g., brain network data, helps retain the utility of the data, e.g., for measuring and modeling blood flows during brain activities, evaluating the effects of stroke, and examining functioning areas of the brain under different tasks. In this sense, it is critical to protect the privacy of the graph-structured data by obfuscating the latent graph structures and stimuli underlying the graph-structured data, while minimizing changes or perturbations to the graph-structured data to best maintain its utility. However, preserving the privacy of the graph data, more explicitly, the privacy of the latent graph structures and stimulus underlying the graph data, has never been addressed in the literature.

This chapter presents a new approach to preserving the privacy of the latent graph structures and stimuli of graph data while maximizing the utility of the graph data. The approach is important to defend against graph inference attacks, and can have extensive applications to protect personal health records (e.g., brain data), finance transactions, and many other graph-structured data.

- A new multi-objective problem is formulated to preserve the privacy of the latent graph structures and stimuli of graph data and maintain the utility of the data. The problem has never been addressed. The restrictiveness of DP in solving the problem is revealed.

- We derive the analytical expressions for the GFT basis that captures the latent graph structures, and the latent stimuli that are the spectral-domain inputs to the latent graphs.

- We decouple the new multi-objective problem to alternately obfuscate the GFT basis and stimulus against the expressions derived. The difference-of-convex (DC) programming and Stiefel manifold gradient descent are orchestrated to efficiently perturb the GFT basis. The DC programming and gradient descent are employed to perturb the latent stimuli.

Extensive experiments are conducted on synthetic graph data generated under the Random Geometric model, and the real-world ADHD dataset. The new approach is demonstrated to effectively protect the privacy of the latent graph structures and stimuli of graph data, while maintaining the utility of the data. The approach can substantially outperform its DP-based benchmark in the face of graph inference attacks based on the latest graph learning techniques.

6.2 System Model and Problem Statement

In this section, we describe the system model and provide the problem statement.

6.2.1 System Model

Let $\mathbf{Y} \in \mathbb{R}^{N \times M}$ denote the observed data with a latent graph structure $\mathcal{G}(\mathcal{V}, \mathcal{E})$. N is the count of vertices on the latent graph. A $1 \times M$-dimensional time series is associated with each vertex in the observed data $\mathbf{Y}$. $\mathcal{V} = \{1, \cdots, N\}$ is the set of vertices, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. The topological structure of $\mathcal{G}$ can be captured by a weighted and undirected adjacency matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$. The adjacency matrix $\mathbf{W}$ collects all the edges with $W_{ij} = W_{ji} \neq 0$, $\forall (i,j) \in \mathcal{E}$.

Let $\mathbf{D} \triangleq \operatorname{diag}(\mathbf{W}\mathbf{1})$ define the degree matrix containing the node degrees at its diagonal. Also, suppose that each node is connected to at least one other node, ensuring no isolated nodes are in the graph. In other words, none of the diagonal elements is zero in $\mathbf{D}$. Then, according to [84], the combinatorial graph Laplacian of $\mathcal{G}$ is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \tag{6.1}$$

which is a semi-definite matrix with positive elements along its main diagonal and non-positive elements anywhere else [84]. The eigenvectors of $\mathbf{L}$, denoted by

$\mathbf{U}^* \in \mathbb{R}^{N\times N}$, make up the so-called GFT basis [10], which captures the latent graph structure or topology of the graph-structured data $\mathbf{Y}$ [84].

Apart from the latent graph structure, the observed graph-structured data $\mathbf{Y}$ also contains a latent stimulus in the spectral domain. The latent spectral-domain representation (or, in other words, the latent spectral-domain stimulus) of $\mathbf{Y}$ is represented by $\mathbf{S}^* \in \mathbb{R}^{N\times M}$. It projects $\mathbf{Y}$ on the spectral-domain subspace spanned by $\mathbf{U}^*$:

$$\mathbf{Y} = \mathbf{U}^*\mathbf{S}^*, \tag{6.2}$$

where $\mathbf{S}^*$ can exhibit some level of sparsity. Clearly, $\mathbf{S}^*$ relies on both $\mathbf{Y}$ and the latent graph structure characterized by the GFT basis, $\mathbf{U}^*$. Therefore, $\mathbf{S}^*$ and $\mathbf{U}^*$ need to be jointly estimated, which, however, is non-trivial [35].

As reported in [35], [167], and [119], the observed graph-structured data can often be sparse in the canonical domain. Let K denote the bandwidth of $\mathbf{S}^*$ in the spectral domain, $K \in \{1, \cdots, N\}$, and $\mathcal{B}_K$ collect all K-block sparse matrices [142]; i.e.,

$$\mathcal{B}_K \triangleq \{\mathbf{S}^* \in \mathbb{R}^{N\times M}, \mathbf{S}^*(i,:) = \mathbf{0}, \forall i \notin \mathcal{K} \in \mathcal{V}, |\mathcal{K}| = K\}, \tag{6.3}$$

where $\mathbf{S}^*(i,:)$ stands for the i-th row of $\mathbf{S}^*$, while $\mathcal{K} \in \mathcal{V}$ has the cardinality of K.

6.2.2 Problem Statement

In this chapter, our goal is to protect the privacy of the latent information underlying graph-structured data (e.g., the graph structure and the stimulus underlying the observed graph-structured data) while minimizing the perturbations on the observed graph-structured data to maintain the utility of the data. Let $\mathbf{U}^*$ and $\mathbf{S}^*$ denote the latent GFT basis and stimulus, respectively, and $\mathbf{U}$ and $\mathbf{S}$ denote their respective obfuscated versions.

The utility is measured by the difference between the observed graph data and their corresponding perturbed versions, i.e., $\|\mathbf{Y} - \mathbf{U}\mathbf{S}\|_F$. The privacy is measured by the difference between the latent graph structures of the graph data and their corresponding perturbed version, i.e., $\|\mathbf{U} - \mathbf{U}^*\|_F$; and by the difference between the latent stimuli of the graph data and their corresponding perturbed version, i.e., $\|\mathbf{S} - \mathbf{S}^*\|_F$.

The considered problem is formulated as

$$\max_{\mathbf{U}\in\mathbb{R}^{N\times N},\mathbf{S}\in\mathbb{R}^{N\times M}} \|\mathbf{U}-\mathbf{U}^*\|_F^2+\beta\,\|\mathbf{S}-\mathbf{S}^*\|_F^2 \tag{6.4a}$$

$$\text{and} \quad \min_{\mathbf{U}\in\mathbb{R}^{N\times N},\mathbf{S}\in\mathbb{R}^{N\times M}} \|\mathbf{U}\mathbf{S}-\mathbf{Y}\|_F^2\,, \tag{6.4b}$$

$$\text{s.t.} \quad \mathbf{U}^T\mathbf{U}=\mathbf{I}_N, \tag{6.4c}$$

$$\mathbf{S}\in\mathcal{B}_K, \tag{6.4d}$$

$$\mathbf{u}_1=1/\sqrt{N}\;\mathbf{1}. \tag{6.4e}$$

Here, (6.4a) aims to prevent the leakage of the private information about the latent graph structure $\mathbf{U}^*$ and the latent stimulus $\mathbf{S}^*$ by maximizing their difference from their respective obfuscated versions. β is an adjustable hyper-parameter to fine-tune the priority of the two terms in (6.4a). The second objective (6.4b) indicates the perturbed version of $\mathbf{Y}$ needs to be close to the original observation $\mathbf{Y}$ to maintain its utility.

Constraint (6.4c) is due to the orthonormal nature of $\mathbf{U}$. Constraint (6.4d) specify the sparsity of $\mathbf{S}$. Constraint (6.4e) provides the necessary condition of a valid graph Laplacian $\mathbf{L}$; i.e., one eigenvalue of $\mathbf{L}$ must be 0, and the corresponding eigenvector is $\mathbf{u}_1$, which is a column of $\mathbf{U}$ [35].

6.3 Proposed Obfuscation for Graph Data

Fig. 6.1 depicts the diagram of the proposed obfuscation algorithm for graph-structured data, where there are two components: The extraction of the latent information of graph-structured data, and the perturbation of the latent information.

6.3.1 Latent Information Extraction

To preserve the privacy of the latent information $\mathbf{U}^*$ and $\mathbf{S}^*$, we first estimate $\mathbf{U}^*$ and $\mathbf{S}^*$ based on the observed graph-structured data $\mathbf{Y}$. Provided $K\in[1,N]$, we cast the problem as

$$\begin{aligned}(\mathbf{U}^*,\mathbf{S}^*)&=\arg\min_{\mathbf{U},\mathbf{S}}\|\mathbf{Y}-\mathbf{U}\mathbf{S}\|_F^2\\ \text{s.t.}&\quad(6.4c),\ (6.4d),\ (6.4e),\end{aligned} \tag{6.5}$$

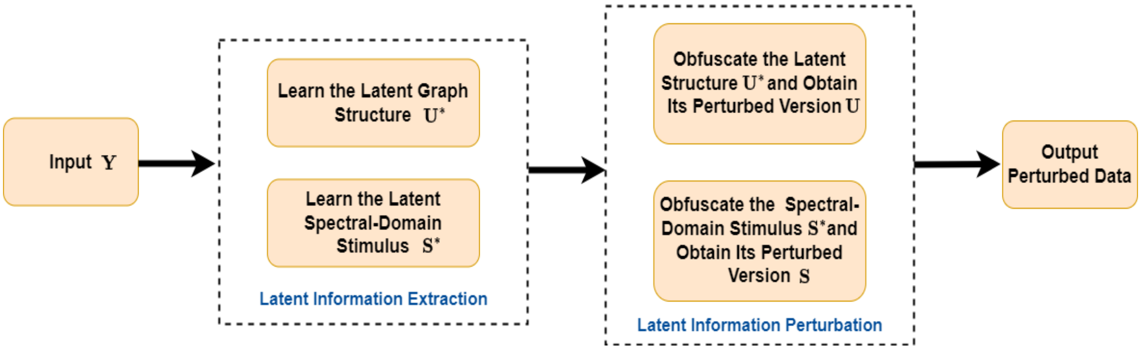

Input $\mathbf{Y}$
Learn the Latent Graph Structure $\mathbf{U}^*$
Learn the Latent Spectral-Domain Stimulus $\mathbf{S}^*$
Latent Information Extraction
Obfuscate the Latent Structure $\mathbf{U}^*$ and Obtain Its Perturbed Version $\mathbf{U}$
Obfuscate the Spectral-Domain Stimulus $\mathbf{S}^*$ and Obtain Its Perturbed Version $\mathbf{S}$
Latent Information Perturbation
Output Perturbed Data

Figure 6.1: The flow diagram of the proposed obfuscation algorithm to perturb the latent information of graph-structured data.

where, with a little abuse of notations, we define $\mathbf{U}$ and $\mathbf{S}$ to be the estimation variables. The closed-form solution to problem (6.5) can be given in the following proposition.

Proposition 1 *Given $K \in [1, N]$, the solution to problem (6.5), $\mathbf{U}^*$ and $\mathbf{S}^*$, is uniquely given in a closed form by*

$$\mathbf{U}^* = \mathrm{Eig}\left[\left(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^T\right)^T\right]; \tag{6.6}$$

$$\mathbf{S}^* = (\mathbf{U}^*)^T\mathbf{Y}, \tag{6.7}$$

where $\mathrm{Eig}[\mathbf{Z}]$ *gives the eigenvectors of* $\mathbf{Z}$.

Remark 6 *While the closed-form solution to problem* (6.5)*, i.e.,* $\mathbf{U}^*$ *and* $\mathbf{S}^*$*, requires the knowledge of K, the solution given in* (6.6) *and* (6.7) *is suitable for any* $K \in [1, N]$*. For this reason, we first configure $K = N$ to access the entire spectrum of the stimulus of* $\mathbf{Y}$*. Subsequently, the norms of all rows in* $\mathbf{S}^*$*, i.e.,* $||\mathbf{S}^*(i,:)||$*, are evaluated. K is typically determined by counting rows yielding larger norms than a predefined threshold.*

6.3.2 Perturbation for Graph Privacy Preservation

Recall that $\mathbf{U}$ and $\mathbf{S}$ are the perturbed versions of $\mathbf{U}^*$ and $\mathbf{S}^*$ in (6.4), respectively. As stated in (6.4), we maximize $||\mathbf{U}-\mathbf{U}^*||_F$ and $||\mathbf{S}-\mathbf{S}^*||_F$, i.e., the perturbations of the latent graph $\mathbf{U}^*$ and stimulus $\mathbf{S}^*$, while minimizing $||\mathbf{US}-\mathbf{Y}||_F$, i.e., the utility loss of the observed graph-structured data $\mathbf{Y}$.

Given $\mathbf{Y}$, $\mathbf{U}^*$, and $\mathbf{S}^*$, problem (6.4) can be recast to improve its mathematical tractability, as given by

$$\begin{aligned} \max_{\mathbf{S}\in\mathbb{R}^{N\times M},\mathbf{U}\in\mathbb{R}^{N\times N}} & \frac{\|\mathbf{U}-\mathbf{U}^*\|_F^2+\beta\|\mathbf{S}-\mathbf{S}^*\|_F^2}{\|\mathbf{U}\mathbf{S}-\mathbf{Y}\|_F^2} \\ \text{s.t.} \quad & (6.4c), \ (6.4e). \end{aligned} \tag{6.8}$$

Despite the conciseness of problem (6.8), it is challenging to obtain the optimal solution due to the non-convexity of objective function and the non-convexity and restrictive equality constraint in (6.4c). We propose to address problem (6.8) using fractional programming [168]. By introducing an auxiliary variable α, problem (6.8) can be reformulated as

$$\begin{aligned} \max_{\mathbf{U},\mathbf{S}} \ & \|\mathbf{U}-\mathbf{U}^*\|_F^2+\beta\|\mathbf{S}-\mathbf{S}^*\|_F^2-\alpha\|\mathbf{U}\mathbf{S}-\mathbf{Y}\|_F^2 \\ \text{s.t.} \quad & (6.4c), \ (6.4e). \end{aligned} \tag{6.9}$$

In essence, the objective of problem (6.8) is to maximize α. Next, we solve problem (6.9) by optimizing α, $\mathbf{U}$, and $\mathbf{S}$ in an alternating manner, as delineated in the following.

Obfuscate Latent Graph Structure U

Given fixed α and $\mathbf{S}$ in the $(i-1)$-th AO iteration, i.e., $\alpha^{(i-1)}$ and $\mathbf{S}^{(i-1)}$ $(i=1,2,\cdots)$, in the i-th iteration, problem (6.9) is reduced to

$$\begin{aligned} \min_{\mathbf{U}} \ & \alpha^{(i-1)}\left\|\mathbf{U}\mathbf{S}^{(i-1)}-\mathbf{Y}\right\|_F^2-\|\mathbf{U}-\mathbf{U}^*\|_F^2 \\ \text{s.t.} \quad & (6.4c), \ (6.4e). \end{aligned} \tag{6.10}$$

Consider the constraint (6.4e), we rewrite $\mathbf{U}=[\mathbf{u}_1,\mathbf{V}]$, where $\mathbf{V}\in\mathbb{R}^{N\times(N-1)}$ collects the rest of the columns-vectors of $\mathbf{U}$ except the first column $\mathbf{u}_1$. Likewise, $\mathbf{S}=[\mathbf{s}_1,\mathbf{H}]^T$, where $\mathbf{s}_1$ is the first row-vector of $\mathbf{S}$ and $\mathbf{H}\in\mathbb{R}^{(N-1)\times M}$ collects the rest of the row-vectors of $\mathbf{S}$. Then, problem (6.10) can be rewritten as

$$\min_{\mathbf{V}} \ \alpha^{(i-1)}\left\|\mathbf{V}\mathbf{H}^{(i-1)}-\mathbf{Y}\right\|_F^2-\|\mathbf{V}-\mathbf{V}^*\|_F^2 \tag{6.11a}$$

$$\text{s.t.} \quad \mathbf{V}^T\mathbf{V}=\mathbf{I}_{N-1}, \tag{6.11b}$$

$$\mathbf{u}_1^T\mathbf{V}=\mathbf{0}, \tag{6.11c}$$

where $\mathbf{U}^* = [\mathbf{u}_1, \mathbf{V}^*]$ with $\mathbf{V}^* \in \mathbb{R}^{N\times(N-1)}$ collecting the rest of the column-vectors of $\mathbf{U}^*$ apart from $\mathbf{u}_1$.

The objective function in (6.11) has the form of DC program:

$$\min_{\mathbf{V}} \; g(\mathbf{V}) - h(\mathbf{V}), \tag{6.12}$$

where $g(\mathbf{V}) = \alpha^{(i-1)} \left\|\mathbf{V}\mathbf{H}^{(i-1)} - \mathbf{Y}\right\|_F^2$ and $h(\mathbf{V}) = \|\mathbf{V} - \mathbf{V}^*\|_F^2$, both of which are convex functions.

By using the difference of convex algorithm (DCA) [169], problem (6.11) can be reformulated to a strongly convex optimization problem. At the l-th iteration of the DCA ($l = 1, 2, \cdots$), we have

$$\begin{aligned} \min_{\mathbf{V}} \; & \alpha^{(i-1)} \left\|\mathbf{V}\mathbf{H}^{(i-1)} - \mathbf{Y}\right\|_F^2 - \langle \mathbf{V}, \nabla h(\mathbf{V}^{l-1})\rangle && \text{(6.13a)} \\ \text{s.t.} \quad & \mathbf{V}^T\mathbf{V} = \mathbf{I}_{N-1}, && \text{(6.13b)} \\ & \mathbf{u}_1^T\mathbf{V} = \mathbf{0}, && \text{(6.13c)} \end{aligned}$$

where $\nabla h(\mathbf{V}^{l-1})$ is a gradient of $h(\mathbf{V})$ at $\mathbf{V}^{l-1}$, the local point obtained in the $(l-1)$-th iteration of the DCA.

The objective function (6.13a) is convex. Consider the orthonormal vector variables in constraint (6.13b); or in other words, the feasible solution region is on the Stiefel manifold. Problem (6.13) is convex on the Stiefel manifold and can be uniquely addressed utilizing the Stiefel manifold gradient descent [122] with the details below.

The Lagrangian of (6.13) is denoted as

$$\begin{aligned} \mathcal{L}(\mathbf{V}, \boldsymbol{\Psi}, \boldsymbol{\Phi}) = & \; \alpha^{(i-1)} \left\|\mathbf{V}\mathbf{H}^{(i-1)} - \mathbf{Y}\right\|_F^2 - \langle \mathbf{V}, \nabla h(\mathbf{V}^{l-1})\rangle \\ & -\frac{1}{2}\mathrm{tr}\Big(\boldsymbol{\Psi}^T(\mathbf{V}^T\mathbf{V} - \mathbf{I}_{N-1})\Big) - \frac{1}{2}\mathrm{tr}\Big(\boldsymbol{\Phi}^T(\mathbf{u}_1^T\mathbf{V})\Big), \end{aligned} \tag{6.14}$$

where $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ are the Lagrangian multipliers corresponding to (6.13b) and (6.13c), respectively.

Applying the Karush-Kuhn-Tucker (KKT) conditions, we differentiate the Lagrangian function regarding $\mathbf{V}$ and equate it to zero:

$$\nabla_{\mathbf{V}}\mathcal{L}(\mathbf{V}, \boldsymbol{\Psi}, \boldsymbol{\Phi}) = \nabla\mathcal{F}(\mathbf{V}) - \mathbf{V}\boldsymbol{\Psi} - \frac{1}{2}\mathbf{u}_1\boldsymbol{\Phi} = \mathbf{0}, \tag{6.15}$$

where $\mathcal{F}(\mathbf{V}) = \alpha^{(i-1)} \left\| \mathbf{V}\mathbf{H}^{(i-1)} - \mathbf{Y} \right\|_F^2 - \langle \mathbf{U}, \nabla h(\mathbf{V}^{l-1}) \rangle$ is defined for the brevity of notation, and $\nabla \mathcal{F}(\mathbf{V})$ is the gradient of $\mathcal{F}(\mathbf{V})$.

By left multiplying $\mathbf{V}^T$ on both sides of (6.15), we have

$$\mathbf{V}^T \nabla \mathcal{F}(\mathbf{V}) - \mathbf{V}^T \mathbf{V} \boldsymbol{\Psi} = \mathbf{0}. \tag{6.16}$$

We substitute (6.13b) into (6.16) and then reorganize (6.16), yielding

$$\boldsymbol{\Psi} = \mathbf{V}^T \nabla \mathcal{F}(\mathbf{V}). \tag{6.17}$$

By left multiplying $\mathbf{u}_1^T$ on both sides of (6.15), we have

$$\mathbf{u}_1^T \nabla \mathcal{F}(\mathbf{V}) - \frac{1}{2} \mathbf{u}_1^T \mathbf{u}_1 \boldsymbol{\Phi} = \mathbf{0}. \tag{6.18}$$

We substitute (6.13c) into (6.18) and then reorganize (6.18), yielding

$$\boldsymbol{\Phi} = 2\mathbf{u}_1^T \nabla \mathcal{F}(\mathbf{V}). \tag{6.19}$$

Since the constraint $\mathbf{V}^T\mathbf{V}$ is symmetric, $\boldsymbol{\Psi}$ is asymmetric and therefore $\boldsymbol{\Psi} = \nabla \mathcal{F}(\mathbf{V})^T \mathbf{V}$. By substituting (6.17) and (6.19) into (6.15), the gradient in (6.15) can be reformulated to

$$\nabla_{\mathbf{V}} \mathcal{L}(\mathbf{V}, \boldsymbol{\Psi}, \boldsymbol{\Phi}) = \nabla_{\mathbf{V}} \mathcal{L}(\mathbf{V}) \tag{6.20a}$$

$$= \nabla \mathcal{F}(\mathbf{V}) - \mathbf{V} \nabla \mathcal{F}(\mathbf{V})^T \mathbf{V} - \mathbf{u}_1 \mathbf{u}_1^T \nabla \mathcal{F}(\mathbf{V}). \tag{6.20b}$$

By running the Stiefel manifold gradient descent, problem (6.13) can be solved by iteratively updating the gradient of the Lagrange function $\nabla_{\mathbf{V}} \mathcal{L}(\mathbf{V})$ with (6.20) and $\mathbf{V}$ with the right-hand scaled gradient projection method [122]:

$$\mathbf{V} \leftarrow \pi(\mathbf{V}' - \tau_{\mathbf{V}} \nabla_{\mathbf{V}} \mathcal{L}(\mathbf{V}) \mathcal{A}(\mu, \tau)), \tag{6.21}$$

where $\pi(\cdot)$ is the projection operator, i.e., $\pi(\mathbf{X}) = \mathbf{Q}\mathbf{I}\mathbf{P}^T$ if $\mathbf{X} = \mathbf{Q}\boldsymbol{\Sigma}\mathbf{P}^T$ by singular value decomposition (SVD) [122]; $\mathcal{A}(\mu, \tau_{\mathbf{V}})$ is a scaling matrix with $\mu \in (0, 1)$, i.e.,

$$\mathcal{A}(\mu, \tau_{\mathbf{V}})) = (\mathbf{I}_{N-1} + \mu \tau_{\mathbf{V}} \mathbf{V}^T \nabla_{\mathbf{V}} \mathcal{L}(\mathbf{V}))^{-1}, \tag{6.22}$$

and $\tau_{\mathbf{V}}$ is the step size and is given by

$$\tau_{\mathbf{V}} = \begin{cases} \frac{\|\mathbf{V}-\mathbf{V}'\|_F^2}{\langle \mathbf{V}-\mathbf{V}', \nabla_{\mathbf{V}}\mathcal{L}(\mathbf{V})-\nabla_{\mathbf{V}}\mathcal{L}(\mathbf{V}')\rangle}, & \text{in odd-numbered iterations,} \\ \frac{\langle \mathbf{V}-\mathbf{V}', \nabla_{\mathbf{U}}\mathcal{L}(\mathbf{V})-\nabla_{\mathbf{V}}\mathcal{L}(\mathbf{V}')\rangle}{\|\nabla_{\mathbf{V}}\mathcal{L}(\mathbf{V})-\nabla_{\mathbf{V}}\mathcal{L}(\mathbf{V}')\|_F^2}, & \text{in even-numbered iterations.} \end{cases} \tag{6.23}$$

Here, $\mathbf{V}'$ is the counterpart of $\mathbf{V}$ obtained at the previous Stiefel manifold gradient descent iteration, $\tau_{\mathbf{V}} \in [\tau_{\min}, \tau_{\max}]$ with $\tau_{\min}$ and $\tau_{\max}$ being the minimum and maximum step sizes, respectively.

Given the convexity of (6.13), the Stiefel manifold gradient descent can certainly converge. $\mathbf{U}^l = [\mathbf{u}_1, \mathbf{V}^l]$, the solution of $\mathbf{U}$ at the l-th iteration of the DCA, can be obtained. $\mathbf{U}^l$ is taken as the local point in the following $(l+1)$-th iteration of DCA. This process repeats until the convergence of the DCA, and $\mathbf{U}^{(i)}$ is obtained.

Obfuscate Latent Spectral-Domain Stimulus S

Given $\mathbf{U}^{(i)}$, problem (6.9) is reformulated to an unconstrained DC program:

$$\min_{\mathbf{S}} \; g(\mathbf{S}) - h(\mathbf{S}), \tag{6.24}$$

where $g(\mathbf{S}) = \alpha^{(i-1)} \left\|\mathbf{U}^{(i)}\mathbf{S} - \mathbf{Y}\right\|_F^2$ and $h(\mathbf{S}) = \beta \|\mathbf{S} - \mathbf{S}^*\|_F^2$, both of which are convex functions. As a result, (6.24) can be solved by the DCA. More specifically, at the l-th iteration of the DCA ($l = 1, 2, \cdots$), the optimization problem below is addressed:

$$\min_{\mathbf{S}} \; \mathcal{F}(\mathbf{S}) = g(\mathbf{S}) - \langle \mathbf{S}, \nabla h(\mathbf{S}^{l-1})\rangle, \tag{6.25}$$

where $\nabla h(\mathbf{S}^{l-1})$ is a gradient of $h(\mathbf{S})$ at $\mathbf{S}^{l-1}$, the local point obtained in the $(l-1)$-th iteration of the DCA. Given its convexity, problem (6.25) is solved using the gradient descent method. Specifically, $\mathbf{S}$ is updated by

$$\mathbf{S} = \mathbf{S}' - \tau_{\mathbf{S}} \nabla \mathcal{F}(\mathbf{S}'), \tag{6.26}$$

where $\mathbf{S}'$ represent the matrix of $\mathbf{S}$ at the previous iteration of the gradient descent, $\nabla \mathcal{F}(\mathbf{S})$ is the gradient of $\mathcal{F}(\mathbf{S})$, and $\tau_{\mathbf{S}}$ is the step size. Upon the convergence of the gradient descent method, $\mathbf{S}^l$ is obtained and used as the local point of the following $(l+1)$-th iteration of the DCA. Upon the convergence of the DCA, $\mathbf{S}^{(i)}$ is obtained.

Update α

Given $\mathbf{U}^{(i)}$ and $\mathbf{S}^{(i)}$, the value of α can be updated for the next AO iteration, i.e., $\alpha^{(i+1)}$, as given by

$$\alpha^{(i+1)} = \frac{\left\|\mathbf{U}^{(i)} - \mathbf{U}^*\right\|_F^2 + \beta \left\|\mathbf{S}^{(i)} - \mathbf{S}^*\right\|_F^2}{\|\mathbf{U}^{(i)}\mathbf{S}^{(i)} - \mathbf{Y}\|_F^2}. \tag{6.27}$$

6.3.3 Complexity Analysis

Algorithm 2 summarizes the proposed AO-based algorithm that solves problem (6.9), the above three parts are conducted in an alternating manner until the convergence of α, i.e., $\|\alpha^{(i+1)} - \alpha^{(i)}\| \leq \epsilon_{\text{AO}}$ with $\epsilon_{\text{AO}} \to 0$ being a preconfigured convergence accuracy. Each AO iteration starts by running the DCA and the Stiefel manifold gradient descent for $\mathbf{U}$ till convergence, followed by the DCA and the gradient descent for $\mathbf{S}$. Their respective convergence criteria are

$$\|\nabla_{\mathbf{V}}\mathcal{L}(\mathbf{V})\|_F \leq \epsilon_{\text{SM}},\ \|\nabla\mathcal{F}(\mathbf{S})\|_F \leq \epsilon_{\text{GD}}, \tag{6.28}$$

$$\|\mathbf{U}^{l+1} - \mathbf{U}^l\| \leq \epsilon_{\text{DC}},\ \text{and}\ \ \|\mathbf{S}^{l+1} - \mathbf{S}^l\| \leq \epsilon_{\text{DC}}, \tag{6.29}$$

where ϵ_{SM}, ϵ_{GD}, and ϵ_{DC} are the preconfigured accuracies for the Stiefel manifold gradient descent, gradient descent, and DCA, respectively.

In each iteration of Algorithm 2, the cost of evaluating $\mathbf{U}$ is primarily influenced by the SVD in (6.21) per Stiefel manifold gradient descent iteration, incurring a cost of $\mathcal{O}(N^3)$. The complexity of evaluating $\mathbf{S}$ is $\mathcal{O}(MN)$ per gradient descent iteration, dominated by calculating the gradient of $\mathcal{F}(\mathbf{S})$. Therefore, the overall cost of Algorithm 2 is $\mathcal{O}\Big(\big(MN\log(\frac{1}{\epsilon_{\text{GD}}}) + N^3\log(\frac{1}{\epsilon_{\text{SM}}})\big)\log(\frac{1}{\epsilon_{\text{DC}}})\log(\frac{1}{\epsilon_{\text{AO}}})\Big)$, where $\log(\frac{1}{\epsilon_{\text{SM}}})$, $\log(\frac{1}{\epsilon_{\text{GD}}})$, $\log(\frac{1}{\epsilon_{\text{DC}}})$, and $\log(\frac{1}{\epsilon_{\text{AO}}})$ give the numbers of iterations for the Stiefel manifold gradient, gradient descent, DCA, and AO to converge, respectively.

6.4 Numerical Evaluation

Extensive simulations and experiments are carried out to gauge the proposed obfuscation technique for graph data, i.e., Algorithm 1, using a desktop computer with an i7-8650U CPU and 16G RAM. By default, we set $\epsilon_{\text{AO}} = \epsilon_{\text{SM}} = \epsilon_{\text{GD}} = \epsilon_{\text{DC}} = 10^{-5}$.

Algorithm 2: The proposed AO-based obfuscation method for graph-structured data

```
Initialization: Set $\epsilon_{\mathrm{AO}}$, $\epsilon_{\mathrm{SM}}$, $\epsilon_{\mathrm{GD}}$, $\epsilon_{\mathrm{DC}}$, $\mu$, and $l = 0$; randomly initialize $\mathbf{U}^0$ and $\mathbf{S}^0$; calculate $\alpha^0$ by (6.27);
while $\alpha$ is yet to converge with accuracy of $\epsilon_{\mathrm{AO}}$ do
    for $l = 0, 1, 2, \cdots$ do
        while $\|\nabla_{\mathbf{V}}\mathcal{L}(\mathbf{V})\|_F \geq \epsilon_{\mathrm{SM}}$ do
            Update $\tau_{\mathbf{V}}$ by (6.23);
            Compute $\tau = \max(\min(\tau_{\mathbf{V}}, \tau_{\max}), \tau_{\min})$;
            Update $\mathbf{V}$ by (6.21);
            $\mathbf{V}' \leftarrow \mathbf{V}$;
        end
        if $\mathbf{U}^l = \mathbf{U}^{l-1}$ then
            Return to $\mathbf{U}^l$;
        else
            $\mathbf{U}^{l-1} \leftarrow \mathbf{U}^l$, $l = l + 1$;
        end
    end
    for $l = 0, 1, 2, \cdots$ do
        while $\|\nabla\mathcal{F}(\mathbf{S})\|_F \geq \epsilon_{\mathrm{GD}}$ do
            Update $\mathbf{S}$ by (6.26);
            $\mathbf{S}' \leftarrow \mathbf{S}$;
        end
        if $\mathbf{S}^l = \mathbf{S}^{l-1}$ then
            Return to $\mathbf{S}^l$;
        else
            $\mathbf{S}^{l-1} \leftarrow \mathbf{S}^l$, $l = l + 1$;
        end
    end
    Update $\alpha$ by (6.27);
end
```

Graph Inference Attack: We assess the resistance of the new graph data obfuscation algorithm to graph interference attacks. Following is a list of the latest graph inference techniques that can be used to launch graph interference attacks.

- Dong's algorithm[18]: This is an alternating minimization algorithm that fixes one variable and solves the others in an alternating manner to infer the optimal Laplacian matrix under the assumption of smooth data.
- Kalofolias' algorithm [19]: This primal-dual algorithm was designed to learn the weighted adjacency matrix of graphs.
- Sardellitti's TV graph learning algorithm [35]; This is a two-step strategy consisting of (a) deriving the GFT basis out of data via AO, and (b) then learning the graph Laplacian matrix using the sparsifying transform via convex optimization technique, named "TV-based graph learning".
- Sardellitti's ESA graph learning algorithm [35]: Different from the TV-based graph learning algorithm, this algorithm introduces a two-step strategy by deriving the graph Laplacian using sparsifying transform, as well as the GFT basis utilizing convex optimization, named "ESA graph learning" in the second step.
- Humbert's algorithm [36]: This method runs Riemannian manifold gradient descent and linear cone programs in an alternating fashion. It learns graphs from multivariate data with smoothness and band-limitedness.

Performance Metric: F-measure is used to measure the graph learning results and is denoted as

$$\text{F-measure} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{6.30}$$

where Precision $= \mathcal{E}_g \cap \mathcal{E}_r / \mathcal{E}_r$ and Recall $= \mathcal{E}_g \cap \mathcal{E}_r / \mathcal{E}_g$. Here, $\mathcal{E}_g$ and $\mathcal{E}_r$ collect ground-truth and detected graphs, respectively.

The correlation coefficient $\rho_{\mathbf{W}}(\mathbf{W}_0, \mathbf{W})$ (or $\rho_{\mathbf{W}}$) between a graph detected by the graph leaning attacks and its corresponding ground-truth is expressed as [35]

$$\rho_{\mathbf{W}}\left(\mathbf{W}_0, \mathbf{W}\right) = \frac{\sum_{ij} W_{0ij} W_{ij}}{\sqrt{\sum_{ij} W_{0ij}^2}\sqrt{\sum_{ij} W_{ij}^2}}, \tag{6.31}$$

where $\mathbf{W}$ is the weighted adjacency matrix of a detected graph with the (i, j)-th element W_{ij}, and $\mathbf{W}_0$ is the weighted adjacency matrix of the corresponding ground-truth graph with the (i, j)-th element W_{0ij}.

DP-based Benchmark: We consider the DP-based Gaussian mechanism as the benchmark for the proposed Algorithm 2. In the DP-based mechanism, the DP noise yields the Gaussian distribution with the variance of the noise given by [170]

$$\sigma_{\text{DP}}^2 = \frac{2\ln\left(1.25/\delta\right)\Delta^2}{\epsilon^2}, \tag{6.32}$$

where δ is the probability of information accidentally being leaked, ϵ is the privacy budget, and Δ is the global sensitivity.

As discussed in Section 6.1, no existing technique has been designed to protect the latent information of graph-structured data in the existing literature. While DP is a general approach to obfuscating data with a balance between privacy and utility, it is unsuitable for graph-structured data, e.g., brain network data. Specifically, adding DP noises to the GFT basis, $\mathbf{U}^*$, is unacceptable and would breach the constraints, e.g., the orthonormality of the GFT basis.

We apply DP to the observed graph-structured data $\mathbf{Y}$. Here, $\Delta = \frac{\|\mathbf{U}-\mathbf{U}^*\|_F^2+\beta\|\mathbf{S}-\mathbf{S}^*\|_F^2}{\|\mathbf{US}-\mathbf{Y}\|_F^2}$. However, the direct use of DP to perturb the graph-structured data can be ineffective in protecting the latent information, because the GFT basis characterizing the latent graph structure is stringently constrained and less susceptible to the added noises, as discussed below.

Remark 7 *The latent graph structures of graph data are resistant to the Gaussian DP noise. This is due to the fact that the GFT basis $\mathbf{U}^*$ is solely determained by the eigenvectors of $\mathbf{Y}\mathbf{Y}^T$, while $\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T - M\sigma_{\text{DP}}^2\mathbf{I}$ is an asymptotic unbiased estimator of $\mathbf{Y}\mathbf{Y}^T$ and $\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T$ has the same eigenvectors as $\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T - M\sigma_{\text{DP}}^2\mathbf{I}$. Here, $\hat{\mathbf{Y}} = \mathbf{Y} + \mathbf{n}$ is the perturbed version of $\mathbf{Y}$ with the DP noise added. $\mathbf{n} \sim \mathcal{N}(0, \sigma_{\text{DP}}^2\mathbf{I})$ is the Gaussian DP noise. Specifically, the expectation of $\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T$ over the Gaussian DP noise is given* *by*

$$\mathbb{E}(\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T) = \mathbf{Y}\mathbf{Y}^T + M\sigma_{\text{DP}}^2\mathbf{I}. \tag{6.33}$$

Hence, the eigenvectors of $\mathbb{E}(\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T)$ *and* $\mathbf{Y}\mathbf{Y}^T$ *are identical.*

Without the DP noise, $\mathbf{U}^*$ *in* (6.6) *is solely determined by the eigenvectors of* $\mathbf{Y}\mathbf{Y}^T$*. In the presence of the DP noise,* $\mathbf{U}^*$ *is given by*

$$\mathbf{U}^* = \text{Eig}\left[\left(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^T\right)\mathbb{E}(\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T)\left(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^T\right)^T\right]. \tag{6.34}$$

Since $\mathbb{E}(\frac{1}{M}\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T - \sigma_{\text{DP}}^2\mathbf{I}) = \frac{1}{M}\mathbf{Y}\mathbf{Y}^T$ *(i.e.,* $\frac{1}{M}\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T - \sigma_{\text{DP}}^2\mathbf{I}$ *is an asymptotic unbiased estimate of* $\frac{1}{M}\mathbf{Y}\mathbf{Y}^T$ *as* $M \to \infty$*) based on* (6.33) *and the eigenvectors of* $\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T - M\sigma_{\text{DP}}^2\mathbf{I}$ *and* $\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T$ *are identical,* $\mathbf{U}^*$ *is asymptotically approximately given by*

$$\mathbf{U}^* \approx \text{Eig}\left[\left(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^T\right)\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T\left(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^T\right)^T\right]. \tag{6.35}$$

In this sense, perturbing the graph-structured data $\mathbf{Y}$ *is less effective in perturbing the latent graph structures of* $\mathbf{Y}$*.*

6.4.1 Performance Evaluation with Synthetic Data

We first generate random graphs using the widely adopted Random Geometric model [124], and generate random graphs with six connections per node.

- *Ground truth:* With a graph created using the Random Geometric graph model, we derive its ground-truth Laplacian, denoted by $\mathbf{L}_0$. We also obtain the ground-truth GFT basis, denoted by $\mathbf{U}_0$, by taking the SVD of $\mathbf{L}_0$.

- *Synthetic data:* We produce the observed graph data $\mathbf{Y} = \mathbf{U}_0\mathbf{S}_0$ with $\mathbf{S}_0 = [\mathbf{s}_{0,1}, \cdots, \mathbf{s}_{0,M}] \in \mathbb{R}^{N\times M}$ randomly generated yielding $\mathbf{s}_{0,m} \sim \mathcal{N}(0, \mathbf{\Lambda}_K^\dagger)$, where $\text{diag}(\mathbf{\Lambda}_K) = (\lambda_1, \cdots, \lambda_K, 0, \cdots, 0)$. We define the precision matrix of $\mathbf{s}_{0,m}$ as the eigenvalue matrix of $\mathbf{L}$ with the most significant $(N-K)$ eigenvalues replaced by 0, as done in [119].

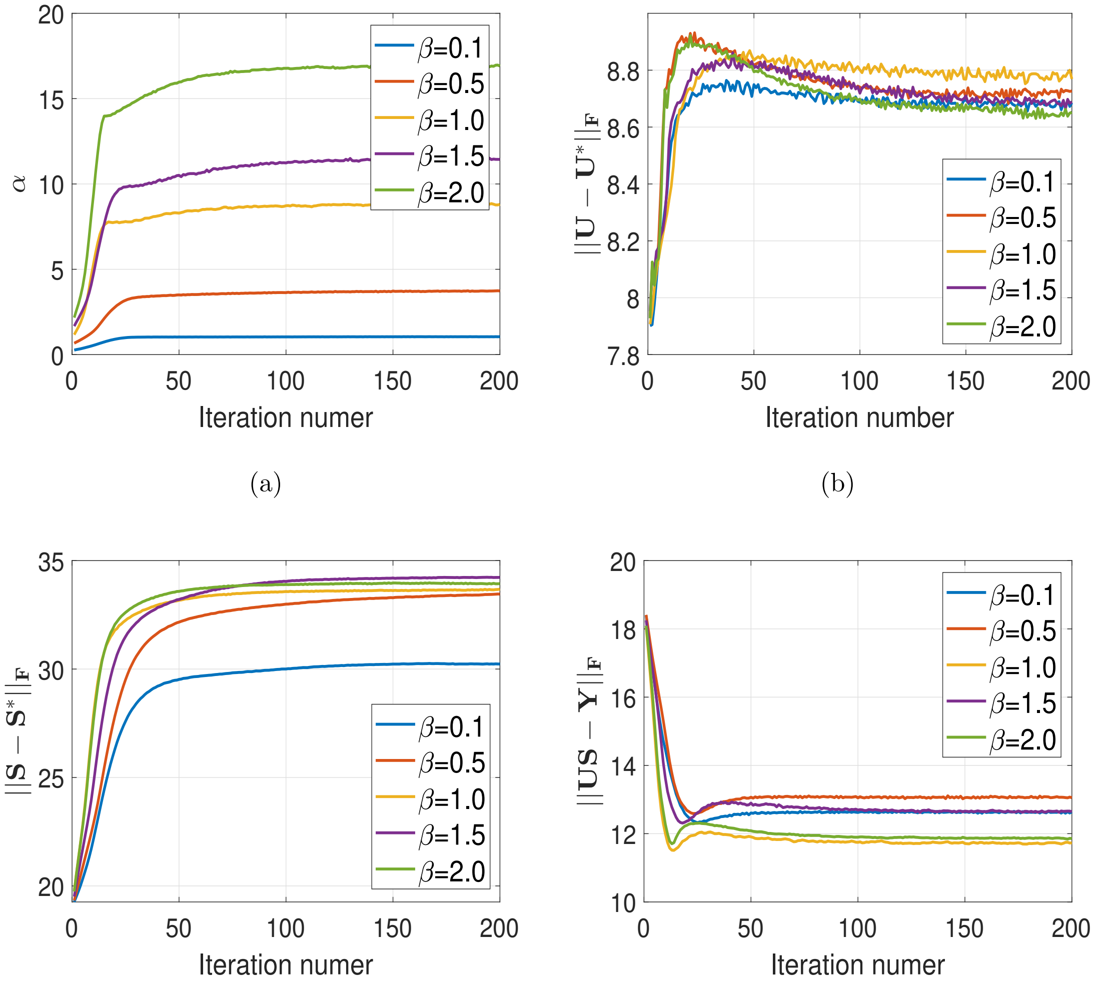
- Obfuscated data (proposed Algorithm 2): The observable graph data $\mathbf{Y}$ is obfuscated using Algorithm 2, where $\mathbf{Y}$ is the input to the algorithm to first detect and then obfuscate $\mathbf{U}^*$ and $\mathbf{S}^*$. The obfuscated versions of $\mathbf{U}^*$ and $\mathbf{S}^*$ are $\mathbf{U}$ and $\mathbf{S}$, respectively.

- Obfuscated data (DP-based benchmark): For a fair comparison with Algorithm 2, we start by gauging the variance of the convergent value of $||\mathbf{US}-\mathbf{Y}||_F$ under Algorithm 2. Then, we obfuscate $\mathbf{Y}$ by adding the DP noise to $\mathbf{Y}$, i.e., $\mathbf{Y}+\mathbf{n}_{\mathrm{DP}}$, where $\mathbf{n}_{\mathrm{DP}} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$. In this way, the utility of $\mathbf{Y}$ can be consistent between the proposed Algorithm 2 and the DP-based benchmark.

1)*Convergence Analysis:* Fig. 6.2 plots the convergence behaviors of Algorithm 2 with an increasing number of iterations, where $K = 15$, $M = 300$, and $\beta = 0.1$, 0.5, 1.0, 1.5, and 2.0. Fig. 6.2(a) plots the auxiliary variable, α, under different values of β. Figs. 6.2(b)–6.2(d) plot the changes of $||\mathbf{U}-\mathbf{U}^*||_F$, $||\mathbf{S}-\mathbf{S}^*||_F$, and $||\mathbf{US}-\mathbf{Y}||_F$, respectively. It is observed in Figs. 6.2(a) and 6.2(c) that α and $||\mathbf{S}-\mathbf{S}^*||_F$ grow quickly at the beginning and then gradually converge at around the 20th and 50th iterations, respectively. It is also observed that $||\mathbf{U}-\mathbf{U}^*||_F$ first increases rapidly and then drops slowly and converges. By contrast, $||\mathbf{US}-\mathbf{Y}||_F$ first drops considerably, and then increases slightly and converges, as shown in Figs. 6.2(b) and 6.2(d).

It is also shown in Fig. 6.2 that α, $||\mathbf{S}-\mathbf{S}^*||_F$, $||\mathbf{U}-\mathbf{U}^*||_F$, and $||\mathbf{US}-\mathbf{Y}||_F$ change differently with the value of β. Specifically, α increases, as β grows from 0.1 to 2.0. $||\mathbf{S}-\mathbf{S}^*||_F$ increases as β grows from 0.1 to 1.5 and decreases slightly when $\beta = 2.0$. However, $||\mathbf{U}-\mathbf{U}^*||_F$ and $||\mathbf{US}-\mathbf{Y}||_F$ show less clear dependence on β. Nevertheless, we can choose a reasonable value for β, i.e., $\beta = 1$, that can achieve the smallest value of $||\mathbf{US}-\mathbf{Y}||_F$ and meanwhile maintain large $||\mathbf{U}-\mathbf{U}^*||_F$ and $||\mathbf{S}-\mathbf{S}^*||_F$. Typically, β is specified empirically, depending on the relative importance or preference of the graph structure and stimulus. No additional experiments or computational overhead is needed for specifying β.

Fig. 6.3 plots the changes of $||\mathbf{U}-\mathbf{U}^*||_F$, $||\mathbf{S}-\mathbf{S}^*||_F$ and $||\mathbf{US}-\mathbf{Y}||_F$ with an increasing number of iterations under different values of K, where $N = 30$, and $\beta = 1$. It is observed that $||\mathbf{U}-\mathbf{U}^*||_F$, $||\mathbf{S}-\mathbf{S}^*||_F$, and $||\mathbf{US}-\mathbf{Y}||_F$ increase with K. In other words, the bigger K can lead to better obfuscations of $\mathbf{U}$ and $\mathbf{S}$, but the utility of $\mathbf{Y}$ can be penalized as a cost. It is also observed that the growths rates of $||\mathbf{U}-\mathbf{U}^*||_F$, $||\mathbf{S}-\mathbf{S}^*||_F$, and $||\mathbf{US}-\mathbf{Y}||_F$ decrease with the increase of K. This is because when K is large enough, the graph learned from the observed data contains almost all information, and a further increase of K has little impact on $||\mathbf{U}-\mathbf{U}^*||_F$, $||\mathbf{S}-\mathbf{S}^*||_F$, and $||\mathbf{US}-\mathbf{Y}||_F$.



(a)

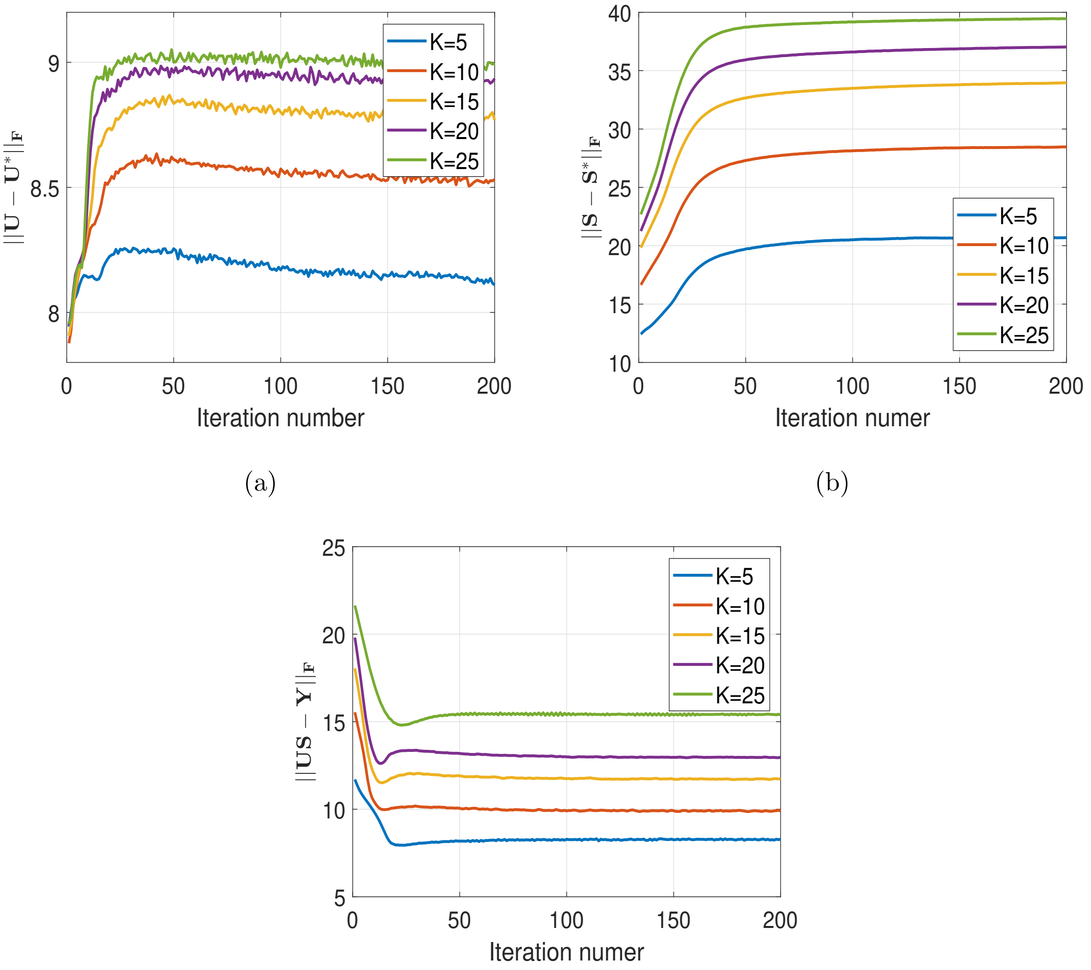
(b)

(c)

(d)

Figure 6.2: The convergence performance of Algorithm 2 under different values of β, where $K = 15$ and $M = 300$.

We note that K is the bandwidth of the observed graph-structured data $\mathbf{Y}$, and is not a parameter of the proposed obfuscation algorithm, i.e., Algorithm 2. The correct identification of K is important to correctly extract the latent information, especially in the presence of non-negligible observation noises. The value of K can be experimentally specified at additional computational overhead, when extracting the latent information $\mathbf{U}^*$ and $\mathbf{S}^*$ from $\mathbf{Y}$. The additional overhead is negligible due to the closed-form expressions derived, i.e., (6.6) and (6.7), that can be used directly to specify K; see **Remark 1**. Nonetheless, the value of K can differ under different detection criteria, e.g., different thresholds used to assess the significance of each spectrum-domain component, as described in **Remark 1**. For generality, we take multiple values for K to assess the proposed algorithm.



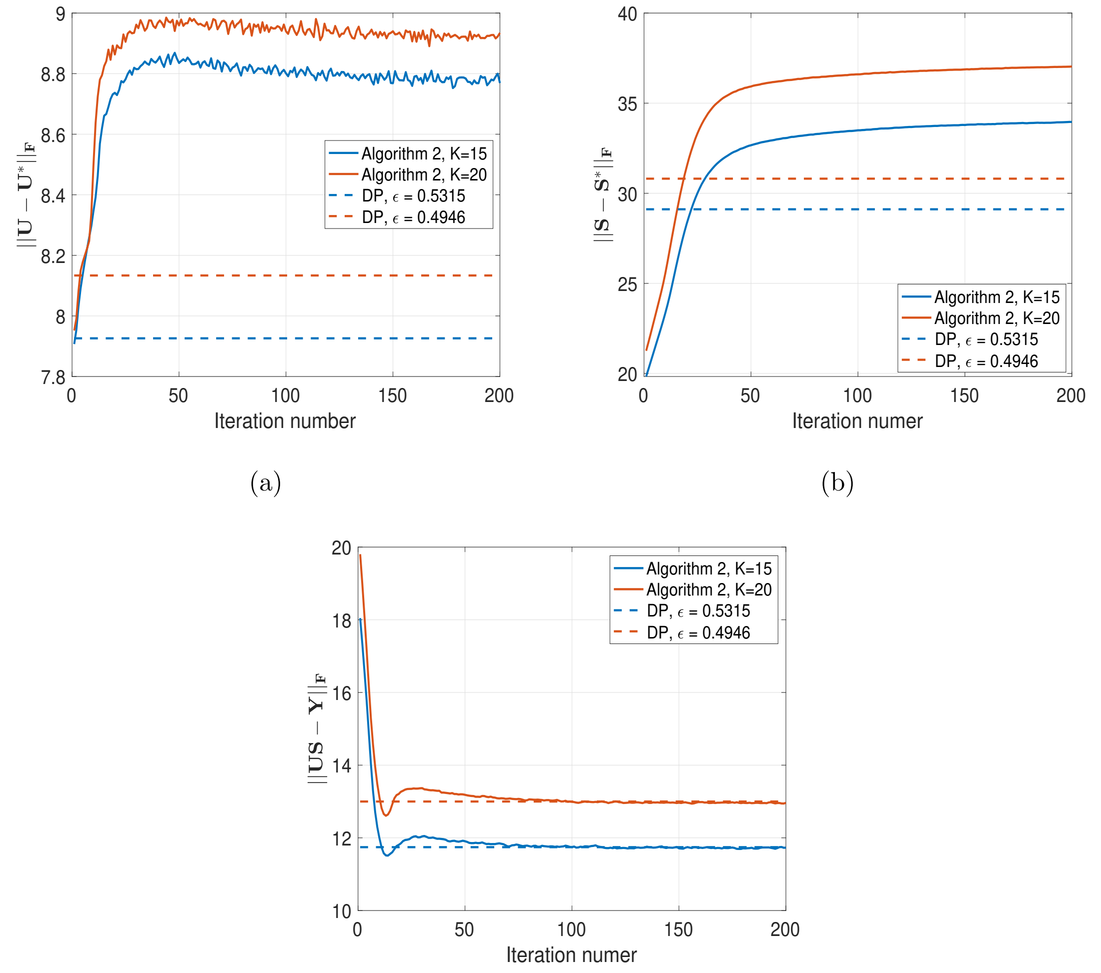
(a) (b)

(c)

Figure 6.3: The performance of Algorithm1 under different values of K, where $\beta = 1.0$.

2) *Comparisons with DP-based Method:* For a fair comparison between Algorithm (2) and the DP-based benchmark, we keep the utility consistent between the two algorithms. Then, we evaluate the privacy budget ϵ required for the DP-based approach to achieve the same perturbation variance as Algorithm 2, i.e., by using [170, Eq. (2)]. As shown in Fig. 6.4(c), the corresponding privacy budget ϵ of the DP-based method is 0.5315 to achieve the same perturbation as Algorithm 2 under $K = 15$, or 0.4946 under $K = 20$. As the privacy budget decreases from 0.5315 to 0.4946, the perturbations of both methods on the graph data and their latent graph structures and stimuli, i.e., $||\mathbf{US}-\mathbf{Y}||_F$, $||\mathbf{U}-\mathbf{U}^*||_F$, and $||\mathbf{S}-\mathbf{S}^*||_F$, increase. As shown in Figs. 6.4(a) and 6.4(b), $||\mathbf{U}-\mathbf{U}^*||_F$ and $||\mathbf{S}-\mathbf{S}^*||_F$ are much larger under Algorithm 2 than they are under the DP-based method, indicating the significantly

stronger perturbations of the latent graph structure $\mathbf{U}^*$ and stimulus $\mathbf{S}^*$ without compromising the utility of the graph data $\mathbf{Y}$ under Algorithm 2.



(a)

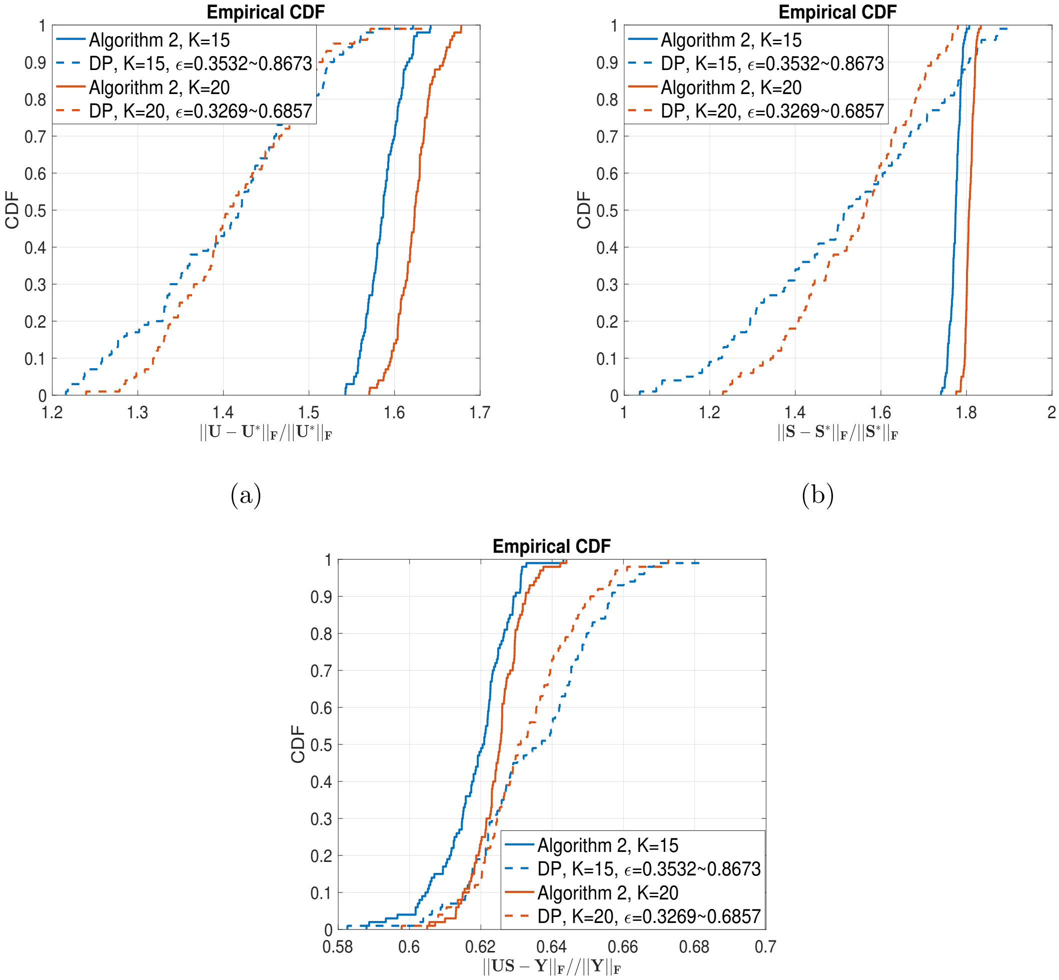
(b)

(c)

Figure 6.4: Comparison between Algorithm 1 and the DP-based method, where $\beta = 1$ and $K = 15, 20$.

Fig. 6.5 compares Algorithm 2 and the DP-based approach by plotting the cumulative distribution functions (CDFs) of their normalized differences between the ground truths and the corresponding perturbed versions, i.e., $\frac{||\mathbf{U}-\mathbf{U}^*||_F}{||\mathbf{U}^*||_F}$, $\frac{||\mathbf{S}-\mathbf{S}^*||_F}{||\mathbf{S}^*||_F}$, and $\frac{||\mathbf{US}-\mathbf{Y}||_F}{||\mathbf{Y}||_F}$. To plot the CDFs of Algorithm 2, we obfuscate 100 independently randomly generated graph-structured data using the algorithm, where K is empirically specified and preconfigured. Under a given K, for each random graph-structured data, the variance of the perturbations on the graph data, i.e., $||\mathbf{US} - \mathbf{Y}||_F$, is evaluated.

To effectively compare Algorithm 2 with the latest DP-based method, we evaluate the privacy budget ϵ required for the DP-based method to achieve the same perturbation variance as Algorithm 2, i.e., by using [170, eq. 2]. Then, we perturb the random graph-structured data using the DP-based approach under the privacy budget, and extract the latent information (i.e., graph structure and stimulus) from the DP-perturbed graph-structured data using the five state-of-the-art graph learning techniques with the K value (if needed). As a result of the randomness of the DP noise, the privacy budget ϵ is within $[0.3523, 0.8673]$ when $K = 15$, and within $[0.3269, 0.6857]$ when $K = 20$. It is observed that $\frac{||\mathbf{U}-\mathbf{U}^*||_F}{||\mathbf{U}^*||_F}$ and $\frac{||\mathbf{S}-\mathbf{S}^*||_F}{||\mathbf{S}^*||_F}$ are consistently better under Algorithm 2 than under the DP-based method, indicating the better obfuscation effect under Algorithm 2 than under the DP-based benchmark.



(a)

(b)

(c)

Figure 6.5: The CDFs of $\frac{||\mathbf{U}-\mathbf{U}^*||_F}{||\mathbf{U}^*||_F}$, $\frac{||\mathbf{S}-\mathbf{S}^*||_F}{||\mathbf{S}^*||_F}$, and $\frac{||\mathbf{US}-\mathbf{Y}||_F}{||\mathbf{Y}||_F}$ under Algorithm 2 and the DP-based benchmark, where $\beta = 1$, and $K = 15, 20$.

Fig. 6.6 plots $\frac{||\mathbf{U}-\mathbf{U}^*||_F}{||\mathbf{U}^*||_F}$ versus $\frac{||\mathbf{S}-\mathbf{S}^*||_F}{||\mathbf{S}^*||_F}$ under Algorithm 2 and the DP-based method, where $\mathbf{n}_{\mathrm{DP}}$ is consistent with Fig. 6.4(c) and the DP-based method is repeated independently for 50 times. It is observed that the DP-based benchmark gives less reliable obfuscation results than Algorithm 2, as it undergoes the randomness of the DP noise. For example, the normalized obfuscation level of $\mathbf{U}$ and $\mathbf{S}$ ranges from 1.3 to 1.8 under the DP-based benchmark, and it is around 1.7 under Algorithm 2, when the benchmark is expected to achieve consistent utility with Algorithm 2. It is also noticed that under Algorithm 1, $\frac{||\mathbf{S}-\mathbf{S}^*||_F}{||\mathbf{S}^*||_F}$ is small when $\beta = 0.1$. With the growth of β, $\frac{||\mathbf{S}-\mathbf{S}^*||_F}{||\mathbf{S}^*||_F}$ increases and tends to stabilize at around 1.85.

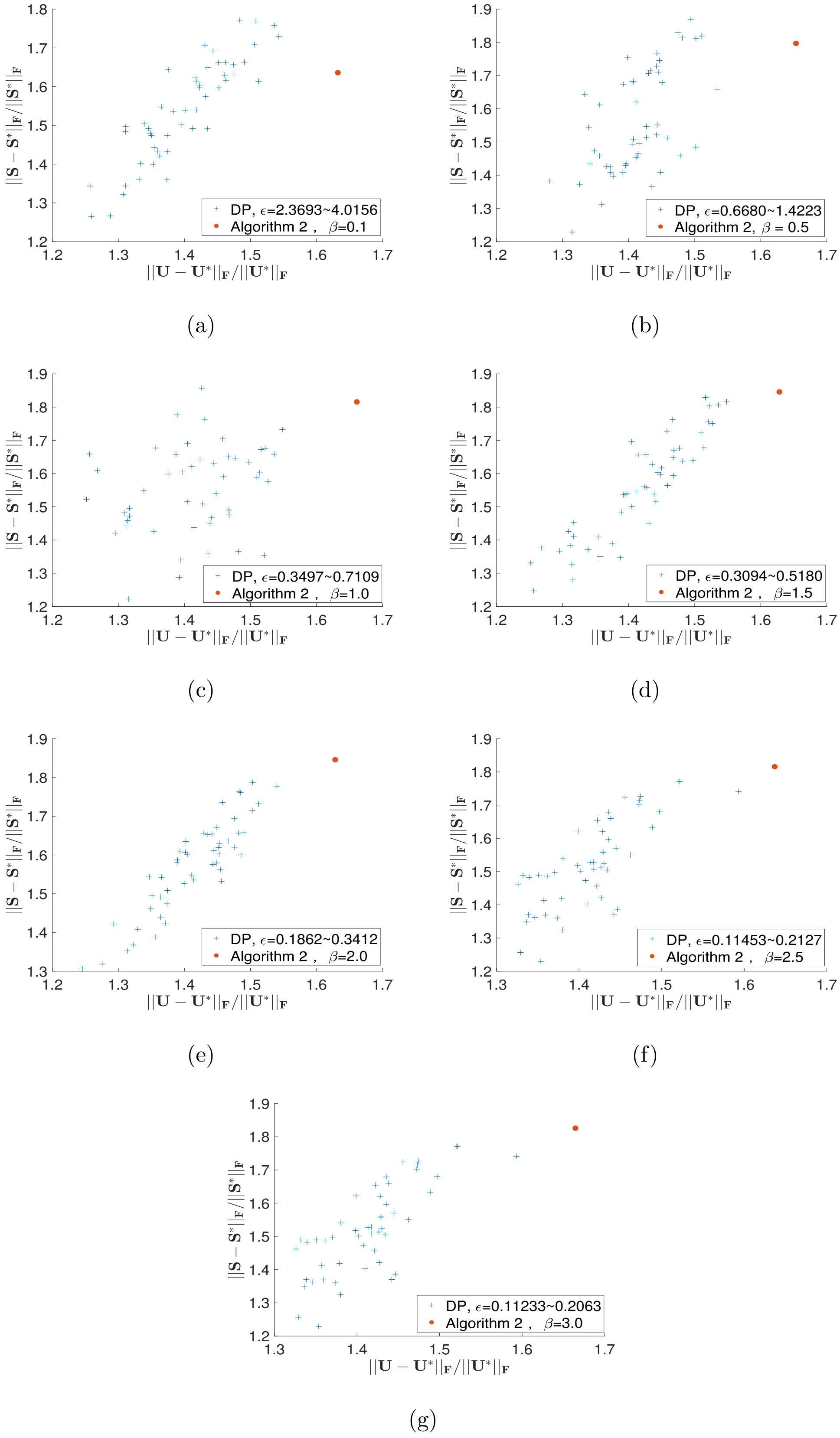
We further compare Algorithm 2 with the DP-based benchmark under different state-of-the-art graph learning techniques utilized for graph inference attacks, including Dong's algorithm [18], Kalofolias' algorithm [19], Sardellitti's TV algorithm [35], Sardellitti's ESA algorithm [35], and Humbert's algorithm [36].

For a fair comparison, all regularization parameters of the graph learning techniques are tested and optimized accordingly. Again, we keep consistent utilities between Algorithm 2 and the DP-based approach, and evaluate their resistance to the latest graph inference attacks. Without loss of generality, we set $\beta = 1$ for Algorithm 2.

As shown in TABLE 6.1, Algorithm 2 is lower by 31.16% and 30.18% than the DP-based approach in F-measure and correlation coefficient $\rho_{\mathbf{W}}$, respectively. In other words, the graphs extracted from the perturbed graph-structured data using the latest graph learning techniques are significantly more dissimilar to their ground truths under Algorithm 2. As a result, the attack success rates of the graph inference attacks can be reduced dramatically.

Fig. 6.7 plots the CDFs of the F-measure and $\rho_{\mathbf{W}}$ values of Algorithm 1 and the DP-based approach under the five graph inference attacks. As shown in the figure, Algorithm 1 is substantially more resistant to the graph inference attacks than the DP-based method due to its significantly weaker similarity between the latent graph structures learned by the graph inference attacks and the ground truths.

Fig. 6.8 plots the attack success rates of the five graph inference attacks on Algorithm 2 and the DP-based approach, where the x-axis provides the threshold of



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 6.6: $\frac{||\mathbf{U}-\mathbf{U}^*||_F}{||\mathbf{U}^*||_F}$ versus $\frac{||\mathbf{S}-\mathbf{S}^*||_F}{||\mathbf{S}^*||_F}$, where Algorithm 2 and the DP-based method are compared under different β values and $K = 20$.

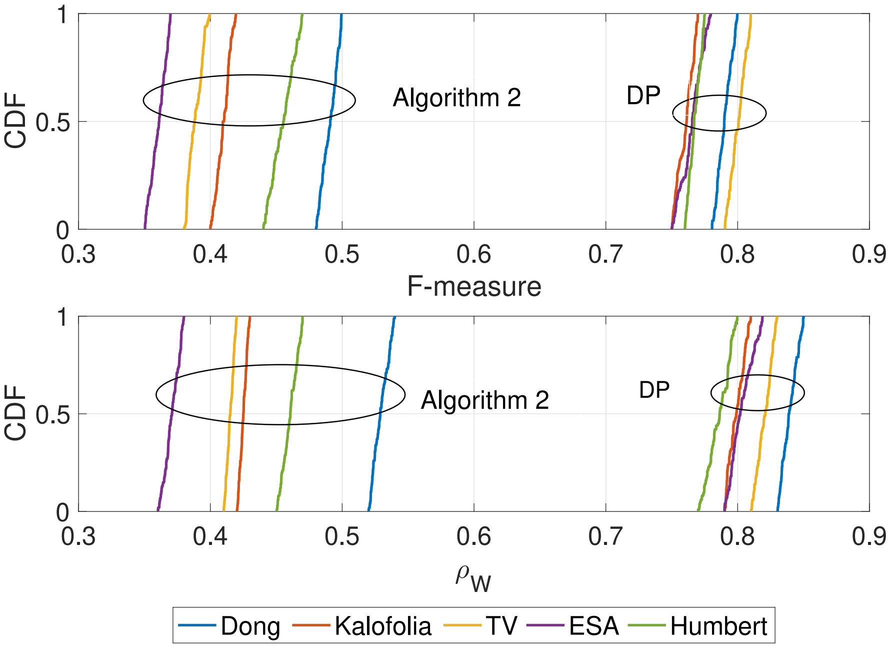


Figure 6.7: CDFs of Algorithm 2 and the DP-based method. $K = 15$, $N = 30$, $M = 300$, and $\beta = 1$.

F-measure or $\rho_{\mathbf{W}}$, above which a graph inference attack is treated as successful. As shown in the figure, Algorithm 2 allows dramatically lower attack success rates for the five attacks. Suppose that a graph inference attack is successful if the F-measure is larger than 0.5; i.e., the F-measure threshold is 0.5. The attack success rates of all five graph interference attacks are 100% under the DP-based approach. By contrast, Algorithm 2 resists all five attacks with an attack success rate of zero.

6.4.2 Results on Attention-Deficit Hyperactivity Disorder

We employ the proposed Algorithm 2 to protect the privacy of a real-world dataset, i.e., the brain functional dataset with ADHD. The studied ADHD dataset comprises the data sample of 42 right-handed boys aged between 11 and 16 years old[1]. Each brain is segmented into 90 ROIs, following an anatomical automatic labeling template. Every node represents an ROI, and there are 232 samples per ROI in the

[1] The dataset is obtained from the ADHD-200 global competition database (`https://www.nitrc.org/projects/neurobureau/`).

Table 6.1: Comparison in utility ($\|\mathbf{US}-\mathbf{Y}\|_F$) and resistance (F-measure and $\rho_{\mathbf{W}}$) between Algorithm 2 and the DP-based benchmark under graph inference attacks launched by the considered five graph learning techniques. $K = 15$, $N = 30$, $M = 300$, and $\beta = 1$.

	Dong [35]	**Kalofolias** [36]	**Sardellitti-TV** [9]	**Sardellitti-ESA** [9]	**Humbert** [10]
Ground-truth					
F-measure	0.8517 (± 0.025)	0.8642 (± 0.022)	0.8125 (± 0.031)	0.8571 (± 0.040)	0.9053 (± 0.019)
$\rho_{\mathbf{W}}$	0.9037 (± 0.017)	0.9102 (± 0.019)	0.9121 (± 0.020)	0.9274 (± 0.027)	0.9217 (± 0.035)
DP					
F-measure	0.7989 (± 0.021)	0.7671 (± 0.039)	0.7927 (± 0.037)	0.7604 (± 0.026)	0.7737 (± 0.043)
$\rho_{\mathbf{W}}$	0.8331 (± 0.037)	0.8035 (± 0.034)	0.8288 (± 0.029)	0.8039 (± 0.046)	0.7865 (± 0.037)
$\|\mathbf{U}-\mathbf{U}^*\|_F$	7.8133 (± 0.119)	7.9537 (± 0.127)	7.8047 (± 0.106)	7.7952 (± 0.201)	7.9249 (± 0.131)
$\|\mathbf{S}-\mathbf{S}^*\|_F$	29.113 (± 0.157)	28.201 (± 0.179)	29.126 (± 0.211)	28.213 (± 0.144)	28.755 (± 0.207)
$\|\mathbf{US}-\mathbf{Y}\|_F$	11.925 (± 0.361)	11.593 (± 0.296)	12.449 (± 0.407)	12.622 (± 0.418)	12.364 (± 0.379)
Algorithm 2					
F-measure	**0.4873 (± 0.011)**	**0.4057 (± 0.026)**	**0.3869 (± 0.014)**	**0.3627 (± 0.021)**	**0.4570 (± 0.015)**
$\rho_{\mathbf{W}}$	**0.5313 (± 0.031)**	**0.4263 (± 0.032)**	**0.4119 (± 0.022)**	**0.3757 (± 0.025)**	**0.4652 (± 0.024)**
$\|\mathbf{U}-\mathbf{U}^*\|_F$	**8.7998 (± 0.125)**	**8.8213 (± 0.241)**	**8.8057 (± 0.209)**	**8.8186 (± 0.117)**	**8.8112 (± 0.126)**
$\|\mathbf{S}-\mathbf{S}^*\|_F$	**33.831 (± 0.241)**	**34.223 (± 0.109)**	**33.982 (± 0.212)**	**34.351 (± 0.153)**	**34.174 (± 0.113)**
$\|\mathbf{US}-\mathbf{Y}\|_F$	**11.743 (± 0.195)**	**11.307 (± 0.228)**	**12.013 (± 0.231)**	**12.147 (± 0.212)**	**11.599 (± 0.239)**

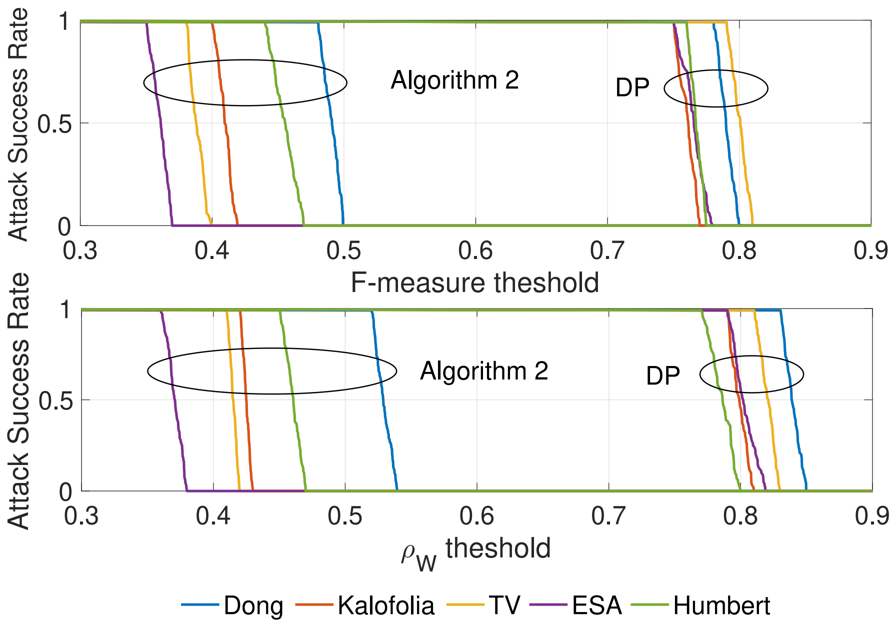


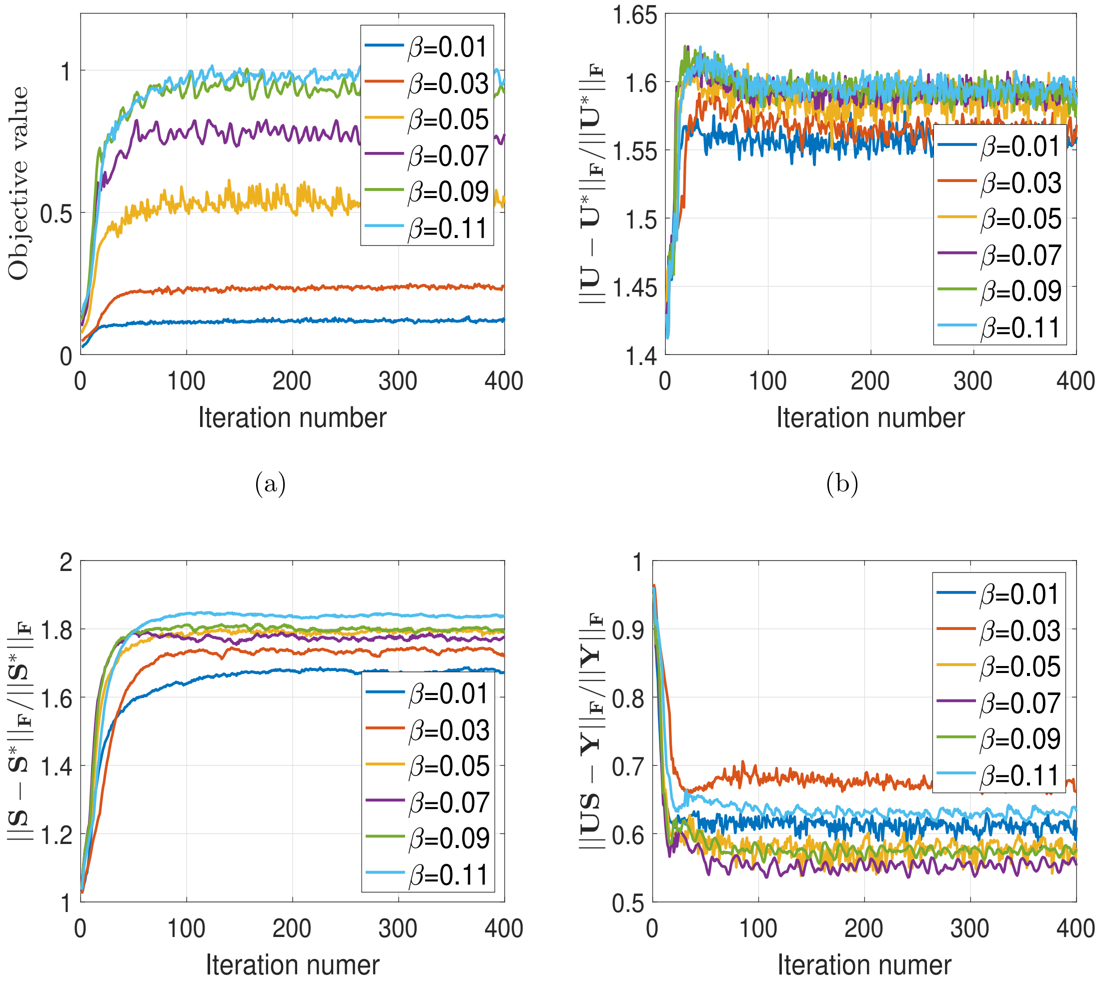
Figure 6.8: Attack success rates of Algorithm 2 and the DP-based method. $K = 15$, $N = 30$, $M = 300$, and $\beta = 1$.

time domain [127]. The observed graph-structured brain data yields $\mathbf{Y} \in \mathbb{R}^{90\times 232}$.

Fig. 6.9 plots the convergence behaviors of α, $\frac{||\mathbf{U}-\mathbf{U}^*||_F}{||\mathbf{U}^*||_F}$, $\frac{||\mathbf{S}-\mathbf{S}^*||_F}{||\mathbf{S}^*||_F}$, and $\frac{||\mathbf{US}-\mathbf{Y}||_F}{||\mathbf{Y}||_F}$ regarding subject 10 in the ADHD dataset, where we set $K = 55$ and different β values are tested. We also plot $\frac{||\mathbf{U}-\mathbf{U}^*||_F}{||\mathbf{U}^*||_F}$ versus $\frac{||\mathbf{S}-\mathbf{S}^*||_F}{||\mathbf{S}^*||_F}$ concerning subject 10 of the dataset in Fig. 6.10. Consistent observations are made here with synthetic data (see Figs. 6.2 and 6.6), indicating the reliability of the Algorithm 2 in practical scenarios.

We also plot the CDF of $\frac{||\mathbf{U}-\mathbf{U}^*||_F}{||\mathbf{U}^*||_F}$, $\frac{||\mathbf{S}-\mathbf{S}^*||_F}{||\mathbf{S}^*||_F}$, and $\frac{||\mathbf{US}-\mathbf{Y}||_F}{||\mathbf{Y}||_F}$ by considering all subjects in the ADHD dataset in Fig. 6.11. We see that the Algorithm 2 consistently achieves better performance than the DP-based method by providing more significant obfuscations on $\mathbf{U}$ and $\mathbf{S}$ in Figs. 6.11(a) and 6.11(b) while maintaining the consistent utility $\mathbf{Y}$ in Fig. 6.11(c).

Using the graph learning algorithm developed in [35], namely, Sardellitti-TV [35], we can learn the brain functional networks from the observed data $\mathbf{Y}$. Then, we



(a)

(b)

(c)

(d)

Figure 6.9: The convergence behavior of Algorithm 2 regarding subject 10 of the ADHD dataset, where $K = 55$.

obfuscate the GFT basis $\mathbf{U}$ and the stimulus $\mathbf{S}$, using Algorithm 2 and its DP-based benchmark. Both the original and the obfuscated brain networks, connecting the ROIs are constructed as weighted adjacent coefficients. We visualize the learned brain networks by the BrainNet Viewer toolbox [128].

Fig. 6.12 visualizes the learned brain functional networks (from the ground truths, and the obfuscated versions based on Algorithm 2 and the DP-based benchmark) from three different views of a brain, i.e., the sagittal view, axial view, and coronal view. To better display the differences between the brain functional networks learned from the ground truth and its obfuscated versions, we transform the brain functional networks into the weighted adjacency matrices and show them in Fig. 6.13. It is found that the weighted adjacency matrix learned from the obfuscated version based on Algorithm 2 differs more substantially from that learned from the ground truth, compared to the obfuscated version based on the DP-based benchmark. In other words, Algorithm 2 can more effectively perturb the GFT basis and stimuli with little impact on the utility of the observed data.

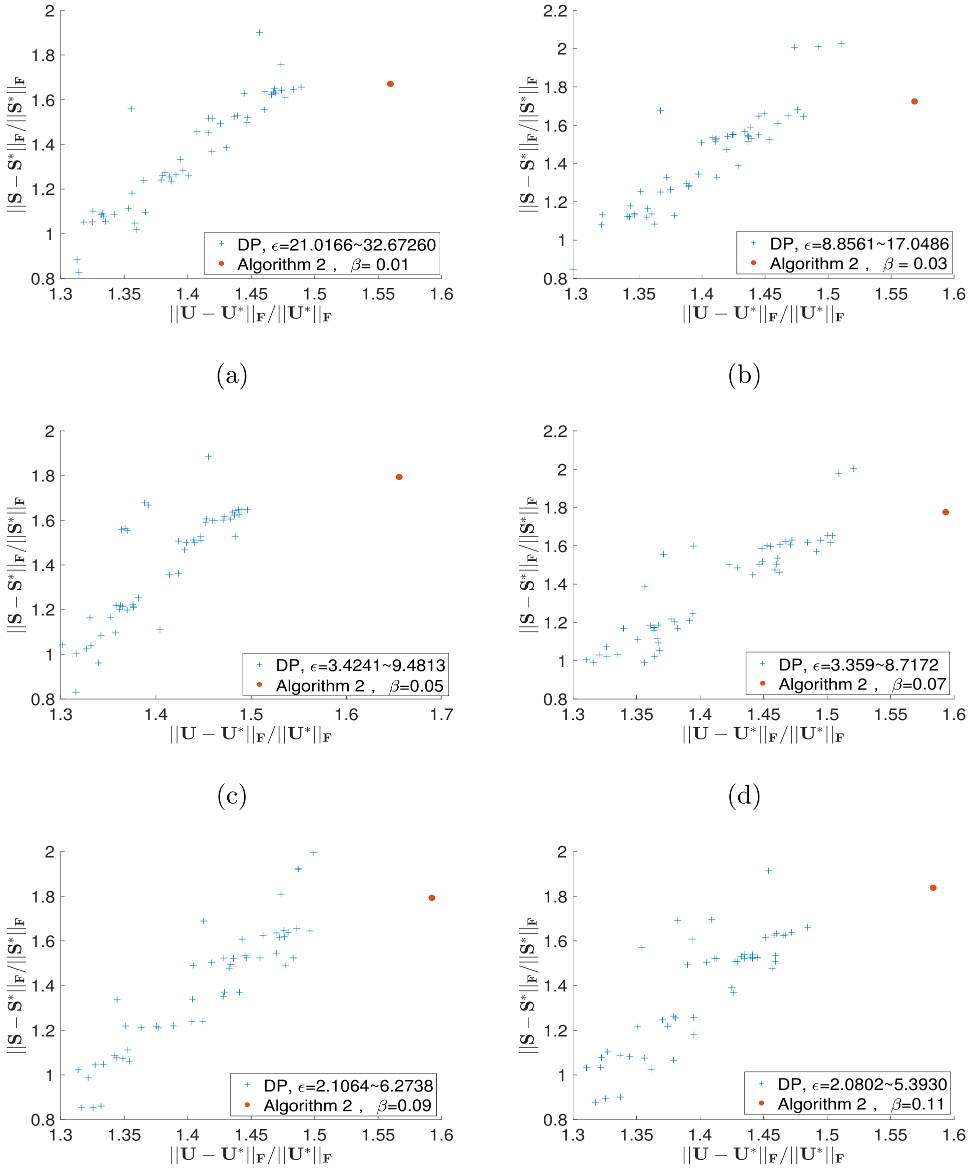
TABLE 6.2 compares Algorithm 2 and the DP-based method in their resistance to the latest graph interference attacks on all subjects in the ADHD dataset. F-measure and $\rho_{\mathbf{W}}$ are used to measure the resistance. For each graph inference method considered, the learned graphs from the original observed data are taken as the ground truths, and then obfuscated using Algorithm 2 and the DP-based benchmark. As shown in the table, Algorithm 1 is more robust to all the considered graph inference attacks than the DP-based method, achieving lower mean scores in both F-measure and $\rho_{\mathbf{W}}$. In particular, TABLE 6.2 shows that Algorithm 2 is at least 36.39% and 36.55% better than the DP-based benchmark in F-measure and $\rho_{\mathbf{W}}$, respectively.

6.5 Concluding Remark

This chapter delivered a new privacy perturbation algorithm to protect the privacy of the latent graph structures and stimuli of graph-structured data. Specifically, we formulated a new multi-objective problem and revealed the limitation of DP. The new algorithm was developed to decouple the multi-objective problem and obfuscate the latent graph structures and stimuli in an alternating manner. Experiments performed on synthetic graph data and the practical ADHD dataset demonstrated that our approach can significantly outperform its DP-based benchmark under graph inference attacks. The approach can have extensive applications to personal health records, finance transactions, and other graph-structured data.

Table 6.2: Comparison between Algorithm 2 and the DP-based method under graph inference attacks launched by the five considered graph learning algorithms on the ADHD dataset, where $K = 55$ and $\beta = 0.05$.

	Dong [18]	**Kalofolias** [19]	**Sardellitti-TV** [35]	**Sardellitti-ESA** [35]	**Humbert** [36]
DP					
F-measure	0.8187 (± 0.021)	0.7832 (± 0.047)	0.7523 (± 0.045)	0.7523 (± 0.028)	0.7773 (± 0.033)
$\rho_{\mathbf{W}}$	0.8431 (± 0.037)	0.8035 (± 0.029)	0.7769 (± 0.045)	0.7617 (± 0.039)	0.7846 (± 0.053)
Algorithm 2					
F-measure	**0.3232 (± 0.042)**	**0.3956 (± 0.018)**	**0.3884 (± 0.042)**	**0.3956 (± 0.034)**	**0.3972 (± 0.027)**
$\rho_{\mathbf{W}}$	**0.3111 (± 0.027)**	**0.4137 (± 0.034)**	**0.3441 (± 0.037)**	**0.3962 (± 0.039)**	**0.4018 (± 0.032)**

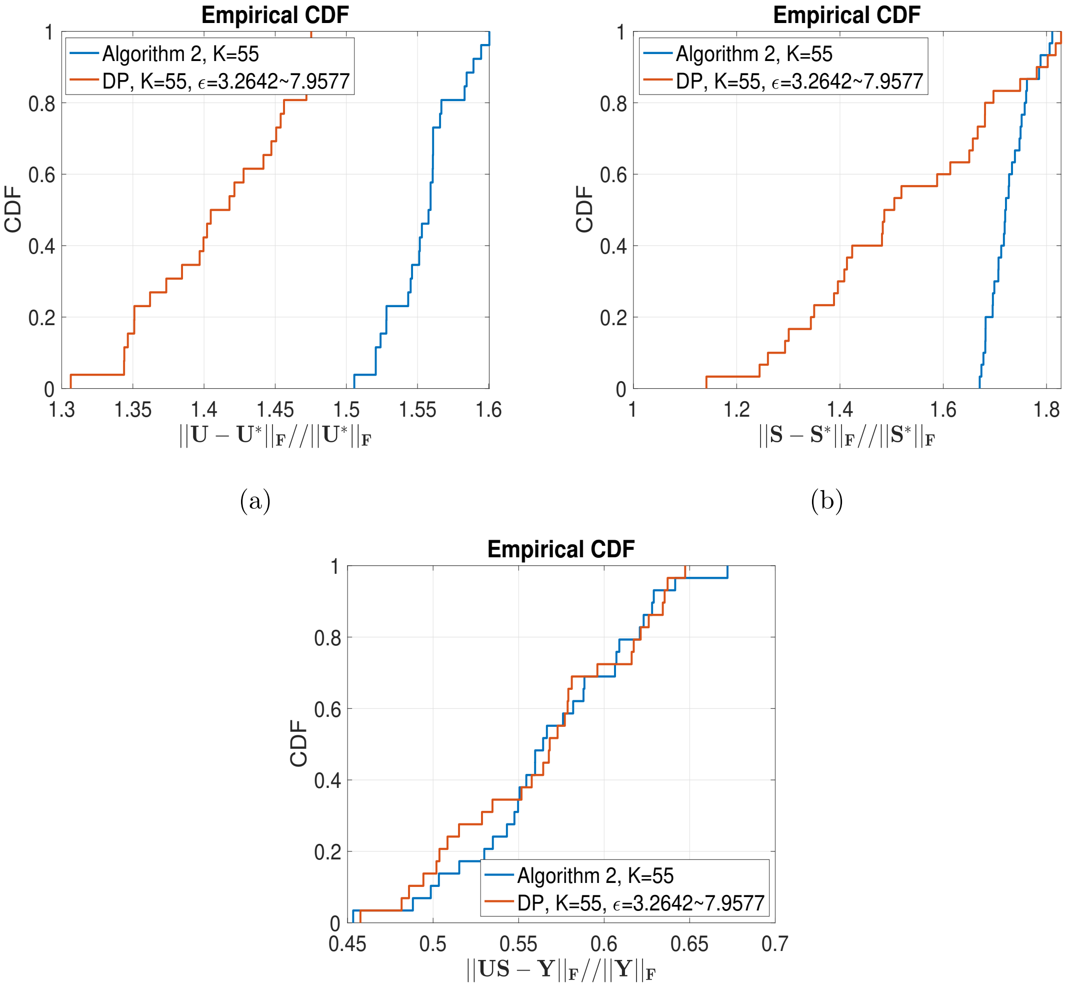


(a) (b)

(c) (d)

(e) (f)

Figure 6.10: $\frac{||\mathbf{U}-\mathbf{U}^*||_F}{||\mathbf{U}^*||_F}$ versus $\frac{||\mathbf{S}-\mathbf{S}^*||_F}{||\mathbf{S}^*||_F}$ for subject 10, where Algorithm 2 and the DP-based method are compared under different values of β, and $K = 55$.

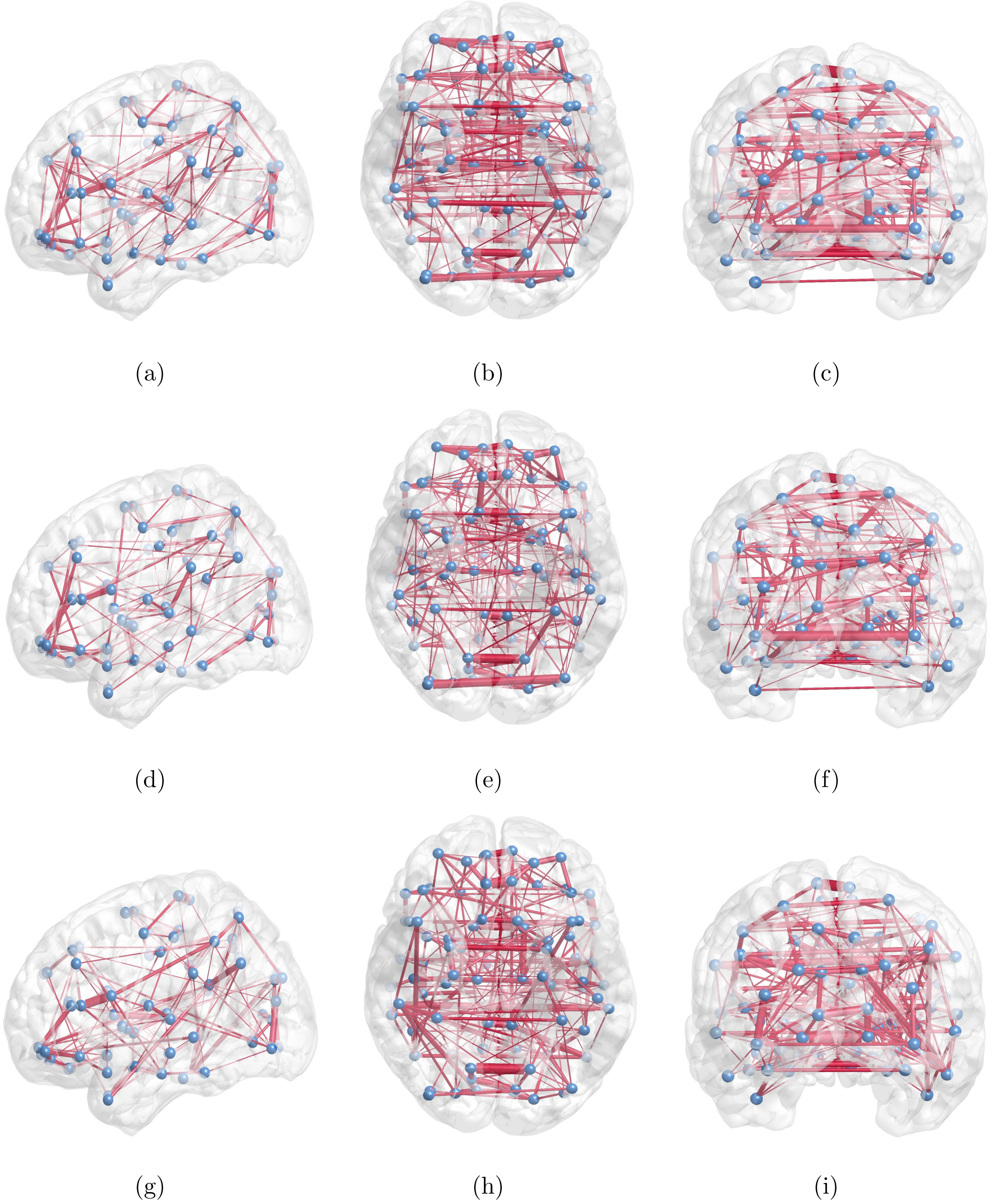


(a)

(b)

(c)

Figure 6.11: The CDFs of $\frac{||\mathbf{U}-\mathbf{U}^*||_F}{||\mathbf{U}^*||_F}$, $\frac{||\mathbf{S}-\mathbf{S}^*||_F}{||\mathbf{S}^*||_F}$, and $\frac{||\mathbf{US}-\mathbf{Y}||_F}{||\mathbf{Y}||_F}$ regarding all subjects in the ADHD dataset.

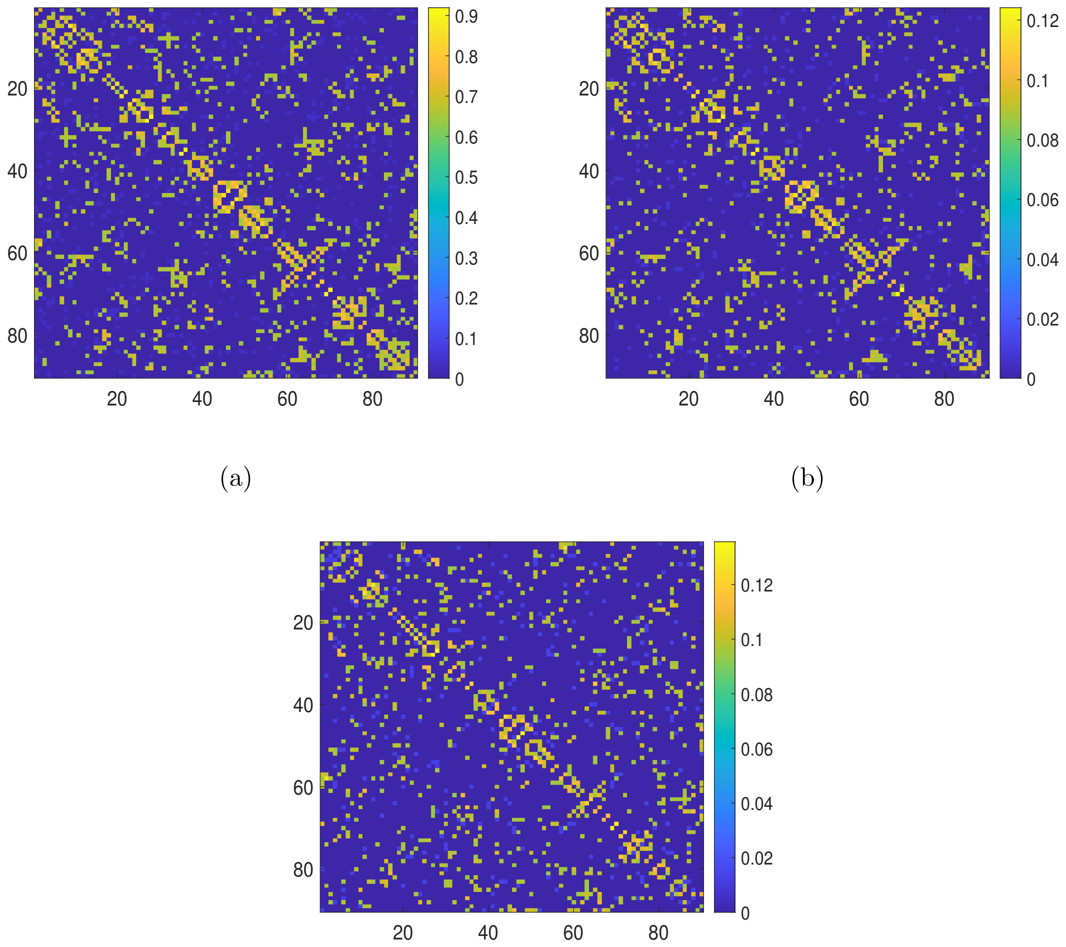


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.12: Visualization of the brain network from the sagittal, axial, and coronal views using BrainNet Viewer. The learned graphs from the data of (a)-(c) the original observation, (d)-(f) the DP-based method, (g)-(i) the proposed Algorithm 1.



(a)

(b)

(c)

Figure 6.13: The weighted adjacency matrices of the learned graphs from the data of (a) the original observation, (b) the DP-based method, and (c) the proposed Algorithm 1.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we introduced a new AO-based graph learning technique to excavate the graphs underlying observed signals. To surpass the fidelity penalized by the AO-based approximation, we proposed a new graph learning technique to efficiently infer the graph structure underlying observed graph signals by deriving a new closed-form analytic expression for the GFT basis. The new graph learning technique was applied to analyze the evolution of the COVID-19 pandemic and reveal the underlying relationship and spreading pattern among different countries. Furthermore, we proposed a new algorithm to protect the privacy of the latent graph structure and stimuli of graph-structured data.

Chapter 3 proposed a graph learning technique, which learned weighted and undirected graph topologies, more specifically, the graph Laplacian matrices, from fMRI-based, band-limited brain signals. Extensive experiments show that our new graph learning technique can efficiently infer the graph of band-limited signals with improved accuracy, compared to the state-of-the-art method. This chapter is supported by the under-reviewed paper at TSIPN [167].

Chapter 4 developed a new graph learning technique to efficiently infer the graph structure of observed band-limited graph signals. The graph Laplacian can be used to analyze network properties or reconstruct missing data resulting from faulty measurements. Simulations and experiments showed that our method outperforms the status quo in accuracy and efficiency. This chapter is supported by the journal publication at SP [171].

Chapter 5 designed a graph learning technique to analyze the evolution of the COVID-19 pandemic and reveal the underlying relationship and spreading pattern among different countries. Extensive numerical tests are performed to assess the graph learning reliability of the new method. Compared with the latest techniques, the new algorithm has the minimum RMSE and the R2 with at least 33.3% and 11.11% improvements, respectively, thereby corroborating the results of our COVID-19 analysis. This chapter is supported by the journal publication at JBHI [172].

Chapter 6 presented a new approach to preserving the privacy of the latent graph structures and stimuli of graph data while maximizing the utility of the graph data. Extensive experiments conducted on synthetic graph data and the real-world ADHD dataset demonstrated that our approach can significantly outperform its DP-based benchmark under graph inference attacks. This chapter is supported by the journal publication at TIFS [173].

7.2 Future Work

Graph learning techniques for time-series data have been widely used in real applications while there are still issues that need to be solved. Moreover, preserving the privacy of graph-structured data is an intricate and challenging undertaking. Therefore, our future work will be focused on the following aspects:

- Dynamic or time-varying networks are of paramount importance in numerous real-world applications, and research endeavors to acquire valuable insights from such data form a crucial aspect of graph learning. In our future works, a primary focus will be devoted to the development of graph learning techniques that effectively capture and model the temporal dynamics inherent in graphs. Specifically, our endeavors will involve the design of dynamic graph networks capable of accurately representing and encapsulating the intricate changes in structure and evolving properties that characterize time-varying networks.

- In our current work, we focused solely on graph learning from time series data, overlooking other important features that are relevant to graph clustering and community detection. Another focus of our future work is to expand our scope to explore graph learning techniques based on multi-feature data, such as the location and air quality in temperature data. Apart from inferring the graph topology, we will uncover underlying patterns, similarities, and relationships among nodes based on their feature attributes to achieve graph clustering and community detection.

- DP provides a systematic and measurable framework for preserving privacy in data analysis. To preserve the privacy of latent graph structures and stimuli of graph-structured data, we will undertake a thorough exploration and refinement of advanced methodologies building upon our existing techniques. Our upcoming endeavors involve conceptualizing and developing a novel method that integrates DP, reinforcing the privacy foundations of graph-structured data. Furthermore, we will extend our focus to the convergence of privacy and federated learning. Here, we will explore innovative approaches to strike a delicate balance between safeguarding individual privacy within graph networks and leveraging the potential of federated learning for holistic insights.

Bibliography

[1] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[2] B. Liu, W. Ni, R. P. Liu, Y. J. Guo, and H. Zhu, "Optimal routing of unmanned aerial vehicle for joint goods delivery and in-situ sensing," *IEEE Trans. Intell. Transp. Syst.*, 2022.

[3] X. Yuan, Z. Feng, W. Ni, Z. Wei, R. P. Liu, and C. Xu, "Connectivity of uav swarms in 3d spherical spaces under (un) intentional ground interference," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8792–8804, 2020.

[4] S. Hu, X. Chen, W. Ni, X. Wang, and E. Hossain, "Modeling and analysis of energy harvesting and smart grid-powered wireless communication networks: A contemporary survey," *IEEE Trans. Green Commun.*, vol. 4, no. 2, pp. 461–496, 2020.

[5] R. Li, L. Millist, E. Foster, *et al.*, "Spike and wave discharges detection in genetic absence epilepsy rat from strasbourg and patients with genetic generalized epilepsy," *Epilepsy Research*, p. 107 181, 2023.

[6] S. Sanei and J. A. Chambers, *EEG signal processing.* John Wiley & Sons, 2013.

[7] K. J. Friston, A. P. Holmes, J. Poline, *et al.*, "Analysis of fMRI time-series revisited," *Neuroimage*, vol. 2, no. 1, pp. 45–53, 1995.

[8] A. K. Jain, S. R. Sahoo, and J. Kaubiyal, "Online social networks security and privacy: Comprehensive review and analysis," *Complex & Intelligent Systems*, vol. 7, no. 5, pp. 2157–2177, 2021.

[9] S. Wang, Y. Zheng, X. Jia, and X. Yi, “PeGraph: A system for privacy-preserving and efficient search over encrypted social graphs,” *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 3179–3194, 2022.

[10] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, “Discrete signal processing on graphs: Sampling theory,” *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, 2015.

[11] A. R. Asadi, E. Abbe, and S. Verdú, “Compressing data on graphs with clusters,” in *Proc. ISIT*, IEEE, 2017, pp. 1583–1587.

[12] S. P. Chepuri and G. Leus, “Graph sampling for covariance estimation,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 451–466, 2017.

[13] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, “Signals on graphs: Uncertainty principle and sampling,” *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4845–4860, 2016.

[14] H. Su, Z. Rong, M. Z. Chen, X. Wang, G. Chen, and H. Wang, “Decentralized adaptive pinning control for cluster synchronization of complex dynamical networks,” *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 394–399, 2012.

[15] A. P. Dempster, “Covariance selection,” *Biometrics*, pp. 157–175, 1972.

[16] J. Friedman, T. Hastie, and R. Tibshirani, “Sparse inverse covariance estimation with the graphical lasso,” *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[17] G. B. Giannakis, Y. Shen, and G. V. Karanikolas, “Topology identification and learning over graphs: Accounting for nonlinearities and dynamics,” *Proc. IEEE*, vol. 106, no. 5, pp. 787–807, 2018.

[18] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, “Learning Laplacian matrix in smooth graph signal representations,” *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, 2016.

[19] V. Kalofolias, “How to learn a graph from smooth signals,” in *Proc. 19th Int. Conf. Artif. Intell. Statist., Cadiz, Spain,*, 2016, pp. 920–929.

[20] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero, “Learning sparse graphs under smoothness prior,” in *Proc. ICASSP*, IEEE, 2017, pp. 6508–6512.

[21] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 467–483, 2017.

[22] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 3, pp. 481–496, 2017.

[23] D. Thanou, X. Dong, D. Kressner, and P. Frossard, "Learning heat diffusion graphs," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 484–499, 2017.

[24] K. Li, W. Ni, X. Yuan, A. Noor, and A. Jamalipour, "Deep-graph-based reinforcement learning for joint cruise control and task offloading for aerial edge internet of things (edgeiot)," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21 676–21 686, 2022.

[25] Y. Liu *et al.*, "Graph learning based on spatiotemporal smoothness for time-varying graph signal," *IEEE Access*, vol. 7, pp. 62 372–62 386, 2019.

[26] R. Li, X. Yuan, M. Radfar, *et al.*, "Graph signal processing, graph neural network and graph learning on biological data: A systematic review," *IEEE Rev. Biomed. Eng.*, 2021.

[27] X. Zhang and S. Mahadevan, "A bio-inspired approach to traffic network equilibrium assignment problem," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1304–1315, 2017.

[28] A. A. Khan, M. Abolhasan, W. Ni, J. Lipman, and A. Jamalipour, "An end-to-end (e2e) network slicing framework for 5g vehicular ad-hoc networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 7, pp. 7103–7112, 2021.

[29] D. Du, X. Li, W. Li, R. Chen, M. Fei, and L. Wu, "Admm-based distributed state estimation of smart grid under data deception and denial of service attacks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 8, pp. 1698–1711, 2019.

[30] X. Lyu, C. Ren, W. Ni, H. Tian, R. P. Liu, and E. Dutkiewicz, "Optimal online data partitioning for geo-distributed machine learning in edge of wire-

less networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2393–2406, 2019.

[31] Y. Krumbeck, Q. Yang, G. W. Constable, and T. Rogers, "Fluctuation spectra of large random dynamical systems reveal hidden structure in ecological networks," *Nat. Commun.*, vol. 12, no. 1, pp. 1–14, 2021.

[32] X. Wang, X. Zha, W. Ni, *et al.*, "Game theoretic suppression of forged messages in online social networks," *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 51, no. 3, pp. 1601–1611, 2019.

[33] S. V. Faraone, T. Banaschewski, D. Coghill, *et al.*, "The world federation of ADHD international consensus statement: 208 evidence-based conclusions about the disorder," *Neurosci. Biobehav. Rev.*, vol. 128, pp. 789–818, 2021.

[34] M. Filippi, S. Basaia, E. Canu, *et al.*, "Changes in functional and structural brain connectome along the alzheimer's disease continuum," *Molecular psychiatry*, vol. 25, no. 1, pp. 230–239, 2020.

[35] S. Sardellitti *et al.*, "Graph topology inference based on sparsifying transform learning," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1712–1727, 2019.

[36] P. Humbert, B. Le Bars, L. Oudre, A. Kalogeratos, and N. Vayatis, "Learning laplacian matrix from graph signals with sparse spectral representation," *J. Mach. Learn. Res.*, vol. 22, no. 195, pp. 1–47, 2021.

[37] A. D. Sarwate and K. Chaudhuri, "Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data," *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 86–94, 2013.

[38] I. E. Olatunji, W. Nejdl, and M. Khosla, "Membership inference attack on graph neural networks," in *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, IEEE, 2021, pp. 11–20.

[39] G. Mateos *et al.*, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 16–43, 2019.

[40] B. Ma, X. Wang, W. Ni, and R. P. Liu, "Personalized location privacy with road network-indistinguishability," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20 860–20 872, 2022.

[41] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, “Learning graphs from data: A signal representation perspective,” *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, 2019.

[42] S. Gao, X. Xia, D. Scheinost, and G. Mishne, “Smooth graph learning for functional connectivity estimation,” *NeuroImage*, vol. 239, p. 118 289, 2021.

[43] S. S. Saboksayr, G. Mateos, and M. Cetin, “Online discriminative graph learning from multi-class smooth signals,” *Signal Process.*, vol. 186, p. 108 101, 2021.

[44] S. Chen *et al.*, “Signal recovery on graphs: Fundamental limits of sampling strategies,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 4, pp. 539–554, 2016.

[45] P. Ferrer-Cid *et al.*, “Graph learning techniques using structured data for iot air pollution monitoring platforms,” *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13 652–13 663, 2021.

[46] C. Hu *et al.*, “Matched signal detection on graphs: Theory and application to brain imaging data classification,” *NeuroImage*, vol. 125, pp. 587–600, 2016.

[47] K. Zhan, C. Zhang, J. Guan, and J. Wang, “Graph learning for multiview clustering,” *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 2887–2895, 2017.

[48] F. Della Rossa, D. Salzano, A. Di Meglio, *et al.*, “A network model of italy shows that intermittent regional strategies can alleviate the covid-19 epidemic,” *Nat. commun.*, vol. 11, no. 1, p. 5106, 2020.

[49] T. Suzumura, H. Kanezashi, M. Dholakia, *et al.*, “The impact of covid-19 on flight networks,” in *2020 ieee international conference on big data (big data)*, IEEE, 2020, pp. 2443–2452.

[50] D. Tsiotas and V. Tselios, “Understanding the uneven spread of covid-19 in the context of the global interconnected economy,” *Sci. Rep.*, vol. 12, no. 1, pp. 1–15, 2022.

[51] Z. Feng, C. Xiao, P. Li, Z. You, X. Yin, and F. Zheng, “Comparison of spatio-temporal transmission characteristics of covid-19 and its mitigation strategies in china and the us,” *J. Geogr. Sci.*, vol. 30, no. 12, pp. 1963–1984, 2020.

[52] D. C. dos Santos Gomes and G. L. de Oliveira Serra, "Machine learning model for computational tracking and forecasting the covid-19 dynamic propagation," *IEEE J. Biomed. Health informat.*, vol. 25, no. 3, pp. 615–622, 2021.

[53] M. K. So, A. Tiwari, A. M. Chu, J. T. Tsang, and J. N. Chan, "Visualizing covid-19 pandemic risk through network connectedness," *Int. J. Infect. Dis.*, vol. 96, pp. 558–561, 2020.

[54] A. Kuzdeuov, D. Baimukashev, A. Karabay, *et al.*, "A network-based stochastic epidemic simulator: Controlling covid-19 with region-specific policies," *IEEE J. Biomed. Health informat.*, vol. 24, no. 10, pp. 2743–2754, 2020.

[55] M. Milano, "Cctv: A new network-based methodology for the analysis and visualization of covid-19 data," in *Proc. Int. Conf. Bioinformat. Biomed. (BIBM)*, IEEE, 2021, pp. 2000–2001.

[56] K. Demertzis, D. Tsiotas, and L. Magafas, "Modeling and forecasting the covid-19 temporal spread in greece: An exploratory approach based on complex network defined splines," *Int. J. Environ. Res. Public Health*, vol. 17, no. 13, p. 4693, 2020.

[57] M. K. So, A. M. Chu, A. Tiwari, and J. N. Chan, "On topological properties of covid-19: Predicting and assessing pandemic risk with network statistics," *Sci. Rep.*, vol. 11, no. 1, pp. 1–14, 2021.

[58] A. Neşe and H. Bakir, "Spatiotemporal analysis of covid-19 in turkey," *Sustain. Cities Soc.*, vol. 76, p. 103 421, 2022.

[59] N. Kianfar, M. S. Mesgari, A. Mollalo, and M. Kaveh, "Spatio-temporal modeling of covid-19 prevalence and mortality using artificial neural network algorithms," *Spatial and Spatio-temporal Epidemiology*, vol. 40, p. 100 471, 2022.

[60] W. M. Dlamini, S. P. Simelane, and N. M. Nhlabatsi, "Bayesian network-based spatial predictive modelling reveals covid-19 transmission dynamics in eswatini," *Spat. Inf. Res.*, vol. 30, no. 1, pp. 183–194, 2022.

[61] M. Yuan, L. Chen, S. Y. Philip, and T. Yu, "Protecting sensitive labels in social network data anonymization," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 633–647, 2011.

[62] Y. Li, X. Tao, X. Zhang, M. Wang, and S. Wang, "Break the data barriers while keeping privacy: A graph differential privacy method," *IEEE Internet Things J.*, 2022.

[63] C. Wei, S. Ji, C. Liu, W. Chen, and T. Wang, "AsgLDP: Collecting and generating decentralized attributed graphs with local differential privacy," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3239–3254, 2020.

[64] W. Lin, B. Li, and C. Wang, "Towards private learning on decentralized graphs with local differential privacy," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 2936–2946, 2022.

[65] Z. Liu, L. Zhang, W. Ni, and I. B. Collings, "Uncoordinated pseudonym changes for privacy preserving in distributed networks," *IEEE Trans. Mob. Comput.*, vol. 19, no. 6, pp. 1465–1477, 2019.

[66] C. Dwork, A. Roth, *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[67] S. K. Narang and A. Ortega, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4673–4685, 2013.

[68] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, 2014.

[69] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3462–3477, 2017.

[70] M. Coutino, E. Isufi, and G. Leus, "Advances in distributed graph filtering," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2320–2333, 2019.

[71] S. Chen, A. Sandryhaila, J. M. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4609–4624, 2015.

[72] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal process. mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[73] P. Di Lorenzo *et al.*, "Adaptive graph signal processing: Algorithms and optimal sampling strategies," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3584–3598, 2018.

[74] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, 2013.

[75] D. Jin, X. Wang, M. Liu, J. Wei, W. Lu, *et al.*, "Identification of generalized semantic communities in large social networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2966–2979, 2020.

[76] Y. Wang, T. Li, S. Li, X. Yuan, and W. Ni, "New adversarial image detection based on sentiment analysis," *IEEE Internet Things J.*, 2023.

[77] H. Huang, J. Zhang, L. Yu, J. Zhang, Q. Wu, and C. Xu, "Toan: Target-oriented alignment network for fine-grained image categorization with few labeled samples," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 853–866, 2021.

[78] I. Jabłoński, "Graph signal processing in applications to sensor networks, smart grids, and smart cities," *IEEE Sens. J.*, vol. 17, no. 23, pp. 7659–7666, 2017.

[79] Y. Emami, B. Wei, K. Li, W. Ni, and E. Tovar, "Joint communication scheduling and velocity control in multi-uav-assisted sensor networks: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10 986–10 998, 2021.

[80] K. Li, W. Ni, H. Kurunathan, and F. Dressler, "Data-driven deep reinforcement learning for online flight resource allocation in uav-aided wireless powered sensor networks," in *ICC 2022-IEEE International Conference on Communications*, IEEE, 2022, pp. 1–6.

[81] S. Hu, X. Chen, W. Ni, E. Hossain, and X. Wang, "Distributed machine learning for wireless communication networks: Techniques, architectures, and applications," *IEEE Commun. Surv. Tutor.*, vol. 23, no. 3, pp. 1458–1493, 2021.

[82] X. Li, L. Lu, W. Ni, A. Jamalipour, D. Zhang, and H. Du, "Federated multi-agent deep reinforcement learning for resource allocation of vehicle-to-vehicle

communications," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8810–8824, 2022.

[83] J. Dai, K. Huang, Y. Liu, C. Yang, and Z. Wang, "Global reconstruction of complex network topology via structured compressive sensing," *IEEE Syst. J.*, vol. 15, no. 2, pp. 1959–1969, 2020.

[84] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constraints," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 825–841, 2017.

[85] H. E. Egilmez, Y.-H. Chao, and A. Ortega, "Graph-based transforms for video coding," *IEEE Trans. Image Process.*, vol. 29, pp. 9330–9344, 2020.

[86] G. Marrelec and P. Fransson, "Assessing the influence of different ROI selection strategies on functional connectivity analyses of fMRI data acquired during steady-state conditions," *PloS one*, vol. 6, no. 4, e14788, 2011.

[87] W. Ni, I. B. Collings, J. Lipman, X. Wang, M. Tao, and M. Abolhasan, "Graph theory and its applications to future network planning: Software-defined online small cell management," *IEEE Wirel. Commun.*, vol. 22, no. 1, pp. 52–60, 2015.

[88] K. M. Albert, G. G. Potter, B. D. Boyd, H. Kang, and W. D. Taylor, "Brain network functional connectivity and cognitive performance in major depressive disorder," *J. Psychiatr. Res.*, vol. 110, pp. 51–56, 2019.

[89] W. K. Chai, "Modelling spreading process induced by agent mobility in complex networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 5, no. 4, pp. 336–349, 2017.

[90] B. Liu, Z. Li, X. Chen, Y. Huang, and X. Liu, "Recognition and vulnerability analysis of key nodes in power grid based on complex network centrality," *IEEE Trans. Circuits and Syst. II, Express Briefs*, vol. 65, no. 3, pp. 346–350, 2017.

[91] J. Zhang, L. Zhou, and L. Wang, "Subject-adaptive integration of multiple SICE brain networks with different sparsity," *Pattern Recognit.*, vol. 63, pp. 642–652, 2017.

[92] P. Aggarwal and A. Gupta, “Low rank and sparsity constrained method for identifying overlapping functional brain networks,” *Plos one*, vol. 13, no. 11, e0208068, 2018.

[93] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, “Graph topology inference based on transform learning,” in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, 2016, pp. 356–360.

[94] P. Humbert, B. Le Bars, L. Oudre, A. Kalogeratos, and N. Vayatis, “Learning Laplacian matrix from graph signals with sparse spectral representation,” Tech. Rep., 2019.

[95] X. Chen, Z. Feng, J. A. Zhang, X. Yuan, and P. Zhang, “Integrated sensing and communication complex cnn csi enhancer for 6g networks,” *arXiv preprint arXiv:2305.17938*, 2023.

[96] K. Zhao, B. Duka, H. Xie, D. J. Oathes, V. Calhoun, and Y. Zhang, “A dynamic graph convolutional neural network framework reveals new insights into connectome dysfunctions in adhd,” *NeuroImage*, vol. 246, p. 118 774, 2022.

[97] X. Li, Y. Zhou, N. Dvornek, *et al.*, “Braingnn: Interpretable brain graph neural network for fmri analysis,” *Medical Image Analysis*, vol. 74, p. 102 233, 2021.

[98] H. Zhou, L. He, Y. Zhang, L. Shen, and B. Chen, “Interpretable graph convolutional network of multi-modality brain imaging for alzheimer’s disease diagnosis,” in *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2022, pp. 1–5.

[99] S. Azad and S. Devi, “Tracking the spread of covid-19 in india via social networks in the early phase of the pandemic,” *J. Travel Med.*, vol. 27, no. 8, taaa130, 2020.

[100] W. Jo, D. Chang, M. You, and G.-H. Ghim, “A social network analysis of the spread of covid-19 in south korea and policy implications,” *Sci. Rep.*, vol. 11, no. 1, pp. 1–10, 2021.

[101] A. M. Chu, J. N. Chan, J. T. Tsang, A. Tiwari, and M. K. So, “Analyzing cross-country pandemic connectedness during covid-19 using a spatial-

temporal database: Network analysis," *JMIR Public Health Surveill.*, vol. 7, no. 3, e27317, 2021.

[102] A. M. Chu, T. W. Chan, M. K. So, and W.-K. Wong, "Dynamic network analysis of covid-19 with a latent pandemic space model," *Int. J. Environ. Res. Public Health*, vol. 18, no. 6, p. 3195, 2021.

[103] S. Syga, D. David-Rus, Y. Schälte, H. Hatzikirou, and A. Deutsch, "Inferring the effect of interventions on covid-19 transmission networks," *Sci. Rep.*, vol. 11, no. 1, pp. 1–11, 2021.

[104] T. McMahon, A. Chan, S. Havlin, and L. K. Gallos, "Spatial correlations in geographical spreading of covid-19 in the united states," *Sci. Rep.*, vol. 12, no. 1, pp. 1–10, 2022.

[105] Y. Pan, L. Zhang, J. Unwin, and M. J. Skibniewski, "Discovering spatial-temporal patterns via complex networks in investigating covid-19 pandemic in the united states," *Sustain. Cities Soc.*, vol. 77, p. 103 508, 2022.

[106] R. Alguliyev, R. Aliguliyev, and F. Yusifov, "Graph modelling for tracking the covid-19 pandemic spread," *Infect. Dis. Model.*, vol. 6, pp. 112–122, 2021.

[107] C. Ieracitano, N. Mammone, M. Versaci, *et al.*, "A fuzzy-enhanced deep learning approach for early detection of covid-19 pneumonia from portable chest x-ray images," *Neurocomputing*, vol. 481, pp. 202–215, 2022.

[108] N. Absar, B. Mamur, A. Mahmud, *et al.*, "Development of a computer-aided tool for detection of covid-19 pneumonia from cxr images using machine learning algorithm," *JRRAS*, vol. 15, no. 1, pp. 32–43, 2022.

[109] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing social networks," *Computer science department faculty publication series*, p. 180, 2007.

[110] X. Ding, C. Wang, K.-K. R. Choo, and H. Jin, "A novel privacy preserving framework for large scale graph data publishing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 331–343, 2019.

[111] F. Yu, M. Chen, B. Yu, W. Li, L. Ma, and H. Gao, "Privacy preservation based on clustering perturbation algorithm for social network," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 11 241–11 258, 2018.

[112] W.-Y. Day, N. Li, and M. Lyu, "Publishing graph degree distribution with node differential privacy," in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 123–138.

[113] H. Huang, D. Zhang, F. Xiao, K. Wang, J. Gu, and R. Wang, "Privacy-preserving approach PBCN in social network with differential privacy," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 2, pp. 931–945, 2020.

[114] Q. Li, R. Heusdens, and M. G. Christensen, "Privacy-preserving distributed optimization via subspace perturbation: A general framework," *IEEE Trans. Signal Process.*, vol. 68, pp. 5983–5996, 2020.

[115] J. He, L. Cai, C. Zhao, P. Cheng, and X. Guan, "Privacy-preserving average consensus: Privacy analysis and algorithm design," *IEEE Trans. Signal Inform. Process. Netw.*, vol. 5, no. 1, pp. 127–138, 2018.

[116] Q. Li, M. Coutino, G. Leus, and M. G. Christensen, "Privacy-preserving distributed graph filtering," in *2020 28th European Signal Processing Conference (EUSIPCO)*, IEEE, 2021, pp. 2155–2159.

[117] A. Kucyi, A. Tambini, S. Sadaghiani, S. Keilholz, and J. R. Cohen, "Spontaneous cognitive processes and the behavioral validation of time-varying brain connectivity," *Netw. Neurosci.*, vol. 2, no. 4, pp. 397–417, 2018.

[118] K. Duan, W. Jiang, K. Rootes-Murdy, *et al.*, "Gray matter networks associated with attention and working memory deficit in adhd across adolescence and adulthood," *Translational psychiatry*, vol. 11, no. 1, pp. 1–12, 2021.

[119] B. Le Bars, P. Humbert, L. Oudre, and A. Kalogeratos, "Learning Laplacian matrix from bandlimited graph signals," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, IEEE, 2019, pp. 2937–2941.

[120] D. Yang, W. Ni, L. Du, H. Liu, and J. Wang, "Efficient attributed scatter center extraction based on image-domain sparse representation," *IEEE Trans. Signal Process.*, vol. 68, pp. 4368–4381, 2020.

[121] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 5302–5316, 2009.

[122] H. Oviedo, O. Dalmau, and H. Lara, "Two adaptive scaled gradient projection methods for stiefel manifold constrained optimization," *Numerical Algorithms*, vol. 87, no. 3, pp. 1107–1127, 2021.

[123] L. Zhao *et al.*, "Optimization algorithms for graph Laplacian estimation via ADMM and MM," *IEEE Trans. Signal Process.*, vol. 67, no. 16, pp. 4231–4244, 2019.

[124] J. Dall and M. Christensen, "Random geometric graphs," *Physical review E*, vol. 66, no. 1, p. 016 121, 2002.

[125] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, no. 1, pp. 17–60, 1960.

[126] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[127] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, *et al.*, "Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain," *Neuroimage*, vol. 15, no. 1, pp. 273–289, 2002.

[128] M. Xia, J. Wang, and Y. He, "Brainnet viewer: A network visualization tool for human brain connectomics," *PloS one*, vol. 8, no. 7, e68910, 2013.

[129] R. eare, C. Adamson, M. A. Bellgrove, V. Vilgis, A. Vance, *et al.*, "Altered structural connectivity in ADHD: A network based analysis," *Brain Imag Behav.*, vol. 11, no. 3, pp. 846–858, 2017.

[130] S.-B. Hong, A. Zalesky, A. Fornito, S. Park, Y.-H. Yang, M.-H. Park, *et al.*, "Connectomic disturbances in attention-deficit/hyperactivity disorder: A whole-brain tractography analysis," *Biological psychiatry*, vol. 76, no. 8, pp. 656–663, 2014.

[131] Q. Cao, N. Shu, L. An, P. Wang, L. Sun, M.-R. Xia, *et al.*, "Probabilistic diffusion tractography and graph theory analysis reveal abnormal white matter structural connectivity networks in drug-naive boys with attention deficit/hyperactivity disorder," *Journal of Neuroscience*, vol. 33, no. 26, pp. 10 676–10 687, 2013.

[132] D. Li, T. Li, Y. Niu, *et al.*, “Reduced hemispheric asymmetry of brain anatomical networks in attention deficit hyperactivity disorder,” *Brain imaging and behavior*, vol. 13, no. 3, pp. 669–684, 2019.

[133] H.-M. Eadeh, K. E. Markon, J. T. Nigg, and M. A. Nikolas, “Evaluating the viability of neurocognition as a transdiagnostic construct using both latent variable models and network analysis,” *Research on Child and Adolescent Psychopathology*, vol. 49, no. 6, pp. 697–710, 2021.

[134] Z. Kang, Z. Lin, X. Zhu, and W. Xu, “Structured graph learning for scalable subspace clustering: From single view to multiview,” *IEEE Trans. Cybern.*, 2021.

[135] Z. Wang, Z. Lin, T. Lv, and W. Ni, “Energy-efficient resource allocation in massive mimo-noma networks with wireless power transfer: A distributed admm approach,” *IEEE Internet Things J.*, vol. 8, no. 18, pp. 14 232–14 247, 2021.

[136] R. Okui, “Asymptotically unbiased estimation of autocovariances and autocorrelations with long panel data,” *Econometric Theory*, vol. 26, no. 5, pp. 1263–1304, 2010.

[137] R.-C. Li, “Matrix perturbation theory,” in *Handbook of linear algebra*, Chapman and Hall/CRC, 2006, pp. 15–1.

[138] World Health Organization. “Coronavirus disease (covid-19) pandemic.” (), [Online]. Available: `https://www.who.int/emergencies/diseases/novel-coronavirus-2019` (visited on 2022).

[139] World Health Organization. “Coronavirus disease (covid-19): Variants of sars-cov-2.” (), [Online]. Available: `https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease` (visited on 2022).

[140] G. Guo, Z. Liu, S. Zhao, L. Guo, and T. Liu, “Eliminating indefiniteness of clinical spectrum for better screening covid-19,” *IEEE J. Biomed. Health informat.*, vol. 25, no. 5, pp. 1347–1357, 2021.

[141] Y. Han, L. Yang, K. Jia, *et al.*, “Spatial distribution characteristics of the covid-19 pandemic in beijing and its relationship with environmental factors,” *Sci. Total Environ.*, vol. 761, p. 144 257, 2021.

[142] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[143] X. Tang, J. Wang, J. Zhong, and Y. Pan, "Predicting essential proteins based on weighted degree centrality," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 11, no. 2, pp. 407–418, 2013.

[144] P. Crescenzi, G. D'angelo, L. Severini, and Y. Velaj, "Greedily improving our own closeness centrality in a network," *ACM Trans. Knowl. Discov. Data*, vol. 11, no. 1, pp. 1–32, 2016.

[145] U. Brandes, "A faster algorithm for betweenness centrality," *J. Math. Sociol.*, vol. 25, no. 2, pp. 163–177, 2001.

[146] J. Zhou, L. Liu, W. Wei, and J. Fan, "Network representation learning: From preprocessing, feature extraction to node embedding," *ACM Comput. Surv.*, vol. 55, no. 2, pp. 1–35, 2022.

[147] K. Berahmand, E. Nasiri, M. Rostami, and S. Forouzandeh, "A modified deepwalk method for link prediction in attributed social network," *Computing*, vol. 103, pp. 2227–2249, 2021.

[148] H. Van Lierde, T. W. Chow, and G. Chen, "Scalable spectral clustering for overlapping community detection in large-scale networks," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 754–767, 2019.

[149] Y. Gong, S. Liu, and Y. Bai, "Efficient parallel computing on the game theory-aware robust influence maximization problem," *Knowl. Based Syst.*, vol. 220, p. 106 942, 2021.

[150] C.-C. Yen, M.-Y. Yeh, and M.-S. Chen, "An efficient approach to updating closeness centrality and average path length in dynamic networks," in *2013 IEEE 13th International Conference on Data Mining*, IEEE, 2013, pp. 867–876.

[151] H. Shirouyehzad, J. Jouzdani, and M. Khodadadi Karimvand, "Fight against covid-19: A global efficiency evaluation based on contagion control and medical treatment," *J. appl. res. ind. eng.*, vol. 7, no. 2, pp. 109–120, 2020.

[152] Y. Deng, Y. Zhang, and K. Wang, “An analysis of the chinese scheduled freighter network during the first year of the covid-19 pandemic,” *J. Transp. Geogr.*, vol. 99, p. 103 298, 2022.

[153] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How powerful are graph neural networks?” *arXiv preprint arXiv:1810.00826*, 2018.

[154] X. Li, Q. Cui, Q. Xue, W. Ni, J. Guo, and X. Tao, “A new batch access scheme with global qos optimization for satellite-terrestrial networks,” in *GLOBECOM 2022-2022 IEEE Global Communications Conference*, IEEE, 2022, pp. 3929–3934.

[155] K. Li, Q. Cui, Z. Zhu, W. Ni, and X. Tao, “Lightweight, privacy-preserving handover authentication for integrated terrestrial-satellite networks,” in *ICC 2022-IEEE International Conference on Communications*, IEEE, 2022, pp. 25–31.

[156] Q. Cui, Y. Wang, K.-C. Chen, *et al.*, “Big data analytics and network calculus enabling intelligent management of autonomous vehicles in a smart city,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2021–2034, 2018.

[157] M. Abolhasan, J. Lipman, N. Shariati, W. Ni, A. Jamalipour, *et al.*, “Joint mobile node participation and multihop routing for emerging open radio-based intelligent transportation system,” *IEEE Access*, vol. 10, pp. 85 228–85 242, 2022.

[158] X. Wang, Y. Wang, Q. Cui, K.-C. Chen, and W. Ni, “Machine learning enables radio resource allocation in the downlink of ultra-low latency vehicular networks,” *IEEE Access*, vol. 10, pp. 44 710–44 723, 2022.

[159] Q. Cui, X. Hu, W. Ni, *et al.*, “Vehicular mobility patterns and their applications to internet-of-vehicles: A comprehensive survey,” *Sci. China Inf. Sci.*, vol. 65, no. 11, pp. 1–42, 2022.

[160] K. Li, R. C. Voicu, S. S. Kanhere, W. Ni, and E. Tovar, “Energy efficient legitimate wireless surveillance of uav communications,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2283–2293, 2019.

[161] X. Yuan, Z. Feng, J. A. Zhang, *et al.*, “Spatio-temporal power optimization for mimo joint communication and radio sensing systems with training overhead,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 514–528, 2020.

[162] M. A. Raza, M. Abolhasan, J. Lipman, N. Shariati, W. Ni, and A. Jamalipour, “Statistical learning-based adaptive network access for the industrial internet-of-things,” *IEEE Internet Things J.*, 2023.

[163] Q. Cui, W. Ni, S. Li, B. Zhao, R. P. Liu, and P. Zhang, “Learning-assisted clustered access of 5g/b5g networks to unlicensed spectrum,” *IEEE Wirel. Commun.*, vol. 27, no. 1, pp. 31–37, 2020.

[164] B. Liu, W. Ni, R. P. Liu, Q. Zhu, Y. J. Guo, and H. Zhu, “Novel integrated framework of unmanned aerial vehicle and road traffic for energy-efficient delay-sensitive delivery,” *IEEE Trans. Intell. Transp. Syst.s*, vol. 23, no. 8, pp. 10 692–10 707, 2021.

[165] X. Chen, H. Wen, W. Ni, *et al.*, “Distributed online optimization of edge computing with mixed power supply of renewable energy and smart grid,” *IEEE Trans. Commun.*, vol. 70, no. 1, pp. 389–403, 2021.

[166] I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman, and W. Ni, “Privysharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities,” *Comput. Secur.*, vol. 88, p. 101 653, 2020.

[167] B. Shan, W. Ni, X. Yuan, D. Yang, X. Wang, and R. P. Liu, “Graph learning from band-limited data by graph Fourier transform analysis,” *Signal Process.*, vol. 207, p. 108 950, 2023.

[168] K. Shen and W. Yu, “Fractional programming for communication systems—Part I: Power control and beamforming,” *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, 2018.

[169] J.-y. Gotoh, A. Takeda, and K. Tono, “DC formulations and algorithms for sparse optimization problems,” *Mathematical Programming*, vol. 169, no. 1, pp. 141–176, 2018.

[170] J. Zhao, T. Wang, T. Bai, *et al.*, “Reviewing and improving the gaussian mechanism for differential privacy,” *arXiv preprint arXiv:1911.12060*, 2019.

[171] B. Shan, X. Yuan, W. Ni, X. Wang, and R. P. Liu, “Raph learning from band-limited data by graph fourier transform analysis,” *IEEE Trans. Signal Inf. Process. Netw*, under reviewed.

[172] B. Shan, X. Yuan, W. Ni, X. Wang, R. P. Liu, and E. Dutkiewicz, "Novel graph topology learning for spatio-temporal analysis of covid-19 spread," *IEEE J. Biomed. Health Inform*, vol. 27, no. 6, pp. 2693–2704, 2023.

[173] B. Shan, X. Yuan, W. Ni, X. Wang, R. P. Liu, and E. Dutkiewicz, "Preserving the privacy of latent information for graph-structured data," *IEEE Trans. Inf. Forensics Secur*, vol. 18, pp. 5041–5055, 2023.

[174] S. Mittal and P. Meer, "Conjugate gradient on grassmann manifolds for robust subspace estimation," *Image Vis. Comput.*, vol. 30, no. 6-7, pp. 417–427, 2012.

Appendix A

Appendix for Chapter 3

The augmented Lagrangian of (3.37) is given by

$$\begin{aligned}\mathcal{L}\left(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \mathbf{C}, \mathbf{Z}\right) &= \operatorname{tr}\left(\mathbf{\Lambda}_1 \widetilde{\mathbf{T}}_1\right) + \operatorname{tr}\left(\mathbf{\Lambda}_2 \widetilde{\mathbf{T}}_2\right) - \log\det\left(\mathbf{\Lambda}_1\right) \\ &- \log\det\left(\mathbf{\Lambda}_2\right) + \operatorname{tr}\Big(\mathbf{Z}^T\left(\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T - \mathbf{C}\right)\Big) \\ &+ \frac{\rho}{2}\left\|\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T - \mathbf{C}\right\|_F^2, \end{aligned} \tag{A.1}$$

where ρ is the step size.

By following the standard ADMM framework, we can update the primal variables, $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$, and $\mathbf{C}$, and the dual variable $\mathbf{Z}$ in an alternating manner in the following steps.

A.1 Update $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$

Based on the primal variables $\mathbf{\Lambda}_1'$, $\mathbf{\Lambda}_2'$, and $\mathbf{C}^l$, and the dual variable $\mathbf{Z}'$ obtained in the last iteration, the primal variable in the current iteration, i.e., $\mathbf{\Lambda}_1^{l+1}$, can be obtained by minimizing the augmented Lagrangian:

$$\begin{aligned}\mathbf{\Lambda}_1 &= \underset{\mathbf{\Lambda}_1 \succeq \mathbf{0}}{\operatorname{argmin}} \mathcal{L}\left(\mathbf{\Lambda}_1', \mathbf{\Lambda}_2', \mathbf{C}', \mathbf{Z}'\right) \\ &= \underset{\mathbf{\Lambda}_1 \succeq \mathbf{0}}{\operatorname{argmin}} \left\{\frac{\rho}{2}\left\|\mathbf{\Lambda}_1' + \frac{1}{\rho}(\widetilde{\mathbf{T}}_1 + \widetilde{\mathbf{Z}}_1 - \rho\widetilde{\mathbf{X}}_1)\right\|_F^2 - \log\det\left(\mathbf{\Lambda}_1'\right)\right\}, \end{aligned} \tag{A.2}$$

where $\widetilde{\mathbf{Z}}_1 = \mathbf{V}_1^T(\mathbf{Z}')^T\mathbf{V}_1$ and $\widetilde{\mathbf{X}}_1 = \mathbf{V}_1^T(\mathbf{C}' - \mathbf{V}_2\mathbf{\Lambda}_2'\mathbf{V}_2^T)\mathbf{V}_1$.

To solve (A.2), we set the first-order derivative of $\frac{\rho}{2}\left\|\mathbf{\Lambda}_1' + \frac{1}{\rho}(\widetilde{\mathbf{T}}_1 + \widetilde{\mathbf{Z}}_1 - \rho\widetilde{\mathbf{X}}_1)\right\|_F^2 - \log\det\left(\mathbf{\Lambda}_1'\right)$ to $\mathbf{0}$, i.e.,

$$\rho\mathbf{\Lambda}_1' - (\mathbf{\Lambda}_1')^{\dagger} = -\left(\widetilde{\mathbf{T}}_1 + \widetilde{\mathbf{Z}}_1 - \rho\widetilde{\mathbf{X}}_1\right). \tag{A.3}$$

Taking the orthogonal eigenvalue decomposition of the right-hand side (RHS) of (A.3) yields $\rho\mathbf{\Lambda}_1' - (\mathbf{\Lambda}_1')^{\dagger} = -\mathbf{Q}_1\mathbf{\Xi}_1\mathbf{Q}_1^T$, Then, we left multiply $\mathbf{Q}_1^T$ and right multiply $\mathbf{Q}_1$ on both sides and obtain $\rho\widetilde{\mathbf{\Lambda}}_1' - (\widetilde{\mathbf{\Lambda}}_1')^{\dagger} = -\mathbf{\Xi}_1 \ \widetilde{\Lambda}_{1i}' = \frac{-\rho\xi_{1i} + \sqrt{\rho^2\xi_{1i}^2 + 4\rho}}{2\rho}$. The optimal solution to (A.2) is given by

$$\mathbf{\Lambda}_1 = \mathbf{Q}_1\widetilde{\mathbf{\Lambda}}_1'\mathbf{Q}_1^T, \tag{A.4}$$

with the diagonal matrix $\widetilde{\mathbf{\Lambda}}_1' = \mathrm{diag}(\widetilde{\Lambda}_{11}', \ldots, \widetilde{\Lambda}_{1K}')$. Given $\mathbf{\Lambda}_1$, the primal variable in $(l+1)$-th iteration, e.g., $\mathbf{\Lambda}_2$, can be obtained in the same way as $\mathbf{\Lambda}_1$.

A.2 Update C

We proceed to update the primal variable $\mathbf{C}$. Given the updated primal variables $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ in the current iteration, $\mathbf{C}$ is obtained as

$$\begin{aligned}\mathbf{C} &= \underset{\mathbf{C}}{\mathrm{argmin}}\ \mathcal{L}\left(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \mathbf{C}', \mathbf{Z}'\right) \qquad (A.5)\\ &= \underset{\mathbf{C}}{\mathrm{argmin}}\left\{\frac{\rho}{2}\left\|\mathbf{C}' - \left(\frac{1}{\rho}(\mathbf{Z}')^T + \mathbf{V}_1\mathbf{\Lambda}_1'\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2'\mathbf{V}_2^T\right)\right\|_F^2\right\}.\end{aligned}$$

Consider the constraints $\mathbf{I}\odot\mathbf{C} \geq \mathbf{0}$ and $\mathbf{A}\odot\mathbf{C} \leq \mathbf{0}$ in (3.37); the diagonal elements are non-negative and the off-diagonal elements are non-positive. We obtain the solution to (A.5):

$$\begin{aligned}\mathbf{C} = \mathbf{I}\odot&\left[\frac{1}{\rho}(\mathbf{Z}')^T + \mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T\right]_+\\ +\mathbf{A}\odot&\left[\frac{1}{\rho}(\mathbf{Z}')^T + \mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T\right]_-.\end{aligned} \tag{A.6}$$

A.3 Update Z

Given the primary variables $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$ and $\mathbf{C}$, the dual variable $\mathbf{Z}$ can be updated by

$$\mathbf{Z} = \mathbf{Z}' + \rho\left(\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T - \mathbf{C}\right). \tag{A.7}$$

Appendix B

Appendix for Chapter 4

B.1

The equivalence is proved between the orthogonality requirement and the orthogonal projection, as follows.

1) Proof of sufficiency condition: Suppose that $\mathbf{P}_{\mathbf{U}_{\mathcal{K}}} = \mathbf{U}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T$ is the orthogonal projection on the subspace spanned by the column-vectors of $\mathbf{U}_{\mathcal{K}}$. $\mathbf{P}_{\mathbf{U}_{\mathcal{K}}}\mathbf{U}_{\mathcal{K}} = \mathbf{U}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}} = \mathbf{U}_{\mathcal{K}}$. By multiplying $\mathbf{U}_{\mathcal{K}}^T$ to the left of both sides of $\mathbf{U}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}} = \mathbf{U}_{\mathcal{K}}$, we obtain

$$\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}} = \mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}}. \tag{B.1}$$

Since $\mathbf{U}_{\mathcal{K}}$ is a full column rank matrix, $\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}}$ is a non-singular matrix. By multiplying $(\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}})^{-1}$ to the left of both sides of (B.1), we have

$$(\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}})^{-1}(\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}})(\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}}) = (\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}})^{-1}\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}}\mathbf{I}. \tag{B.2}$$

As a result, we conclude that $\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}} = \mathbf{I}$, i.e., $\mathbf{U}_{\mathcal{K}}$ is orthonormal.

2) Proof of necessary condition: Suppose that $\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}} = \mathbf{I}$. Based on the definition of orthogonal projection, we have $\mathbf{P}_{\mathbf{U}_{\mathcal{K}}} = \mathbf{U}_{\mathcal{K}}(\mathbf{U}_{\mathcal{K}}^T\mathbf{U}_{\mathcal{K}})^{-1}\mathbf{U}_{\mathcal{K}}^T = \mathbf{U}_{\mathcal{K}}\mathbf{U}_{\mathcal{K}}^T$. In other words, $P_{\mathbf{U}_{\mathcal{K}}}$ is the orthogonal projection on the subspace spanned by the column-vectors of $\mathbf{U}_{\mathcal{K}}$. In summary, the orthogonal requirement and the orthogonal projection are equivalent. The orthogonality requirement is preserved in (4.12) by using the orthogonal projection $\mathbf{P}_{\mathbf{U}_{\mathcal{K}}}$.

B.2

Problem (4.15) is an unconstrained optimization problem that can be converted to a problem defined on a Grassmann manifold. The Grassmann manifold is a closed set, the maximum or minimum of a continuous function defined on the closed set, e.g., the optimal solution to (4.15), exists [174].

Suppose that the eigenvalues of $\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T$ are $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N$, corresponding to the eigenvectors $\{\mathbf{v}_1, \cdots, \mathbf{v}_N\}$. Let $S_1 = \operatorname{span}\{\mathbf{v}_1, \cdots, \mathbf{v}_K\}$, corresponding to the K largest eigenvalues $\{\sigma_1, \cdots, \sigma_K\}$. Let $S_2\,(S_2 \neq S_1)$ be any other K-dimensional subspace. Let $E_0 = S_1 \cap S_2$ and suppose that $S_1 = E_0 \oplus E_1$ and $S_2 = E_0 \oplus E_2$, where E_1 is the subset of S_1 and E_2 is the subset of $S_1^{\perp}$, i.e., $E_1 \subset \operatorname{span}\{\mathbf{v}_1, \cdots, \mathbf{v}_K\}$ and $E_2 \subset \operatorname{span}\{\mathbf{v}_{K+1}, \cdots, \mathbf{v}_N\}$; and $S_1^{\perp}$ is the orthogonal complement of S_1. Suppose that $\dim(E_1) = \dim(E_2) = t$. Based on the Minimax theorem,

$$\begin{aligned}
&\operatorname{tr}\left(\mathbf{P}_{E_1}\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T\right) \geq t\sigma_K;\\
&\operatorname{tr}\left(\mathbf{P}_{E_2}\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T\right) \leq t\sigma_{K+1}.
\end{aligned} \tag{B.3}$$

As a result, we have

$$\begin{aligned}
&\operatorname{tr}\left(\mathbf{P}_{S_1}\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T\right)\\
&=\operatorname{tr}\left(\mathbf{P}_{E_0}\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T\right)+\operatorname{tr}\left(\mathbf{P}_{E_1}\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T\right)\\
&\geq\operatorname{tr}\left(\mathbf{P}_{E_0}\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T\right)+\operatorname{tr}\left(\mathbf{P}_{E_2}\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T\right)\\
&=\operatorname{tr}\left(\mathbf{P}_{S_2}\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T\right),
\end{aligned} \tag{B.4a}$$

where the two equations are based on $S_1 = E_0 \oplus E_1$ and $S_2 = E_0 \oplus E_2$. Hence, the projection of $\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)$ is the largest on the span associated with the K largest eigenvalues. The solution to (4.15) is $\mathbf{U}_{\mathcal{K}}^{*} = \operatorname{span}\{\mathbf{v}_1, \cdots, \mathbf{v}_K\}$, i.e., the K largest eigenvalues of $\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)\mathbf{Y}\mathbf{Y}^T\left(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right)^T$.

B.3

ADMM is an augmented Lagrangian method. With the step size ρ (e.g., $\rho = 1$ by default in Section 4.6), the augmented Lagrangian of (4.26) is given by

$$\mathcal{L}\left(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \mathbf{C}, \mathbf{Z}\right) = \mathrm{tr}\left(\mathbf{\Lambda}_1 \widetilde{\mathbf{T}}_1\right) + \mathrm{tr}\left(\mathbf{\Lambda}_2 \widetilde{\mathbf{T}}_2\right) - M \log\det\left(\mathbf{\Lambda}_1\right) \tag{B.5a}$$

$$- M \log\det\left(\mathbf{\Lambda}_2\right) + \mathrm{tr}\Big(\mathbf{Z}^T\left(\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T - \mathbf{C}\right)\Big) \tag{B.5b}$$

$$+ \frac{\rho}{2}\left\|\mathbf{V}_1\mathbf{\Lambda}_1\mathbf{V}_1^T + \mathbf{V}_2\mathbf{\Lambda}_2\mathbf{V}_2^T - \mathbf{C}\right\|_F^2, \tag{B.5c}$$

where $\mathbf{Z}$ is the dual variable associated with (4.26c).

Following the standard ADMM framework, we update the primal variables, $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$, and $\mathbf{C}$, and the dual variable $\mathbf{Z}$ in an alternating manner:

Update $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$: Based on $\mathbf{\Lambda}_2^l$, $\mathbf{C}^l$, and $\mathbf{Z}^l$ obtained in the l-th iteration, $\mathbf{\Lambda}_1^{l+1}$ in the $(l+1)$-th iteration can be obtained by

$$\begin{aligned}\mathbf{\Lambda}_1^{l+1} &= \arg\min\nolimits_{\mathbf{\Lambda}_1 \succeq \mathbf{0}} \mathcal{L}\left(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2^l, \mathbf{C}^l, \mathbf{Z}^l\right) \\ &= \arg\min\nolimits_{\mathbf{\Lambda}_1 \succeq \mathbf{0}}\left\{\frac{\rho}{2}\left\|\mathbf{\Lambda}_1 + \frac{1}{\rho}(\widetilde{\mathbf{T}}_1 + \widetilde{\mathbf{Z}}_1 - \rho\widetilde{\mathbf{X}}_1)\right\|_F^2 - M\log\det\left(\mathbf{\Lambda}_1^l\right)\right\},\end{aligned} \tag{B.6}$$

where $\widetilde{\mathbf{Z}}_1 = \mathbf{V}_1^T(\mathbf{Z}^l)^T\mathbf{V}_1$ and $\widetilde{\mathbf{X}}_1 = \mathbf{V}_1^T(\mathbf{C}^l - \mathbf{V}_2\mathbf{\Lambda}_2^l\mathbf{V}_2^T)\mathbf{V}_1$. By setting the first-order derivative of $\mathcal{L}\left(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2^l, \mathbf{C}^l, \mathbf{Z}^l\right)$ with respect to $\mathbf{\Lambda}_1$ to $\mathbf{0}$, we have

$$\rho\mathbf{\Lambda}_1 - M(\mathbf{\Lambda}_1)^\dagger = -(\widetilde{\mathbf{T}}_1 + \widetilde{\mathbf{Z}}_1 - \rho\widetilde{\mathbf{X}}_1). \tag{B.7}$$

Taking the orthogonal eigenvalue decomposition of the right-hand side of (B.7) yields $\rho\mathbf{\Lambda}_1 - M(\mathbf{\Lambda}_1)^\dagger = \mathbf{Q}_1\mathbf{\Xi}_1\mathbf{Q}_1^T$. Then, we left multiply $\mathbf{Q}_1^T$ and right multiply $\mathbf{Q}_1$ on both sides, yielding $\rho\widetilde{\mathbf{\Lambda}}_1 - M(\widetilde{\mathbf{\Lambda}}_1)^\dagger = \mathbf{\Xi}_1$. $\widetilde{\mathbf{\Lambda}}_1 = \mathbf{Q}_1^T\mathbf{\Lambda}_1\mathbf{Q}_1$. Here, $\mathbf{Q}_1$ is the unitary matrix with $\mathbf{Q}_1\mathbf{Q}_1^T = \mathbf{I}$. The diagonal matrix $\mathbf{\Xi}_1 = \mathrm{diag}(\xi_{11}, \cdots, \xi_{1K})$ collects the eigenvalues of $(\widetilde{\mathbf{T}}_1 + \widetilde{\mathbf{Z}}_1 - \rho\widetilde{\mathbf{X}}_1)$. Using the quadratic formula, the non-negative solution to $\rho\widetilde{\mathbf{\Lambda}}_1 - M(\widetilde{\mathbf{\Lambda}}_1)^\dagger = \mathbf{\Xi}_1$ is $\widetilde{\Lambda}_{1i} = \frac{-\rho\xi_{1i} + \sqrt{\rho^2\xi_{1i}^2 + 4M\rho}}{2\rho}$. With $\widetilde{\mathbf{\Lambda}}_1 = \mathrm{diag}(\widetilde{\Lambda}_{11}, \cdots, \widetilde{\Lambda}_{1K})$, the solution to (B.6) is

$$\mathbf{\Lambda}_1^{l+1} = \mathbf{Q}_1\widetilde{\mathbf{\Lambda}}_1\mathbf{Q}_1^T, \tag{B.8}$$

Given $\mathbf{\Lambda}_1^{l+1}$, the primal variable $\mathbf{\Lambda}_2^{l+1}$ can be obtained in the same way.

Update $\mathbf{C}$*:* Given $\mathbf{\Lambda}_1^{l+1}$ and $\mathbf{\Lambda}_2^{l+1}$ in the $(l+1)$-th iteration, we obtain

$$\begin{aligned}&\mathbf{C}^{l+1} = \arg\min_{\mathbf{C}} \mathcal{L}\left(\mathbf{\Lambda}_1^{l+1}, \mathbf{\Lambda}_2^{l+1}, \mathbf{C}^l, \mathbf{Z}^l\right) = \\ &\arg\min_{\mathbf{C}} \left\{ \frac{\rho}{2} \left\| \mathbf{C}^l - \left(\frac{1}{\rho}(\mathbf{Z}^l)^T + \mathbf{V}_1 \mathbf{\Lambda}_1^{l+1} \mathbf{V}_1^T + \mathbf{V}_2 \mathbf{\Lambda}_2^{l+1} \mathbf{V}_2^T \right) \right\|_F^2 \right\}.\end{aligned} \tag{B.9}$$

Consider the constraints $\mathbf{I} \odot \mathbf{C} \geq \mathbf{0}$ and $\mathbf{A} \odot \mathbf{C} \leq \mathbf{0}$ in (4.26); the diagonal elements are non-negative and the off-diagonal elements are non-positive. We obtain the solution to (B.9):

$$\begin{aligned}\mathbf{C}^{l+1} = \mathbf{I} \odot &\left[\frac{1}{\rho}(\mathbf{Z}^l)^T + \mathbf{V}_1 \mathbf{\Lambda}_1^{l+1} \mathbf{V}_1^T + \mathbf{V}_2 \mathbf{\Lambda}_2^{l+1} \mathbf{V}_2^T \right]_+ \\ + \mathbf{A} \odot &\left[\frac{1}{\rho}(\mathbf{Z}^l)^T + \mathbf{V}_1 \mathbf{\Lambda}_1^{l+1} \mathbf{V}_1^T + \mathbf{V}_2 \mathbf{\Lambda}_2^{l+1} \mathbf{V}_2^T \right]_- .\end{aligned} \tag{B.10}$$

Update $\mathbf{Z}$*:* Given $\mathbf{\Lambda}_1^{l+1}$, $\mathbf{\Lambda}_2^{l+1}$ and $\mathbf{C}^{l+1}$, then $\mathbf{Z}^{l+1}$ can be obtained by

$$\mathbf{Z}^{l+1} = \mathbf{Z}^l + \rho\left(\mathbf{V}_1 \mathbf{\Lambda}_1^{l+1} \mathbf{V}_1^T + \mathbf{V}_2 \mathbf{\Lambda}_2^{l+1} \mathbf{V}_2^T - \mathbf{C}^{l+1}\right). \tag{B.11}$$

The convergence criterion of the ADMM is $\|\mathbf{C}^{l+1} - \mathbf{C}^l\| / \|\mathbf{C}^l\| < \epsilon$ and $\left\|\mathbf{Z}^{l+1} - \mathbf{Z}^l\right\| / \left\|\mathbf{Z}^l\right\| < \epsilon$, where ϵ is a predefined threshold.